

CT_EX 宏集手册

CTEX.ORG

2026/07/08 v2.6.3*

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	16
第 2 节 简明教程	3	7.1 编号相关	17
2.1 CT _E X 宏集的组成	3	7.2 格式相关	19
2.2 CT _E X 宏集的安装和更新	3	7.3 间距、缩进相关	23
2.3 使用 CT _E X 文档类	4	7.4 目录、附录相关	27
2.4 使用 <code>ctex</code> 宏包	5	7.5 辅助命令	28
第 3 节 宏包选项与 <code>\ctexset</code> 命令	5	7.6 示例	29
第 4 节 编译方式、编码与中文字库	6	第 8 节 实用命令	30
4.1 编译方式	6	8.1 字号与间距	30
4.2 中文编码	6	8.2 中文数字转换	31
4.3 中文字库	6	8.3 杂项	31
第 5 节 排版格式设定	9	第 9 节 LuaL ^A T _E X 下的中文支持方式	31
5.1 文档默认字号	9	9.1 LuaL ^A T _E X 下替代字体的设置	32
5.2 章节标题风格	10	第 10 节 CT _E X 宏集的配置文件的	33
5.3 排版方案选项	11	10.1 修改宏包默认选项	33
第 6 节 文档汉化	13	10.2 宏包载入后的配置	34
6.1 日期汉化	13	10.3 配置标题中文翻译	34
6.2 文档标题汉化	13	10.4 自定义字体集	34
6.3 页面格式设置与汉化	15	第 11 节 第三方宏包兼容	34
		11.1 与 <code>cleveref</code> 联用	34
		11.2 与 <code>babel</code> 联用	35
		11.3 与 <code>biblatex</code> 联用	36

*`ctex-kit` rev. 91d37826.

第 12 节 对旧版本的兼容性	37	第 13 节 宏集依赖情况与手工安装方法	40
12.1 CT _E X 0.8a 及以前的版本	37	第 14 节 开发人员	42
12.2 CT _E X 0.9–CT _E X 1.0d	37	第 15 节 代码实现	43
12.3 CT _E X 1.02c 以后的 SVN 开发版	39	第 16 节 代码索引	168
12.4 CT _E X 2.2 之前的版本	39	版本历史	191
12.5 CT _E X 2.4.1 和 2.4.2	40		
12.6 CT _E X 2.5 之前的版本	40		

第 1 节 介绍

历史

CT_EX 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 X_YTeX 引擎的支持。在开发新版本时，考虑到合作开发和调试的方便，我们放弃了 `doc` 和 `DocStrip`，采取了直接编写宏包代码的方式。

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

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 [GitHub](https://github.com/CTeX-org/ctex-kit)²。

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^ATeX 对中文的支持一直不是很好。即使在 CJK 宏包解决了中文字符处理的问题以后，中文用户使用 L^ATeX 仍然要面对许多困难。这些困难里，以章节标题的中文化为最。由于中文和西文书写习惯的差异，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 CT_EX 宏集的目的之一就是解决这些 L^ATeX 文档的汉化难题。

另一方面，随着 T_EX 引擎和 L^ATeX 宏包的不断发展，L^ATeX 的中文支持方式从早期的专用系统（如 CCT）发展为适用于不同引擎的多种方式³。这些方式的适用情况和使用方式有不少细节上的差异，同时操作系统的不同、语言环境的不同等客观情况又进一步带来了更多的细节差异。我们设计 CT_EX 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。

CT_EX 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。

特别感谢夏明宇 (Mingyu Hsia, myhsia@outlook.com) 对 CT_EX 宏集的重构工作：他将原先单一的 `ctex.dtx` 源文件按功能拆分为内核、引擎适配、编排方案、辅助宏包与字库配置等多

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：pdf_TTeX 引擎下的 CJK、zhmCJK 宏包，X_YTeX 引擎下的 xeCJK 宏包和 Lua_TTeX 引擎下的 LuaTeX-ja 宏包。

⁴2018 年，CT_EX 论坛因故无限期关闭，此链接现已失效。

个源文件,大幅改善了宏集的可维护性。

关于宏集名字的说明

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

CT_EX 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
```

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。

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 C_TE_X 宏集⁷。

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

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

来安装 C_TE_X 宏集。

若希望了解 C_TE_X 宏集具体的依赖情况或手工安装宏集的方法, 请参阅第 13 节。

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

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

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

来进行更新。

2.3 使用 C_TE_X 文档类

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

C_TE_X 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}
```

⁷*nix 用户可能需要超级用户权限(sudo)才能正确安装宏集。

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

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

2.4 使用 `ctex` 宏包

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

对于建立在 L^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` `\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 $\langle key \rangle = \langle value \rangle$ 格式的定義。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 $\text{\LaTeX}$ 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

$\text{\CTeX}$ 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 $\text{\CTeX}$ 宏集的中文支持方式

编译方式	(pdf) $\text{\LaTeX}$	$\text{\XeLaTeX}$	$\text{\LuaLaTeX}$	$\text{\upLaTeX}$ *
支持宏包	CJK	xeCJK	LuaTeX-ja	原生

* $\text{\pLaTeX-ng}$ (或称 $\text{\ApLaTeX}$) 与 $\text{\upLaTeX}$ 兼容。使用 $\text{\pLaTeX-ng}$ 编译时, `ctex` 采用与 $\text{\upLaTeX}$ 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 $\text{\CTeX}$ 宏集的行为,比如宏包对文档编码、字体选择、空格、标点等的处理。具体细节将在本文档后续内容中进行阐述。

4.2 中文编码

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

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

用户需要保证编译方式、源文件编码、宏包编码选项三者一致。

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

4.3 中文字库

以往,为 $\text{\LaTeX}$ 文档配置中文支持是一件相当繁琐的事情。默认情况下, $\text{\CTeX}$ 宏集能自动检测用户使用的编译方式(参见 4.1 节)和操作系统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 $\text{\CTeX}$ 宏集进行的自动配置已经足够使用,无需用户手工干预;但是 $\text{\CTeX}$ 仍然提供了一系列选项,供在 $\text{\CTeX}$ 的自动选择机制因为意外情况失效,或者在用户有特殊需求的情况下使用。除非必要,用户不应使用这些选项。

⁹ $\text{\LaTeX}$ 、pdf $\text{\LaTeX}$ 、 $\text{\XeLaTeX}$ 、 $\text{\LuaLaTeX}$ 及 $\text{\upLaTeX}$ 。

¹⁰ $\text{\CTeX}$ 宏集现在能够识别 macOS 及 Windows 系统,并将其他系统统一归为 Linux。

表 3 CT_EX 宏集自动配置字体策略

	macOS Old ¹	macOS New ²	Windows ³	其他
X _Y LaTeX	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
LuaLaTeX 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
pdfLaTeX	不可用	不可用	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	不可用
LaTeX + DVIPDFMx	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
upLaTeX + DVIPDFMx	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 LuaLaTeX 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。

zhmap ☆ `zhmap = <true|false|zhmCJK>`

指定字体映射机制。本选项只在使用 (pdf)LaTeX 编译时有意义。

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

false 使用传统的 CJK 字库(Type 1)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|hanyi|mac|macnew|macold|ubuntu|windows|none|...>`

指定 CT_EX 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 预定义了以下七种中文字库。

adobe 使用 Adobe 公司的四款中文字体, 不支持 pdfLaTeX。

fandol 使用 Fandol 中文字体, 不支持 pdfLaTeX。

founder 使用方正公司的中文字体。

hanyi 使用汉仪字库的中文字体。

mac 使用 macOS 系统下的字体, 不支持 pdfLaTeX, 根据版本分为 macnew 和 macold 两种。

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

注意: 上述自动检测仅对 macnew 字体集中的默认字体生效。如果需要在 LuaLaTeX 下通过 `\setCJKmainfont` 等命令自定义使用 macOS 的 *downloadable* 字体, 需要配置 OSFONTDIR 环境

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。

¹² 使用 (pdf)LaTeX 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 LaTeX + 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](#)。

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

变量, 将字体目录加入搜索路径。例如, 在 shell 配置文件中添加:

例 5

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

配置后需运行 `luaotfload-tool --update` 重建字体数据库。 $\text{X}\text{Y}\text{L}\text{A}\text{T}\text{E}\text{X}$ 不受此限制, 因为 $\text{X}\text{Y}\text{L}\text{A}\text{T}\text{E}\text{X}$ 自身可通过系统接口发现字体。

macold 使用 Yosemite 或之前的华文字体。

ubuntu 使用 Ubuntu 系统下的思源宋体、思源黑体和 TEX 发行版自带的文鼎楷体, 不支持 $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ 。使用 $\text{X}\text{Y}\text{L}\text{A}\text{T}\text{E}\text{X}$ 或 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 编译时, 仿宋按朱雀仿宋(lxgw-fonts 宏包)、FandolFang、思源宋体的顺序自动选用已安装的字体; 其他编译方式的仿宋使用思源宋体。

windows 使用 Windows 系统下的中易字体和微软雅黑字体。当使用 (pdf) $\text{L}\text{A}\text{T}\text{E}\text{X}$ 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 $\text{C}\text{T}\text{E}\text{X}$ 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。 $\text{C}\text{T}\text{E}\text{X}$ 宏集将载入名为 `ctex-fontset-(name).def` 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录树下合适位置建立一个名为 `ctex-fontset-(name).def` 的文件, 在这个文件里面自定义中文字体。然后通过使用 `fontset=(name)` 选项来调用它。字体配置文件的具体写法可以参考 $\text{C}\text{T}\text{E}\text{X}$ 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 $\text{C}\text{T}\text{E}\text{X}$ 宏集的操作系统检测功能和自动设定字库功能)。例如:

例 6

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

$\text{C}\text{T}\text{E}\text{X}$ 宏集预定义的中文字库还定义了一些字体命令。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}`。在 macOS 15 及之后, 苹方字体可能尚未下载, 此时该命令将回退为 `\heiti` 的别名。

第 5 节 排版格式设定

5.1 文档默认字号

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

New: 2026-05-04

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

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

CT_EX 提供两种预设字号系统:

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

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

两种字号系统仅在以下六个字号上存在差异:

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

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

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

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

zihao

☆

zihao = {<-4|5|false}

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

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

表 4 标准字体命令与字号的对应

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

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

10pt

☆

CT_EX 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_T_EX 宏包定义的字号

11pt

☆

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

12pt

☆

档类的选项的时候,C_T_EX 文档类的字号选项会被抑制。亦即,在 `zihao` 选项之后设置 10pt 选项,`zihao` 选项将不再起作用。

标准文档类的其他选项在 C_T_EX 文档类中依旧有效。例如,设置纸张大小和方向的 `a4paper` 和 `landscape`,设置单双面的 `oneside` 和 `twoside` 等。C_T_EX 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}

本选项只能在调用 `ctex.sty` 时作为宏包选项使用。

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

C_T_EX 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 C_T_EX 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。

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,也可能影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中,那么该选项将会被自动激活。

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

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

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

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

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

5.3 排版方案选项

scheme ☆ `scheme = <chinese|plain>`

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

chinese 对 `beamer` 以外的文档类, 调整默认字号为五号字, 并调整行距为 1.3; 汉化文档中的标题名字 (如“图”、“表”、“目录”和“参考文献”等, 见 6.2 节); 在 `heading = true` 的情况下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 `indentfirst` 宏包, 以实现章节标题后的段首缩进。

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

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

设置标点处理格式。预定义好的格式有:

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

banjiao 半角式: 所有标点占半个汉字宽度;

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

CCT CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

plain 原样 (不调整标点间距)。

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

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

¹⁵使用 $\text{\LaTeX}$ 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 $\text{\LaTeX}$ 将单个换行视作一个空格。

例 7

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

将得到“汉字 分词 技术 English”;使用

例 8

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

则会得到“汉字分词技术 English”。

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

linespread ★ linespread = <数值>

接受一个浮点数值, 设置行距倍数。本选项的初始值与 scheme 有关。

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

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

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

在字体大小发生变化时, 是否自动调整段首缩进 (`\parindent`) 的大小。出于兼容性考虑, 自动调整只在当前 `\parindent` 非零时执行; 若字号变化前 `\parindent` 为 0pt, 则继续保持零缩进。这样既兼容旧文档以 `\parindent=0pt` 关闭自动缩进的做法, 也避免在 minipage、居中段落等本来不缩进的结构中因字号变化重新引入缩进。因此, autoindent 是让已有的非零缩进跟随字号变化, 而不是无条件设置 `\parindent`。

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

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

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

例 9

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

linestretch ★ linestretch = <数值或长度>

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

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

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

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

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

New: 2026-05-03

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

在 $X_{\text{g}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 下, 此选项等价于设置 `\xeCJKsetup{CJKecglue={\hskip ...}}`; 在 $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 和 $\text{upL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 下, 此选项设置 `xkanjiskip`。例如:

例 10

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

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

第 6 节 文档汉化

6.1 日期汉化

$\text{CT}_{\text{E}}\text{X}$ 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“2026 年 7 月 18 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“2026 年 7 月 18 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇二六年七月十八日”。使用全汉字的日期格式。

`old` 效果为“July 18, 2026”。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 11

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

$\text{CT}_{\text{E}}\text{X}$ 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 $\text{\LaTeX}$ 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 $\text{\LaTeX}$ 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 $\text{\LaTeX}$ 文档类)中使用 `refname` 选项会报错。

`algorithmname` ★ `algorithmname = <名字>`

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

continuation ★ continuation = {名字}

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

6.3 页面格式设置与汉化

页面格式设置与汉化的功能(及章节标题样式设置功能, 见第 7 节)由 `ctexheading` 宏包完成。加载该宏包时, 或者使用 $\text{\CTEX}$ 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, $\text{\CTEX}$ 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。

$\text{\CTEX}$ 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。 $\text{\CTEX}$ 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 $\text{\LATEX}$ 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 12

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

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

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 13

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

本例的页眉效果大致如下(有页眉线):

第 1 节——天地玄黄

- 1 -

第 7 节 章节标题样式设置

CT_EX 宏集对 L^AT_EX 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_EX 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。

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

例 14

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

7.1 编号相关

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

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

例 15

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

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

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

`secnumdepth` ★ `secnumdepth =` 〈整数或章节名称〉

New: 2020-05-06

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

本选项对 `beamer/ctexbeamer` 文档类无效。

表 5 章节层次

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

表 6 章节层次的默认设置

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

如果没有特别说明, 以下将用“...”代表各级章节标题名。

`.../name` ★ `name =` {〈前名字〉, 〈后名字〉}
`name =` {〈前名字〉}

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

例 16

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

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

表 7 `name` 选项的默认设置

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

`.../number` ★ `number = {⟨数字输出命令⟩}`

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

例 17

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

将会使 `\section` 的编号变为大写罗马数字(如 I、II 等)。

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

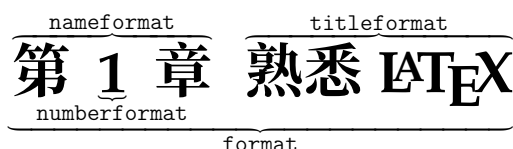
但是, `number` 选项不会影响计数器本身的输出。即设置 `section/number` 不会影响 `\thesection` 的定义(但该选项会影响 `\CTEXthesection` 的定义, 见后)。该选项的默认设置见表 8。

表 8 `number` 选项的默认设置

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

7.2 格式相关

CT_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。

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

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

Updated: 2020-04-22

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

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

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

例 18

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

`format` 选项的默认设置见表 9。

表 9 `format` 选项的默认设置

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

* 为了与 $\text{\LaTeX}2_{\epsilon}$ 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。

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

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

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

`nameformat` 选项的默认设置见表 10。

表 10 nameformat 选项的默认设置

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

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

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

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

例 19

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

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

第 4 章

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

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

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

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

例 20

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

上面的代码可以做出类似这样的章标题效果:

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

`titleformat` 选项的默认设置见表 11。

表 11 `titleformat` 选项的默认设置

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

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

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

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

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

表 12 aftername 选项的默认设置

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

表 13 aftertitle 选项的默认设置

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

part/pagestyle ★ pagestyle = {<页面格式>}

chapter/pagestyle ★

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

表 14 pagestyle 选项的默认设置

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

7.3 间距、缩进相关

section/runin ★ runin = true|false

subsection/runin ★

subsubsection/runin ★ runin 选项只对 \section 级以下标题有意义, 用于确定标题与随后的正文是否排在同一段之

paragraph/runin ★ 上。该选项的默认设置见表 15。

subparagraph/runin ★

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

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

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

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

表 15 runin 选项的默认设置

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

.../hang ★ hang = true|false

Updated: 2020-04-23

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

注意，当 hang = true 时，不恰当地设置选项 aftername 的值，可能会引发错误。这是因为当 hang = true 时， $\text{\LaTeX}$ 内部会构造一个 $\text{\hbox}$ 而进入受限水平模式（restricted horizontal mode）。若在 aftername 中加入包含 $\text{\vskip}$ 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 $\text{\section}$ 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	$\text{\parindent}$
(sub3section)	0pt
(sub4section)	同上

.../indent ★ indent = {⟨缩进间距⟩}

Updated: 2020-04-23

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

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

例 21

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

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

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

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

.../afterskip ★ afterskip = {⟨弹性间距⟩}

afterskip 选项控制章节标题下方或后面的间距。

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

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

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

.../fixskip ★ fixskip = true|false

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

表 18 before skip 选项的默认设置

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

表 19 after skip 选项的默认设置

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

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

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

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

例 22

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

该选项的默认设置见表 20。

表 20 break 选项的默认设置

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

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

afterindent 选项用于设置章节标题后首段的缩进。

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

该选项的默认设置见表 21。

表 21 afterindent 选项的默认设置

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

7.4 目录、附录相关

`tocdepth` ★ `tocdepth = <整数或章节名称>`

New: 2020-05-06

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

`.../tocline` ★ `tocline = <{格式定义}>`

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

表 22 tocline 选项的默认设置

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

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

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

其中, `\CTEXifname` 的定义见 7.5 小节。

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

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

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

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

`appendix/numbering` ★ `numbering = true|false`

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

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

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

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

该选项的默认设置见表 23。

表 23 `appendix/name` 选项的默认设置

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

7.5 辅助命令

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

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

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

<code>\CTEXheadingnumber</code>	★	<code>\CTEXheadingnumber {<层级>}</code>
<code>\CTEXheadinglabel</code>	★	<code>\CTEXheadinglabel {<层级>}</code>
<code>\CTEXifheadingnumbering</code>	★	<code>\CTEXifheadingnumbering {<层级>} {<开启时的内容>} {<关闭时的内容>}</code>

New: 2026-07-14

这组命令按给定的标题层级查询由 `\ctexset` 设置的编号数据。可用的层级取决于文档类;例如 `ctexbeamer` 支持 `part`、`section` 和 `subsection`。

`\CTEXheadingnumber` 只给出对应 `number` 选项的结果。

`\CTEXheadinglabel` 在此基础上加上 `name` 选项设置的前缀和后缀, 等价于相应的 `\CTEXthe<层级>` 命令。

`\CTEXifheadingnumbering` 仅根据对应的 `numbering` 选项选择分支。

这些查询在使用时读取当前值, 因此会响应后续和分组内的 `\ctexset` 设置。

这组命令不会应用 `format`、`nameformat`、`numberformat` 或其他标题样式。以 `ctexbeamer` 为例, 主题可用它们读取本地化编号, 同时继续用 `\insertsection` 取得标题文本, 并由 Beamer 模板控制字体、颜色和间距。

<code>\CTEXifname</code>	<code>\CTEXifname {<有名字时的内容>} {<无名字时的内容>}</code>
--------------------------	--

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

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

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

例 23

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

7.6 示例

我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 24

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

本例的设置效果大致如下:

熟悉 L^AT_EX

第 8 节 实用命令

8.1 字号与间距

`\zihao` `\zihao {<字号>}`

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

表 25 中文字号

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

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

 $\backslash\text{ziju}$ $\backslash\text{ziju} \{(\text{中文字符宽度的倍数})\}$

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

这个命令会影响 $\backslash\text{ccwd}$ 的值, 但不会影响英文字距。

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

8.2 中文数字转换

$\text{CT}\text{\TeX}$ 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。

 $\backslash\text{chinese}$ $\backslash\text{chinese} \{(\text{counter})\}$
 $\backslash\text{pagenumbering} \{(\text{chinese})\}$

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

 $\backslash\text{zhnumber}$ $\backslash\text{zhnumber} \{(\text{number})\}$

以中文格式输出数字。这里的数字可以是整数、小数和分数。

 $\backslash\text{zhdigits}$ $\backslash\text{zhdigits} \{(\text{number})\}$

将阿拉伯数字转换为中文数字串。

 $\backslash\text{CTEXnumber}$ $\backslash\text{CTEXnumber} \backslash(\text{macro}) \{(\text{number})\}$

$\backslash(\text{macro})$ 必须是一个 $\text{\TeX}$ 宏, 不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 $\langle\text{number}\rangle$ 转为中文数字, 最后将结果存储在 $\backslash(\text{macro})$ 里。对 $\backslash(\text{macro})$ 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 $\backslash\text{CTEXnumber}$, 直接使用 $\backslash\text{zhnumber}$ 即可。但是, 如果在文档中需要多次使用同一个数字 $\langle\text{number}\rangle$ 的中文形式, 就可以先用 $\backslash\text{CTEXnumber}$ 将结果保存起来备用, 而不是每次使用时都用 $\backslash\text{zhnumber}$ 现场转换一次。

 $\backslash\text{CTEXdigits}$ $\backslash\text{CTEXdigits} \backslash(\text{macro}) \{(\text{number})\}$

$\backslash\text{CTEXdigits}$ 与 $\backslash\text{CTEXnumber}$ 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

 $\backslash\text{CTeX}$ 用于显示 $\text{CT}\text{\TeX}$ 标志。

第 9 节 $\text{Lua}\text{\LaTeX}$ 下的中文支持方式

在 $\text{Lua}\text{\LaTeX}$ 下, $\text{CT}\text{\TeX}$ 宏集依赖 $\text{LuaTeX-j}\text{\LaTeX}$ 宏包来完成中文支持。该宏包是日本 $\text{\TeX}$ 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 $\text{Lua}\text{\TeX}$ 引擎下实现日本 $\text{p}\text{\TeX}$ 引擎的(大部分)功能。它为了兼容 $\text{p}\text{\LaTeX}$ 的使用习惯, 对 $\text{\LaTeX} 2_{\epsilon}$ 的 NFSS 作了不少修改和扩

充。这对于简体中文用户来说不是必要的,因而 C_T_E_X 禁用了它在 L_AT_EX 格式下的大部分设置,只保留了必要的部分。同时修改了它的字体设置方式,使得相关命令与 x_eCJK 宏包大致相同。

20150420 版以后的 Lua_{TeX}-ja 宏包开始支持竖排,但 C_T_E_X 暂不支持竖排。

9.1 *Lua_{La}T_EX* 下替代字体的设置

AlternateFont `\setCJKfamilyfont {<family>} {<base font name>}`
Updated: 2020-04-30

```
[
  <base font features> ,
  AlternateFont =
  {
    {<character range_1>} {<alternate font name_1>} ,
    {<character range_2>} {<alternate font name_2>} {<alternate font features_2>} ,
    .....
  }
]
```

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

CharRange `\setCJKfamilyfont {<family>} {<alternate font name>}`
Updated: 2020-04-30

```
[
  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 25

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

declarecharrange ★ `\ctexset`
New: 2020-04-30

```
{
  declarecharrange =
  {
    {<name_1>} {<character range_1>} ,
    {<name_2>} {<character range_2>} ,
    ...
  }
}
```

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

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

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

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

```
clearalternatefont * \ctexset
resetalternatefont * {
    clearalternatefont = {\family_1, family_2, ...} ,
    resetalternatefont = {\family_1, family_2, ...} ,
    clearalternatefont ,
    resetalternatefont
}
```

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

第 10 节 C_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 26

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

如上例所示,宏包选项通常使用 L^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 $\text{\TeX}$ 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 $\text{\TeX}$ 3 语法。

例 27

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

例 28

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

10.3 配置标题中文翻译

由于 $\text{\TeX}$ 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 $\text{\TeX}$ 宏集的配置文件,也应该与其他本地配置文件一起保存在本地 TDS 目录树下,并可以配合 `ctexopts.cfg` 等配置文件使用。

第 11 节 第三方宏包兼容

本节介绍 $\text{\TeX}$ 宏集与部分第三方宏包联用时的已知边界和推荐设置。除非另有说明,第三方宏包仍需由用户自行载入;示例只使用各宏包的公开接口。

11.1 与 `cleveref` 联用

$\text{\TeX}$ 宏集为使 `\labelformat` 功能与章节标题设置兼容,对 `cleveref` 等宏包进行了补丁。由于 `cleveref` 长期未维护(CTAN 上仍为 0.21.4),与新版 $\text{\TeX}$ 内核和 `hyperref` 存在兼容问题, $\text{\TeX}$ 提供了选项以控制补丁行为。

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

New: 2026-04-24

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

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

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

例 29

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

11.2 与 **babel** 联用

babel 负责保存和切换文档的语言状态; **C_TE_X** 宏集则通过 `\ctexset` 管理章节标签、中文数字和图表名称等中文排版设置。 **C_TE_X** 宏集不会自动载入 **babel**, 也不会把自身的本地化交给 **babel** 或访问其私有命令。

只加载两个宏包并不能让 **C_TE_X** 的章节格式随语言自动变化。例如, 切换到 `english` 后, **C_TE_X** 不会自行把“第一章”改为“Chapter 1”。需要这种结构性语言切换时, 可以在 **babel** 的 `\extraschinese` 和 `\extrasenglish` 钩子中设置 **C_TE_X** 的公开选项:

例 30

```
\documentclass[fontset=fandol]{ctexbook}
\usepackage[english,chinese,provide=*]{babel}

\newcommand\CTEXBabelChinese{%
  \ctexset{
    part/name      = {第, 部分},
    part/number    = \chinese{part},
    chapter/name   = {第, 章},
    chapter/number = \chinese{chapter},
    figurename      = 图,
    tablename      = 表,
  }%
}
\newcommand\CTEXBabelEnglish{%
  \ctexset{
    part/name      = {Part~, },
    part/number    = \thepart,
    chapter/name   = {Chapter~, },
    chapter/number = \thechapter,
    figurename      = Figure,
    tablename      = Table,
  }%
}
\addto\extraschinese{\CTEXBabelChinese}
\addto\extrasenglish{\CTEXBabelEnglish}

\begin{document}
\selectlanguage{chinese}
\part{中文部分}
```

```

\chapter{中文章节}
\begin{figure}[h]
  \centering\rule{20mm}{10mm}
  \caption{中文图题}
\end{figure}

% 下面只为对照同一编号的中英文格式；实际文档无需重置计数器。
\setcounter{part}{0}
\setcounter{chapter}{0}
\selectlanguage{english}
\part{English Part}
\chapter{English Chapter}
\begin{figure}[h]
  \centering\rule{20mm}{10mm}
  \caption{English figure caption}
\end{figure}
\end{document}

```

对章节、部分、图表标题等文档结构进行语言切换时，应使用 `\selectlanguage` 或 `otherlanguage` 环境。`\foreignlanguage` 面向行内文字，不应把它当作切换章节标签和图表名称的接口。

上例显式设置图表名称，是为了保持不同编译方式之间的结果一致。当前 `babel` 的中文语言设置在 `XYLaTeX` 和 `LuaLaTeX` 下可以提供这些名称，但在 `upLaTeX` 和 `pdfLaTeX` 的 DVI 路径下并未完整提供。完整的四引擎 MWE 和输出对比见 [Issue #626 的测试资料](#)；本节对应的文档工作见 [Issue #986](#)。

本节的设置只适用于 `babel`。`polyglossia` 使用不同的语言接口，并且可能与 `xeCJK` 的字符间机制产生冲突，不能直接套用上述示例。

11.3 与 `biblatex` 联用

`CTEX` 的 `bibname` 选项面向标准文档类的 `thebibliography` 环境：它根据文档类设置 `\bibname` 或 `\refname`。`biblatex` 则在自己的本地化流程中管理 `bibliography`、`references` 和 `shorthands` 等字符串，并在文档开始或语言切换时更新相关名称。因此，`CTEX` 的 `bibname` 选项不是控制所有 `biblatex` 标题的总开关。

根据需要，应使用 `biblatex` 提供的以下公开接口：

- 只修改当前文献表的标题，使用 `\printbibliography[title={...}]`；
- 修改当前本地化语言中的默认标题，使用 `\DefineBibliographyStrings`；
- 同时控制标题层级、目录和页眉，使用 `\defbibheading` 定义标题，再通过 `heading` 选项选择。

下面的完整示例分别演示这三类需求。实际文档通常只需要保留所需的 `\printbibliography` 调用。

例 31

```

\begin{filecontents}{\jobname.bib}
@book{sample,
  author = {Doe, Jane},
  title  = {A Sample Bibliography Entry},
  date   = {2026},
}
\end{filecontents}

\documentclass[fontset=fandol]{ctexbook}
\usepackage[backend=biber,style=numeric]{biblatex}
\addbibresource{\jobname.bib}

```

```
% 未使用语言宏包时，biblatex 的默认本地化语言通常是 english。
\DefineBibliographyStrings{english}{
  bibliography = {默认参考文献标题},
  references    = {默认参考文献标题},
}
\defbibheading{ctexbibintoc}[\bibname]{%
  \chapter*{#1}%
  \addcontentsline{toc}{chapter}{#1}%
  \markboth{#1}{#1}%
}

\begin{document}
\tableofcontents
\nocite{*}

\printbibliography[title={仅用于这一次的标题}]
\printbibliography
\printbibliography[heading=ctexbibintoc,title={加入目录的参考文献}]
\end{document}
```

不使用语言宏包时, biblatex 的默认本地化语言通常是 english, 所以上例修改 english 的字符串。若文档使用 babel 等语言框架, 应针对实际启用的 biblatex 本地化语言设置。当前 biblatex 尚未提供 chinese.lbx; babel 能切换到中文语言状态, 并不表示 biblatex 已具有完整的中文本地化字符串。

主文献表、子文献表和缩略语表等结构并不共享同一个标题。使用 biblatex-gb7714-2015 等专门的文献样式时, 应优先查阅相应样式的手册, 并使用其公开选项; 不要复制修改 \blx@... 私有命令的旧方案。CT_EX 宏集不会载入 biblatex、改写它的私有命令或替用户选择文献样式。

关于标题设置被 biblatex 本地化流程覆盖的完整复现、MWE 和输出图示, 见 [Issue #350 的测试资料](#)。与 babel 联用时, 还应同时参阅 11.2 小节。

第 12 节 对旧版本的兼容性

12.1 CT_EX 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

12.2 CT_EX 0.9–CT_EX 1.0d

在 2009 年在 ctex-kit 项目成立后, 新增了 Xe_{La}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 CT_EX 宏包相对 1.02d 版本 (2014/06/09) 的兼容性。

第 2 版的 CT_EX 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 CT_EX 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。

CT_EX 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 CT_EX 宏包第 2 版中则完全不再支持 CCT。

下面这些是在旧版本 CT_EX 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。

<u>cs4size</u> <u>c5size</u>	分别相当于 <code>zihao=-4</code> 和 <code>zihao=5</code> , 过时选项。
<u>CCT</u> <u>CCTfont</u>	相关选项已删除。
<u>indent</u> <u>noindent</u>	<code>indent</code> 和 <code>noindent</code> 什么也不做, 过时选项。 在中文版式下, <code>ctex</code> 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。 CT _E X 文档类的相关功能由章节标题的 <code>afterindent</code> 选项的值来确定。
<u>zhmap</u> <u>nozhmap</u>	<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> 选项相当于 <code>zhmap=false</code> 。过时选项。
<u>winfonts</u> <u>adobefonts</u> <u>nofonts</u>	宏包选项 <code>winfonts</code> 相当于 <code>fontset=windows</code> , <code>adobefonts</code> 相当于 <code>fontset=adobe</code> , <code>nofonts</code> 相当于 <code>fontset=none</code> 。这几个选项是过时选项, 对于新文档, 应使用 <code>fontset</code> 选项设置不同字体集。 另外, 第 2 版 CT _E 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)。
<u>punct</u> <u>nopunct</u>	旧版本中宏包 <code>punct</code> 选项没有参数, 现在可以用参数设定标点风格(5.3 节)。原有无参形式的 <code>punct</code> 选项相当于 <code>punct=quanjiao</code> 。 旧版宏包中 <code>nopunct</code> 选项的效果大致相当于 <code>punct=plain</code> 。过时选项, 不推荐使用。
<u>cap</u> <u>nocap</u>	原有的 <code>cap</code> 和 <code>nocap</code> 选项由新的 <code>scheme</code> 选项代替。(5.3 节) <code>cap</code> 选项相当于 <code>scheme = chinese</code> , <code>nocap</code> 选项相当于 <code>scheme = plain</code> 。它们均已过时, 仅因兼容性而保留。
<u>space</u> <u>nospace</u>	新版本宏包 <code>space</code> 选项增加真假值参数。(5.3 节) <code>nospace</code> 选项相当于 <code>space=false</code> , 成为过时选项。
<u>fancyhdr</u>	新版本宏包中总是自动处理对 <code>fancyhdr</code> 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 <code>fancyhdr</code> 宏包。 <code>fancyhdr</code> 选项过时, 因兼容性保留, 功能是载入 <code>fancyhdr</code> 宏包。
<u>hyperref</u>	新版本宏包中总是自动处理对 <code>hyperref</code> 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 <code>hyperref</code> 宏包。 <code>hyperref</code> 选项过时, 因兼容性保留, 功能是在导言区末尾载入 <code>hyperref</code> 宏包。
<u>fntef</u>	旧版本的 <code>fntef</code> 选项用于统一 CCT _{fntef} 与 CJK _{fntef} 的界面, 新版本 CT _E X 宏集不再支持 CCT, 也不再自动载入 CJK _{fntef} 或 xeCJK _{fntef} 宏包, 而仅在其末尾做适当格式调整。 <code>fntef</code> 选项过时, 因兼容性保留, 功能是根据引擎载入 CJK _{fntef} (pdf _T _E X) 或 xeCJK _{fntef} (X ₃ _T _E X) 宏包。 在 Lua _T _E X 下, 传统 CJK _{fntef} 会载入 CJK 宏包, 与 CT _E X 使用的 Lua _T _E X-ja 字体族接口冲突, 因此会被禁止载入。需要可断行的下划线、删除线或高亮等效果时, 可以使用 <code>lua-ul</code> 宏包提供的 <code>\underline</code> 、 <code>\strikeThrough</code> 和 <code>\highLight</code> 等命令。

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

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

<code>\CTEXindent</code>	更新 <code>\ccwd</code> 宽度后设置 <code>\parindent=2\ccwd</code> 。过时命令。
--------------------------	---

<code>\CTEXnoindent</code>	设置 <code>\parindent=0pt</code> 。过时命令。
----------------------------	---------------------------------------

<code>\CTEXsetup</code>	<code>\CTEXsetup[⟨选项⟩]{⟨标题⟩}</code> 相当于设置了 <code>\ctexset{⟨标题⟩ = {⟨选项⟩}}</code> 。过时命令。
-------------------------	---

<code>\CTEXoptions</code>	<code>\CTEXoptions[⟨选项⟩]</code> 相当于设置了 <code>\ctexset{⟨选项⟩}</code> 。过时命令。
---------------------------	--

<code>\Chinese</code>	<code>\Chinese{⟨counter⟩}</code>
-----------------------	----------------------------------

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。

<code>captiondelimit</code>	原为 <code>\CTEXoptions</code> 命令的选项, 用于控制 <code>\caption</code> 编号后面的标点。此选项已过时, 并在新版本的 $\text{\CTEX}$ 宏包中失效。
-----------------------------	---

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

例 32

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

12.3 $\text{\CTEX}$ 1.02c 以后的 SVN 开发版

$\text{\CTEX}$ 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。

$\text{\CTEX}$ 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

12.4 $\text{\CTEX}$ 2.2 之前的版本

<code>part/beforeskip</code> <code>chapter/beforeskip</code> <code>section/beforeskip</code> <code>subsection/beforeskip</code> <code>subsubsection/beforeskip</code> <code>paragraph/beforeskip</code> <code>subparagraph/beforeskip</code>	在 $\text{\CTEX}$ 2.2 之前的版本中, <code>beforeskip</code> 选项的符号还用于确定章节标题后首段的缩进。当 <code>beforeskip</code> 是负值时, 章节标题后的第一段按英文文档的排版习惯, 没有首行缩进, 否则保留首行缩进。
--	--

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

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

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

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

12.5 C_TE_X 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。

12.6 C_TE_X 2.5 之前的版本

C_TE_X 2.5 有一些比较大的变动。

```
UTF8
GBK
```

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

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

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

```
fntef
```

作为兼容性保留, 会视编译引擎载入相应宏包。

```
windows
```

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

```
windowsold
windowsnew
```

过时字库选项, 作为兼容性保留, 功能是载入 windows 字库。

```
ubuntu
```

改用思源(Noto CJK)和文鼎字库。该字库不再支持pdfL^AT_EX编译。

```
AlternateFont
```

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

除了以上列出的选项以外, 当用户使用 C_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 13 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。
- 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 pdf $\text{\LaTeX}$ 或 $\text{\LaTeX}$ + DVIPDFMx 的编译方式所需要的依赖项, 其中 zhmCJK 是可选的。
- xeCJK 宏集, 它还依赖
 - xtemplate 宏包, 它属于 l3packages 宏集。
 - fontspec 宏包。
- ➔ 以上是使用 X $\text{\LaTeX}$ 编译时的依赖项。
- luatexja 宏包, 它还依赖
 - adobemapping 宏包。
 - luaotfload 宏包, 它还依赖 lualibs 宏包。
 - luatexbase 宏包, 它还依赖 ctablestack 宏包。
 - atbegshi 宏包。
 - etoolbox 宏包。
 - iftex 宏包。
 - infwarerr 宏包。
 - ltxcmds 宏包。
 - pdftexcmds 宏包。
 - xkeyval 宏包。
 - chinese-jfm 宏包。
- fontspec 宏包。
- ➔ 以上是使用 Lua $\text{\LaTeX}$ 编译时的依赖项。
- zhmetrics-uptex 宏包。
- ➔ 以上是使用 up $\text{\LaTeX}$ 编译时的依赖项。

出于一些原因, zhmCJK 尚未被收入 $\text{\TeX}$ Live 和 MiK $\text{\TeX}$ 。因此, 若你希望使用 zhmCJK 作为 $\text{\CTEX}$ 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 $\text{\TeX}$ 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 $\text{\TeX}$ 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。

$\text{\CTEX}$ 宏集已被 $\text{\TeX}$ Live 和 MiK $\text{\TeX}$ 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 $\text{T}_{\text{E}}\text{X}$ 宏集的 **TDS 安装包**,
2. 按目录结构将文件复制到 $\text{T}_{\text{E}}\text{X}$ 发行版的本地 TDS 根目录,
3. 最后执行 `texhash` 刷新 $\text{T}_{\text{E}}\text{X}$ 发行版的 `ls-R` 数据库以完成安装。

第 14 节 开发人员

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

参考文献

- [1] DONALD ERVIN KNUTH. *The $\text{T}_{\text{E}}\text{X}$ book, Computers & Typesetting*, volume A. Addison-Wesley, 1986
- [2] FRANK MITTELBACH and MICHEL GOOSSENS. *The $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ Companion*. Tools and Techniques for Computer Typesetting. Boston: Addison-Wesley, second edition, 2004

第 15 节 代码实现

ctex-kernel.dtx

```

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

```

检查 expl3 的版本。

```

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

```

\c__ctex_engine_str 引擎检查。目前 L^AT_EX 3 将 A^PT_EX 识别为 u^PT_EX。

```

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

```

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

```

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

```

宏包载入检查。

```

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

```

`\c__ctex_everysel_loaded_bool` L^AT_EX 2021-06-01 以后的版本内建了 `everysel` 包的功能。

```

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

```

15.1 内部函数与变量

`\l__ctex_tmp_tl` 临时变量。

```

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

```

`\ctex_define_option:n` 在宏包内部使用的键值选项定义、设置命令。

```

\ctex_define:n
\ctex_set:n 59 </!ctexsize>
\ctex_set:nn 60 \cs_new_protected:Npn \ctex_define_option:n
61   { \keys_define:nn { ctex / option } }
62 <!*ctexsize>
63 \cs_new_protected:Npn \ctex_define:n
64   { \keys_define:nn { ctex } }
65 \cs_new_protected:Npn \ctex_set:n
66   { \keys_set:nn { ctex } }
67 \cs_new_protected:Npn \ctex_set:nn #1
68   { \keys_set:nn { ctex / #1 } }

```

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

```

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

```

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

```

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

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

```

对旧版本的宏包给出错误信息。

```

88 \msg_new:nnnn { ctex } { package-too-old }

```

```

89 { Support~package~`#1'~too~old. }
90 {
91   Please~update~an~up-to-date~version~of~the~package~`#1'\
92   using~your~TeX~package~manager~or~from~CTAN.
93 }

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

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

```

`\ctex_if_preamble:TF` 测试是否在 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_2\epsilon$ 的导言区。在宏包内部初始为真,文档最开始位置再设置为假。注意,钩子 `\ctex_after_end_preamble:n` 在 `\AtBeginDocument` 之后执行,可以与 `\@onlypreamble` 的行为一致。

```

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

```

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

```

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

```

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

```

110 \tl_new:N \g__ctex_encoding_tl

```

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

```

111 \bool_new:N \g__ctex_zhmCJK_bool

```

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

```

112 \tl_new:N \l__ctex_autoindent_tl

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

117 <*class>

```

```

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

```

对无效选项给出警告。

```

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

```

对过时选项或命令给出警告。

```

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

150 </class|ctex>

```

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

```

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

```

15.2 宏包选项

```

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

```

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

```

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

```

```

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

```

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

```

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

```

autoindent 自动调整段落的首行缩进功能。

```

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

```

indent 仅为兼容性保留, 已过时。

```

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

```

GBK 文档编码, 默认为 UTF-8。

UTF8

```

215     GBK .code:n =

```



```

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

```

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

```

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

```

```

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

```

punct 设置标点符号输出格式。

```

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

```

```

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

```

heading 265 heading .bool_set:N = \l__ctex_heading_bool ,

```

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

```

```

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

```

```

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

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

```

cap 和 **nocap** 是过时选项。

```

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

```

以下三项都是过时的兼容选项, 它们会载入有关宏包。

```

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

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

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

```

```

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

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

```

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

```

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

```

将未知选项传给标准文档类。

```

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

```

载入选项配置文件。

```

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

```

处理宏包选项。

```

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

```

pdfL^AT_EX 下, 如果没有显式指定编码为 UTF8, 则给出警告信息。

```

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

```

```

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

```

五号字使用标准文档类的 10pt 字体大小设置,小四号字则使用 12pt。

```

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

```

载入标准文档类。

```

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

```

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。

```

399 <*class|ctex>
400 \@ifpackageloaded { hyperref }
401 {
402   \cs_new_protected:Npn \ctex_hypersetup:n #1
403   { \hypersetup {#1} }
404 }
405 {
406   \cs_new_protected:Npn \ctex_hypersetup:n #1
407   { \PassOptionsToPackage {#1} { hyperref } }
408 }

```

15.2.1 载入引擎定义文件

最后载入 15.13.7 中的各个编译引擎的定义文件。

```

409 \ctex_file_input:n { \c__ctex_engine_file_str }
410 </class|ctex>

```

15.3 用户设置接口

```
\ctexset 411 <*class|ctex|ctexheading>
412 \NewDocumentCommand \ctexset { } { \ctex_set:n }
413 </class|ctex|ctexheading>
```

`\CTEXsetup` 过时命令。

`\CTEXoptions`

```
414 <*class|ctex>
415 \NewDocumentCommand \CTEXsetup { +0 { } } > { \TrimSpaces } m }
416 {
417   \tl_if_blank:nTF {#1}
418   { \ctex_deprecated_command:Nn \CTEXsetup { } }
419   {
420     \ctex_deprecated_command:Nn \CTEXsetup
421     { \ctexset {~#2~={~#1~}~}is~set. }
422     \ctex_set:nn {#2} {#1}
423   }
424 }
425 \NewDocumentCommand \CTEXoptions { +0 { } }
426 {
427   \tl_if_blank:nTF {#1}
428   { \ctex_deprecated_command:Nn \CTEXoptions { } }
429   {
430     \ctex_deprecated_command:Nn \CTEXoptions
431     { \ctexset {~#1~}is~set. }
432     \ctex_set:n {#1}
433   }
434 }
```

15.4 字距与缩进

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

```
435 \ctex_define:n
436 {
437   autoindent .choice: ,
438   autoindent .default:n = { true } ,
439   autoindent / true .code:n =
440   {
441     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
442     \ctex_select_size:
443   } ,
444   autoindent / false .code:n =
445   { \tl_clear:N \l__ctex_autoindent_tl } ,
446   autoindent / unknown .code:n =
447   {
448     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
449     \ctex_select_size:
450   }
451 }
```

`\CTEXsetfont` 无论字体大小是否变化都更新相关信息。

```
452 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
453 \cs_new_protected:Npn \ctex_select_size:
454 { \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。

```
455 \cs_new_protected:Npn \ctex_update_size:
456 {
457   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
458   {
```

```

459     \ctex_update_stretch:
460     \ctex_update_parindent:
461   }
462   { \ctex_update_ziju: }
463 }
464 \tl_const:Nc \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
465 \tl_new:N \l__ctex_ziju_tl
466 \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。

```

467 \ctex_add_to_selectfont:n
468 { \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`。

```

469 \ctex_define:n
470 {
471   linestretch .code:n =
472   {
473     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
474     \ctex_select_size:
475   } ,
476   linestretch .value_required:n = true
477 }
478 \tl_new:N \l__ctex_line_stretch_tl
479 \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 - $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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。

```

480 \cs_new_protected:Npn \ctex_update_stretch:
481 {
482   \ctex_update_em_unit:
483   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
484   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
485     { \__ctex_update_stretch_auxi: }
486     { \__ctex_update_stretch_auxii: }
487 }
488 \cs_new_protected:Npn \__ctex_update_stretch_auxi:
489 {
490   \ctex_if_ccglue_touched:TF
491     { \ctex_update_ccwd: }
492     {
493       \dim_set:Nn \l__ctex_tmp_dim
494       { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
495       \skip_set:Nn \l__ctex_ccglue_skip
496       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
497       \ctex_update_ccglue:
498     }
499 }
500 \cs_new_protected:Npn \__ctex_update_stretch_auxii:
501 { \__ctex_update_stretch_auxiii: }
502 \cs_new_protected:Npn \__ctex_update_stretch_auxiii:
503 {
504   \int_set:Nn \l__ctex_tmp_int

```

```

505     { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
506     \tex_divide:D \l__ctex_tmp_int \ccwd
507     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
508     {
509         \skip_set:Nn \l__ctex_ccglue_skip
510         {
511             \c_zero_dim plus \dim_eval:n
512             {
513                 ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
514                 \l__ctex_tmp_int
515             }
516         }
517     }
518     { \skip_zero:N \l__ctex_ccglue_skip }
519     \ctex_update_ccglue:
520 }

```

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

```

521 \cs_new_protected:Npn \ctex_update_parindent:
522 {
523     \tl_if_empty:NF \l__ctex_autoindent_tl
524     {
525         \dim_compare:nNnF \parindent = \c_zero_dim
526         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
527     }
528 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

529 \NewDocumentCommand \ziju { m }
530 { \exp_args:Ne \ctex_ziju:n {#1} \tex_ignorespaces:D }
531 \cs_new_protected:Npn \ctex_ziju:n #1
532 {
533     \tl_set:Nn \l__ctex_ziju_tl { \fp_eval:n {#1} }
534     \ctex_select_size:
535 }

```

`\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。

```

536 \cs_new_protected:Npn \ctex_update_ziju:
537 {
538     \ctex_update_em_unit:
539     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
540     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
541     \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。

```

542     {
543         \dim_set:Nn \l__ctex_tmp_dim
544         { \linewidth - \ccwd + \l__ctex_ziju_dim }
545         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
546         \tex_divide:D \l__ctex_tmp_int \ccwd
547         \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。

```

548     \dim_compare:nNnF \parindent = \c_zero_dim
549     {
550         \int_compare:nNnF \l__ctex_tmp_int < 3
551         { \int_sub:Nn \l__ctex_tmp_int { 2 } }

```



```

552     }
553     \skip_set:Nn \l__ctex_ccglue_skip
554     {
555         \l__ctex_ziju_dim
556         plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
557         minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
558         { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
559     }
560 }
561 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
562 \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 $2\backslash\text{ccwd}$ 。

```

563     \ctex_update_parindent:
564 }
565 \dim_new:N \l__ctex_ziju_dim

```

$\backslash\text{CTEXindent}$ 过时命令。
 $\backslash\text{CTEXnoindent}$

```

566 \NewDocumentCommand \CTEXindent { }
567 {
568     \ctex_deprecated_command:Nn \CTEXindent
569     { \parindent is~set~to~2\ccwd. }
570     \ctex_update_ccwd:
571     \dim_set:Nn \parindent { 2 \ccwd }
572 }
573 \NewDocumentCommand \CTEXnoindent { }
574 {
575     \ctex_deprecated_command:Nn \CTEXnoindent
576     { \parindent is~set~to~0pt. }
577     \dim_zero:N \parindent
578 }

```

15.5 中文数字与日期

需要注意的是, `ltkeys` 设置的选项列表是 $\backslash\text{raw@opt@}(\backslash\text{currname}).(\backslash\text{currentx})$, 该列表不会将 `\PassOptionsToPackage` 传递的选项完全展开。

```

579 \exp_args:Ne \PassOptionsToPackage
580 { encoding = \g__ctex_encoding_tl }
581 { zhnumber }
582 \RequirePackage { zhnumber }

```

$\backslash\text{chinese}$ 583 $\backslash\text{cs_new:Npn } \backslash\text{chinese } \{ \backslash\text{zhnum_counter:n } \}$
 584 $\backslash\text{cs_new_eq:NN } \backslash\text{@chinese } \backslash\text{@zhnum}$
 585 $\backslash\text{cs_new_eq:NN } \backslash\text{Chinese } \backslash\text{chinese}$
 586 $\backslash\text{cs_new_eq:NN } \backslash\text{CTEXcounter } \backslash\text{use_none:n}$

给 `enumitem` 宏包注册 $\backslash\text{chinese}$ 、 $\backslash\text{zhnum}$ 和 $\backslash\text{zhdig}$ 。

```

587 \ctex_at_end_package:nn { enumitem }
588 {
589     \cs_if_free:NF \AddEnumerateCounter
590     {
591         \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
592         \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
593         \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
594     }
595 }

```

```

\CTEXnumber 596 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 597 { \protected@edef #1 { \zhnumber {#2} } }
598 \NewDocumentCommand \CTEXdigits { m m }
599 { \protected@edef #1 { \zhdigits {#2} } }

today 600 \cs_set_eq:NN \CTEX@todayold \today
601 \ctex_define:n
602 {
603   today .choice: ,
604   today / old .code:n =
605     { \cs_set_eq:NN \today \CTEX@todayold } ,
606   today / small .code:n =
607     {
608       \cs_set_eq:NN \today \zhtoday
609       \zhnumsetup { time = Arabic }
610     } ,
611   today / big .code:n =
612     {
613       \cs_set_eq:NN \today \zhtoday
614       \zhnumsetup { time = Chinese }
615     } ,
616   today / unknown .code:n =
617     { \msg_error:nne { ctex } { today-undef } {#1} }
618 }
619 \msg_new:nnnn { ctex } { today-undef }
620 { Today~format~`#1'~is~undefined. }
621 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

15.6 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。

```

622 \tl_if_exist:NF \proofname
623 {
624   \tl_new:N \proofname
625   \tl_set:Nn \proofname { Proof }
626 }

627 \ctex_define:n
628 {
629   contentsname .tl_set:N = \contentsname ,
630   listfigurename .tl_set:N = \listfigurename ,
631   listtablename .tl_set:N = \listtablename ,
632   figurename .tl_set:N = \figurename ,
633   tablename .tl_set:N = \tablename ,
634   abstractname .tl_set:N = \abstractname ,
635   indexname .tl_set:N = \indexname ,
636   appendixname .tl_set:N = \appendixname ,
637   proofname .tl_set:N = \proofname ,
638   bibname .tl_set:N = \refname
639   bibname .tl_set:N = \bibname
640 }
641 \ifbeamer
642   algorithmname .tl_set:N = \algorithmname ,
643   bibname .tl_set:N = \bibname ,
644   refname .tl_set:N = \refname ,
645   continuation .tl_set:N = \insertcontinuationtext
646 \endifbeamer

647 \ifctex
648 \msg_new:nnn { ctex } { ctexbibname }
649 {
650   Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
651   The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
652 }
653 \tl_if_exist:NTF \insertcontinuationtext

```

```

654 {
655   \ctex_define:n
656   {
657     algorithmname .tl_set:N = \algorithmname ,
658     bibname       .tl_set:N = \bibname ,
659     refname       .tl_set:N = \refname ,
660     continuation .tl_set:N = \insertcontinuationtext
661   }
662 }
663 {
664   \tl_if_exist:NTF \bibname
665   { \ctex_define:n { bibname .tl_set:N = \bibname } }
666   {
667     \tl_if_exist:NTF \refname
668     { \ctex_define:n { bibname .tl_set:N = \refname } }
669     {
670       \msg_warning:nn { ctex } { ctexbibname }
671       \ctex_define:n { bibname .tl_set:N = \ctexbibname }
672     }
673   }
674 }
675 </ctex>

676 </class|ctex>

```

15.7 中文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```
677 <*class|heading>
```

15.7.1 定义标题格式选项

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

```

678 <*article|book|report>
679 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
680 { section , subsection , subsubsection , paragraph , subparagraph }
681 </article|book|report>

```

\c__ctex_headings_seq 保存各级标题名字。

```

<article|book|report|beamer> 682 \seq_const_from_clist:Nn \c__ctex_headings_seq
683 <*article|book|report>
684 {
685   part ,
<book|report> 686   chapter ,
687   section , subsection , subsubsection , paragraph , subparagraph
688 }
689 </article|book|report>
<beamer> 690 { part , section , subsection }

```

```

__ctex_initial_heading:n 691 \cs_new_protected:Npn __ctex_initial_heading:n #1
692 {
693   \tl_new:c { CTEX@pre#1 }
694   \tl_new:c { CTEX@post#1 }
695   \tl_const:ce { CTEX@the#1 }
696   {
697     \exp_not:c { CTEX@pre#1 }
698     \exp_not:c { CTEX@the#1 }
699     \exp_not:c { CTEX@post#1 }
700   }
701   \tl_const:ce { CTEX@#1name }

```

```

702 {
703   \group_begin:
704   \exp_not:c { CTEX@#1@nameformat }
705   {
706     \exp_not:c { CTEX@pre#1 }
707     \exp_not:N \tl_if_empty:NTF
708     \exp_not:c { CTEX@#1@numberformat }
709     { \exp_not:c { CTEX@the#1 } }
710     {
711       \group_begin:
712       \exp_not:c { CTEX@#1@numberformat }
713       \exp_not:c { CTEX@the#1 }
714       \group_end:
715     }
716     \exp_not:c { CTEX@post#1 }
717   }
718   \group_end:
719 }
720 }

```

```

\__ctex_def_heading_keys:n 721 \cs_new_protected:Npn \__ctex_def_heading_keys:n #1
722 {
723   \exp_args:NNe \tl_put_right:Nn \l__ctex_tmp_tl
724   {
725     #1 .meta:nn = { ctex / #1 } { ##1 } ,
726     #1 / name .code:n =
727     { \ctex_assign_heading_name:nn {#1} { ##1 } } ,
728     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
729     #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
730     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
731     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
732     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
733     #1 / numbering .initial:n = true ,
734     #1 / beforeskip .initial:n = \c_zero_skip ,
735     #1 / afterskip .initial:n = \c_zero_skip ,
736     #1 / indent .initial:n = \c_zero_dim ,
737     #1 / beforeskip .value_required:n = true ,
738     #1 / afterskip .value_required:n = true ,
739     #1 / indent .value_required:n = true ,
740     <article|book|report>
741     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
742     #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
743     #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
744     #1 / hang .initial:n = true ,
745     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
746     #1 / tocline .cs_set:Np = \exp_not:c { CTEX@#1@tocline } ##1##2 ,
747     \__ctex_plus_key_aux:nn {#1} { break } ,
748     </article|book|report>
749     \__ctex_plus_key_aux:nn {#1} { format } ,
750     \__ctex_plus_key_aux:nn {#1} { nameformat } ,
751     \__ctex_plus_key_aux:nn {#1} { numberformat } ,
752     \__ctex_plus_key_aux:nn {#1} { titleformat } ,
753     \__ctex_plus_key_aux:nn {#1} { aftername } ,
754     \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
755   }
756 }
757 \cs_new:Npn \__ctex_plus_key_aux:nn #1#2
758 {
759   #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
760   #1 / #2 + .code:n =
761   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { ##1 } } ,
762   #1 / #2 ~ + .code:n =
763   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { ##1 } }
764 }

```

\ctex_assign_heading_name:nn name 的值是一个至多两个元素的逗号分隔列表。由于 L^AT_EX 3 的 clist 总是会自动忽略空元素，__ctex_assign_heading_name:nnn

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。

```

765 \NewDocumentCommand \ctex_assign_heading_name:nn
766 { m > { \SplitArgument { 1 } { , } } +m }
767 { \__ctex_assign_heading_name:nnn {#1} #2 }
768 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
769 {
770   \tl_set:cn { CTEX@pre#1 } {#2}
771   \tl_if_novalue:nTF {#3}
772     { \tl_clear:c { CTEX@post#1 } }
773     { \tl_set:cn { CTEX@post#1 } {#3} }
774 }

```

`part/pagestyle` 只在 `ctexbook` 和 `ctexrep` 下有定义。
`chapter/pagestyle`
`chapter/lofskip`
`chapter/lotskip`

```

775 \group_begin:
776 <*book|report>
777 \tl_set:Nn \l__ctex_tmp_tl
778 {
779   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
780   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
781   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
782   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
783   chapter / lofskip .initial:n = \c_zero_skip ,
784   chapter / lotskip .initial:n = \c_zero_skip ,
785   chapter / lofskip .value_required:n = true ,
786   chapter / lotskip .value_required:n = true ,
787 }
788 </book|report>
789 <*article|beamer>
790 \tl_clear:N \l__ctex_tmp_tl
791 </article|beamer>

```

定义标题键值选项。

```

792 \seq_map_inline:Nn \c__ctex_headings_seq
793 {
794   \__ctex_initial_heading:n {#1}
795   \__ctex_def_heading_keys:n {#1}
796 }
797 \exp_args:NNo \group_end: \ctex_define:n { \l__ctex_tmp_tl }

```

`\CTEXheadingnumber` 查询标题设置,但不应用任何标题格式。`\CTEXheadinglabel` 是 `\CTEXthe`(层级)的参数化等价形式。
`\CTEXheadinglabel`
`\CTEXifheadingnumbering`

```

798 \NewExpandableDocumentCommand \CTEXheadingnumber { m }
799 { \use:c { CTEX@the#1 } }
800 \NewExpandableDocumentCommand \CTEXheadinglabel { m }
801 { \use:c { CTEXthe#1 } }
802 \NewExpandableDocumentCommand \CTEXifheadingnumbering { m }
803 { \bool_if:cTF { CTEX@#1@numbering } }

```

`\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。

```

804 \cs_new_protected:Npn \CTEX@heading@format@initial
805 {
806   \normalfont
807   \tl_clear:N \l__ctex_autoindent_tl
808   \int_set:Nn \tex_interlinepenalty:D { 10 000 }
809   \tex_noindent:D
810 }

```

`\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。

```

811 \cs_new_protected:Npn \ctex_indent_box:n #1
812 {
813   \dim_set:Nn \tex_parindent:D {#1}
814   \__ctex_insert_indent:
815 }
816 \cs_new_protected:Npn \__ctex_insert_indent:
817 {
818   \dim_compare:nNnF \tex_parindent:D = \c_zero_dim
819     { \tex_indent:D }
820 }
821 \cs_new_eq:NN \CTEX@indentbox \ctex_indent_box:n

```

15.7.2 标准标题命令的修改

822 `<*article|book|report>`

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

```

823 <*book|report>
824 \cs_new_protected:Npn \CTEX@fixtopskip
825 {
826   \CTEX@fixheadingskip
827   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
828     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
829 }
830 </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 `before` 或者 `after` 作为修正。

```

831 \cs_new_protected:Npn \CTEX@fixheadingskip
832 {
833   \par
834   \dim_set:Nn \tex_prevdepth:D { -1000pt }
835   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
836 }
837 \skip_new:N \l__ctex_heading_skip
838 \cs_new_protected:Npn \CTEX@setheadingskip
839 { \skip_set:Nn \l__ctex_heading_skip }
840 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\CTEX@setthispagestyle` 将 `\CTEX@(part|chapter)@pagestyle` 展开后再传给 `\thispagestyle`。

```

841 <*book|report>
842 \cs_new_protected:Npn \CTEX@setthispagestyle #1
843 { \exp_args:Ne \thispagestyle { \use:c { CTEx@#1@pagestyle } } }
844 </book|report>

```

`\partmark` 提供 `\partmark`。

```

845 \ProvideDocumentCommand \partmark { m }
846 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
847 \cs_new_eq:NN \CTEXifname \use_ii:nn
848 \cs_new_protected:Npn \CTEX@ifnametrue
849 { \cs_set_eq:NN \CTEXifname \use_i:nn }
850 \cs_new_protected:Npn \CTEX@ifnamefalse
851 { \cs_set_eq:NN \CTEXifname \use_ii:nn }

```

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

```

852 <*book|report>
853 \cs_new_protected:Npn \CTEX@addloflotskip #1
854 {
855   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
856   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
857   {
858     \addtocontents { lof }
859     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
860   }
861   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
862   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
863   {
864     \addtocontents { lot }
865     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
866   }
867 }
868 </book|report>

```

```

\CTEX@addtocline 869 \cs_new_protected:Npn \CTEX@addtocline #1#2
870 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

15.7.2.1 part 的标题

```

\part 871 <*article>
872 \renewcommand\part{%
873   \if@noskipsec \leavevmode \fi
874   \par
875   \CTEX@part@break
876   % \addvspace{4ex}%
877   \CTEX@setheadingskip \CTEX@part@beforeskip
878   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
879   \addvspace \CTEX@headingskip
880   \ifodd \CTEX@part@afterindent
881     \@afterindenttrue
882   \else
883     \@afterindentfalse
884   \fi
885   \secdef\@part\@spart}
886 </article>
887 <*book|report>
888 \renewcommand\part{%
889   % \if@openright
890   % \cleardoublepage
891   % \else
892   % \clearpage
893   % \fi
894   \CTEX@part@break
895   % \thispagestyle{plain}%
896   \CTEX@setthispagestyle{part}%
897   \if@twocolumn
898     \onecolumn
899     \@tempwatrue
900   \else
901     \@tempwafalse
902   \fi
903   % \null\vfil
904   \CTEX@setheadingskip \CTEX@part@beforeskip
905   \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
906   \vspace*{\CTEX@headingskip}%
907   \secdef\@part\@spart}
908 </book|report>

\@part 909 <*article>
910 \def\@part[#1]#2{%

```



```

911 \ifnum \c@secnumdepth >\m@ne
912 \ifodd \CTEX@part@numbering
913 \CTEX@ifnametrue
914 \refstepcounter{part}%
915 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
916 \else
917 \CTEX@ifnamefalse
918 \CTEX@makeanchor{part*}%
919 % \addcontentsline{toc}{part}{#1}%
920 \fi
921 \else
922 \CTEX@ifnamefalse
923 \CTEX@makeanchor{part*}%
924 % \addcontentsline{toc}{part}{#1}%
925 \fi
926 \CTEX@getttitle{#1}%
927 \CTEX@addtocline{part}{#1}%
928 \partmark{#1}%
929 \begingroup
930 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
931 \CTEX@heading@format@initial
932 \CTEX@part@format{%
933 % \ifnum \c@secnumdepth >\m@ne
934 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
935 % \fi
936 \CTEX@headinghang{part}%
937 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
938 % \huge\bfseries #2%
939 \CTEX@part@titleformat{#2}%
940 % \markboth{}{}\par
941 \CTEX@part@aftertitle}\par
942 \endgroup
943 \nobreak
944 % \vskip 3ex
945 \CTEX@setheadingskip \CTEX@part@afterskip
946 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
947 \vskip \CTEX@headingskip
948 \@afterheading}
949 </article>
950 <*book|report>
951 \def\@part[#1]#2{%
952 \ifnum \c@secnumdepth >-2\relax
953 \ifodd \CTEX@part@numbering
954 \CTEX@ifnametrue
955 \refstepcounter{part}%
956 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
957 \else
958 \CTEX@ifnamefalse
959 \CTEX@makeanchor{part*}%
960 % \addcontentsline{toc}{part}{#1}%
961 \fi
962 \else
963 \CTEX@ifnamefalse
964 \CTEX@makeanchor{part*}%
965 % \addcontentsline{toc}{part}{#1}%
966 \fi
967 \CTEX@getttitle{#1}%
968 \CTEX@addtocline{part}{#1}%
969 % \markboth{}{}%
970 \partmark{#1}%
971 \begingroup
972 % \centering \interlinepenalty \@M \normalfont
973 \CTEX@heading@format@initial
974 \CTEX@part@format{%
975 % \ifnum \c@secnumdepth >-2\relax
976 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
977 % \fi

```

```

978      \CTEX@headinghang{part}%
979      {\CTEXifname{\CTEX@partname\CTEX@part@aftername}{}}%
980 %    \Huge\bfseries #2\par
981      \CTEX@part@titleformat{#2}%
982      \CTEX@part@aftertitle}\par
983    \endgroup
984    \@endpart}
985 </book|report>

\@spart 986 <*article>
987 \def\@spart#1{%
988   \CTEX@ifnamefalse
989   \CTEX@makeanchor@spart{part*}%
990   \CTEX@getttitle{#1}%
991   \begingroup
992 %    \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
993   \CTEX@heading@format@initial
994   \CTEX@part@format{%
995     \CTEX@headinghang{part}{}%
996 %    \huge \bfseries #1\par
997     \CTEX@part@titleformat{#1}%
998     \CTEX@part@aftertitle}\par
999   \endgroup
1000   \nobreak
1001 %    \vskip 3ex
1002   \CTEX@setheadingskip \CTEX@part@afterskip
1003   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
1004   \vskip \CTEX@headingskip
1005   \@afterheading}
1006 </article>
1007 <*book|report>
1008 \def\@spart#1{%
1009   \CTEX@ifnamefalse
1010   \CTEX@makeanchor@spart{part*}%
1011   \CTEX@getttitle{#1}%
1012   \begingroup
1013 %    \centering \interlinepenalty \@M \normalfont
1014   \CTEX@heading@format@initial
1015   \CTEX@part@format{%
1016     \CTEX@headinghang{part}{}%
1017 %    \Huge \bfseries #1\par%
1018     \CTEX@part@titleformat{#1}%
1019     \CTEX@part@aftertitle}\par
1020   \endgroup
1021   \@endpart}
1022 </book|report>

\@endpart 1023 <*book|report>
1024 \def\@endpart{%
1025 %    \vfil
1026   \CTEX@setheadingskip \CTEX@part@afterskip
1027   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
1028   \vskip \CTEX@headingskip
1029   \newpage
1030   \if@twoside
1031     \if@openright
1032       \null
1033       \thispagestyle{empty}%
1034       \newpage
1035     \fi
1036   \fi
1037   \if@tempswa
1038     \twocolumn
1039   \fi}
1040 </book|report>

```

15.7.2.2 chapter 的标题

```

1041 <*book|report>

\chapter 1042 \renewcommand\chapter{%
1043 % \ifopenright\cleardoublepage\else\clearpage\fi
1044 % \thispagestyle{plain}%
1045 \CTEX@chapter@break
1046 \CTEX@setthispagestyle{chapter}%
1047 \global\@topnum\z@
1048 % \@afterindentfalse
1049 \ifodd \CTEX@chapter@afterindent
1050 \@afterindenttrue
1051 \else
1052 \@afterindentfalse
1053 \fi
1054 \secdef\@chapter\@schapter}

\@chapter 1055 \def\@chapter[#1]#2{%
1056 \ifnum \c@secnumdepth >\m@ne
1057 <*book>
1058 \if@mainmatter
1059 </book>
1060 \ifodd \CTEX@chapter@numbering
1061 \CTEX@ifnametrue
1062 \refstepcounter{chapter}%
1063 % \typeout{\@chapapp\space\thechapter.}%
1064 \typeout{\CTEXthechapter}%
1065 % \addcontentsline{toc}{chapter}
1066 % {\protect\numberline{\thechapter}#1}%
1067 \else
1068 \CTEX@ifnamefalse
1069 \CTEX@makeanchor{\Hy@chapapp*}%
1070 % \addcontentsline{toc}{chapter}{#1}%
1071 \fi
1072 <*book>
1073 \else
1074 \CTEX@ifnamefalse
1075 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
1076 % \addcontentsline{toc}{chapter}{#1}%
1077 \fi
1078 </book>
1079 \else
1080 \CTEX@ifnamefalse
1081 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
1082 % \addcontentsline{toc}{chapter}{#1}%
1083 \fi
1084 \CTEX@getttitle{#1}%
1085 \CTEX@addtocline{chapter}{#1}%
1086 \chaptermark{#1}%
1087 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
1088 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
1089 \CTEX@addloflotskip{chapter}%
1090 \if@twocolumn
1091 \@topnewpage[\@makechapterhead{#2}]%
1092 \else
1093 \@makechapterhead{#2}%
1094 \@afterheading
1095 \fi}

\@schapter 1096 \def\@schapter#1{%
1097 \CTEX@ifnamefalse
1098 \CTEX@makeanchor@schapter{\Hy@chapapp*}%
1099 \CTEX@getttitle{#1}%
1100 \if@twocolumn
1101 \@topnewpage[\@makeschapterhead{#1}]%
1102 \else
1103 \@makeschapterhead{#1}%

```

```

1104 \afterheading
1105 \fi}

\@makechapterhead 1106 \def\@makechapterhead#1{%
1107 % \vspace*{50\p@}%
1108 \CTEX@setheadingskip \CTEX@chapter@beforeskip
1109 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
1110 \vspace*{\CTEX@headingskip}%
1111 \begingroup
1112 % \parindent \z@ \raggedright \normalfont
1113 \CTEX@heading@format@initial
1114 \CTEX@chapter@format{%
1115 % \ifnum \c@secnumdepth >\m@ne
1116 % \if@mainmatter
1117 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
1118 % \fi
1119 % \fi
1120 \CTEX@headinghang{chapter}%
1121 {\CTEXifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
1122 % \Huge \bfseries #1\par\nobreak
1123 \CTEX@chapter@titleformat{#1}%
1124 \CTEX@chapter@aftertitle}\par
1125 \endgroup
1126 \nobreak
1127 % \vskip 40\p@
1128 \CTEX@setheadingskip \CTEX@chapter@afterskip
1129 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
1130 \vskip \CTEX@headingskip}

\@makeschapterhead 1131 \def\@makeschapterhead#1{%
1132 % \vspace*{50\p@}%
1133 \CTEX@setheadingskip \CTEX@chapter@beforeskip
1134 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
1135 \vspace*{\CTEX@headingskip}%
1136 \begingroup
1137 % \parindent \z@ \raggedright \normalfont \interlinepenalty\@M
1138 \CTEX@heading@format@initial
1139 \CTEX@chapter@format{%
1140 \CTEX@headinghang{chapter}{}%
1141 % \Huge \bfseries #1\par\nobreak
1142 \CTEX@chapter@titleformat{#1}%
1143 \CTEX@chapter@aftertitle}\par
1144 \endgroup
1145 \nobreak
1146 % \vskip 40\p@
1147 \CTEX@setheadingskip \CTEX@chapter@afterskip
1148 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
1149 \vskip \CTEX@headingskip}

1150 </bookreport>

```

15.7.2.3 section 类的标题

\@startsection L^AT_EX 的标准参数是：

```

{\(name)}{\(level)}{\(indent)}{\(beforeskip)}{\(afterskip)}{\(style)} * [ \(\altheading) ] {\(heading)}

1151 \def\@startsection#1#2#3#4#5#6{%
1152 \if@noskipsec \leavevmode \fi
1153 \par
1154 % \@tempskipa #4\relax
1155 % \@afterindenttrue
1156 % \ifdim \@tempskipa <\z@
1157 % \@tempskipa -\@tempskipa \@afterindentfalse
1158 % \fi
1159 \CTEX@update@sectionformat@n{#1}%
1160 \ifodd \CTEX@afterindent

```

```

1161 \afterindenttrue
1162 \else
1163 \afterindentfalse
1164 \fi
1165 \ifnobreak
1166 \everypar{}%
1167 \else
1168 % \addpenalty\@secpenalty\addvspace\@tempskipa
1169 \csname CTEX@#1@break\endcsname
1170 \CTEX@setheadingskip{#4}%
1171 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
1172 \addvspace \CTEX@headingskip
1173 \fi
1174 \@ifstar
1175 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
1176 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

\@secntformat 1177 \def\@secntformat#1{%
1178 % \csname the#1\endcsname\quad}%
1179 \csname CTEX@#1name\endcsname
1180 \csname CTEX@#1@aftername\endcsname}

\@sect 1181 \def\@sect#1#2#3#4#5#6[#7]#8{%
1182 \ifnum #2>\c@secnumdepth
1183 \CTEX@ifnamefalse
1184 \CTEX@makeanchor@ssect{#1*}%
1185 \let\@svsec\@empty
1186 \else
1187 \ifodd \csname CTEX@#1@numbering\endcsname
1188 \CTEX@ifnametrue
1189 \refstepcounter{#1}%
1190 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1191 \else
1192 \CTEX@ifnamefalse
1193 \CTEX@makeanchor{#1*}%
1194 \let\@svsec\@empty
1195 \fi
1196 \fi
1197 \CTEX@getttitle{#7}%
1198 % \@tempskipa #5\relax
1199 % \ifdim \@tempskipa>\z@
1200 \unless \ifodd \CTEX@runin
1201 \begingroup
1202 \CTEX@heading@format@initial
1203 #6{%
1204 % \hangfrom{\hskip #3\relax\@svsec}%
1205 % \interlinepenalty \@M #8\@@par
1206 \CTEX@sectionhang{#3}{\@svsec}%
1207 \csname CTEX@#1@titleformat\endcsname{#8}%
1208 \csname CTEX@#1@aftertitle\endcsname}\par
1209 \endgroup
1210 \csname #1mark\endcsname{#7}%
1211 % \addcontentsline{toc}{#1}{%
1212 % \ifnum #2>\c@secnumdepth \else
1213 % \protect\numberline{\csname the#1\endcsname}%
1214 % \fi
1215 % #7}%
1216 \CTEX@addtocline{#1}{#7}%
1217 \else
1218 \def\@svsechd{%
1219 #6{%
1220 % \hskip #3\relax \@svsec #8
1221 {\CTEX@indentbox{#3}}\@svsec
1222 \csname CTEX@#1@titleformat\endcsname{#8}%
1223 \csname CTEX@#1@aftertitle\endcsname}%
1224 \csname #1mark\endcsname{#7}%
1225 % \addcontentsline{toc}{#1}{%

```

```

1226 %      \ifnum #2>\c@secnumdepth \else
1227 %      \protect\numberline{\csname the#1\endcsname}%
1228 %      \fi
1229 %      #7}%
1230      \CTEX@addtocline{#1}{#7}}%
1231 \fi
1232 \xsect{#5}}

\@ssect 1233 \def\@ssect#1#2#3#4#5{%
1234   \CTEX@ifnamefalse
1235   \CTEX@getttitle{#5}%
1236 % \@tempskipa #3\relax
1237 % \ifdim \@tempskipa>\z@
1238   \unless \ifodd \CTEX@runin
1239     \begingroup
1240       \CTEX@heading@format@initial
1241       #4{%
1242 %      \@hangfrom{\hskip #1}%
1243 %      \interlinepenalty \@M #5\__ctexpar
1244       \CTEX@sectionhang{#1}{}%
1245       \CTEX@titleformat@n{#5}%
1246       \CTEX@aftertitle}\par
1247     \endgroup
1248   \else
1249 %   \def\@svsechd{#4{\hskip #1\relax #5}}%
1250     \def\@svsechd{#4{\CTEX@indentbox{#1}}}%
1251     \CTEX@titleformat@n{#5}\CTEX@aftertitle}%
1252   \fi
1253   \xsect{#3}}

\@xsect 1254 \def\@xsect#1{%
1255 % \@tempskipa #1\relax
1256 % \ifdim \@tempskipa>\z@
1257   \unless \ifodd \CTEX@runin
1258     \par \nobreak
1259 %   \vskip \@tempskipa
1260     \CTEX@setheadingskip{#1}%
1261     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
1262     \vskip \CTEX@headingskip
1263     \@afterheading
1264   \else
1265     \@nobreakfalse
1266     \global\@noskipsectrue
1267     \everypar{%
1268       \if@noskipsec
1269         \global\@noskipsecfalse
1270         {\setbox\z@\lastbox}%
1271         \clubpenalty\@M
1272         \begingroup \@svsechd \endgroup
1273         \unskip
1274 %       \@tempskipa #1\relax
1275 %       \hskip -\@tempskipa
1276       \CTEX@heading@glue{#1}%
1277     \else
1278       \clubpenalty \@clubpenalty
1279       \everypar{}%
1280     \fi}%
1281   \fi
1282   \ignorespaces}

```

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

```

1283 \cs_new_protected:Npn \CTEX@headinghang #1
1284 {
1285   \ctex_heading_hang:cnn
1286   { \CTEX@#1@hang }
1287   { \use:c { \CTEX@#1@indent } }

```

```

1288 }
1289 \cs_new_protected:Npn \CTEX@sectionhang
1290 { \ctex_heading_hang:Nnn \CTEX@hang }

```

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

```

1291 \cs_new_protected:Npn \ctex_heading_hang:Nnn #1#2#3
1292 {
1293   \dim_set:Nn \tex_parindent:D {#2}
1294   \bool_if:NTF #1
1295     { \ctex_hang_from:n }
1296     { \use:n }
1297     { \__ctex_insert_indent: #3 }
1298 }
1299 \cs_new_protected:Npn \ctex_hang_from:n #1
1300 {
1301   \tex_noindent:D
1302   \hbox_set:Nn \l__ctex_tmp_box {#1}
1303   \tex_hangindent:D = \box_wd:N \l__ctex_tmp_box
1304   \box_use_drop:N \l__ctex_tmp_box
1305 }
1306 \cs_generate_variant:Nn \ctex_heading_hang:Nnn { c }

```

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

```

1307 \cs_new_protected:Npn \ctex_heading_glue:n #1
1308 {
1309   \group_begin:
1310     \skip_set:Nn \l__ctex_heading_skip {#1}
1311     \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
1312       { \skip_horizontal:N \l__ctex_heading_skip }
1313   \group_end:
1314 }
1315 \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。

```

1316 \cs_new_protected:Npn \CTEX@update@sectionformat@n #1
1317 {
1318   \cs_set_eq:Nc \CTEX@titleformat@n { \CTEX@#1@titleformat }
1319   \cs_set_eq:Nc \CTEX@aftertitle { \CTEX@#1@aftertitle }
1320   \cs_set_eq:Nc \CTEX@afterindent { \CTEX@#1@afterindent }
1321   \cs_set_eq:Nc \CTEX@fixskip { \CTEX@#1@fixskip }
1322   \cs_set_eq:Nc \CTEX@hang { \CTEX@#1@hang }
1323   \cs_set_eq:Nc \CTEX@runin { \CTEX@#1@runin }
1324 }
1325 \cs_new_eq:NN \CTEX@titleformat@n \use:n
1326 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
1327 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
1328 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
1329 \cs_new_eq:NN \CTEX@hang \c_true_bool
1330 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

```

\CTEX@part@tocline 1331 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 1332 {
1333   \CTEXifname
1334     { \CTEXthepart \hspace { 1em } }
1335     { }
1336   #2
1337 }
1338 <*book|report>
1339 \cs_new:Npn \CTEX@chapter@tocline #1#2
1340 {
1341   \CTEXifname
1342     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
1343     { }

```



```

1344     #2
1345   }
1346 </book|report>

\CTEXnumberline 1347 \cs_new:Npn \CTEXnumberline #1
1348   {
1349     \CTEXifname
1350     { \protect \numberline { \use:c { CTEXthe #1 } } }
1351     { }
1352   }

1353 \int_zero:N \l__ctex_tmp_int
1354 \seq_map_inline:Nn \c__ctex_section_headings_seq
1355   {
1356     \int_incr:N \l__ctex_tmp_int
1357     \cs_gset_protected:cpx {#1}
1358     {
1359       \exp_not:N \@startsection {#1}
1360       { \int_use:N \l__ctex_tmp_int }
1361       { \exp_not:c { CTEX@#1@indent } }
1362       { \exp_not:c { CTEX@#1@beforeskip } }
1363       { \exp_not:c { CTEX@#1@afterskip } }
1364       { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
1365     }
1366     \cs_new:cpn { CTEX@#1@tocline } ##1##2
1367     { \CTEXnumberline { ##1 } ##2 }
1368   }

```

15.7.2.4 附录标题

```

appendix/name 1369 \ctex_define:n
appendix/number 1370   {
appendix/numbering 1371     appendix .meta:nn = { ctex / appendix } {#1} ,
1372     appendix / name .code:n =
1373     { \ctex_assign_heading_name:nn { appendix } {#1} } ,
1374     appendix / number .tl_set:N = \CTEX@appendix@number ,
1375     appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
1376     appendix / numbering .initial:n = true
1377   }
1378 \tl_new:N \CTEX@preappendix
1379 \tl_new:N \CTEX@postappendix

\appendix 1380 \cs_new_eq:NN \CTEX@save@appendix \appendix
1381 \cs_gset_protected:Npn \appendix
1382   {
1383     \CTEX@save@appendix
1384     <*article>
1385     \gdef \CTEX@presection { \CTEX@preappendix }
1386     \gdef \CTEX@thesection { \CTEX@appendix@number }
1387     \gdef \CTEX@postsection { \CTEX@postappendix }
1388     \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
1389     </article>
1390     <*book|report>
1391     \gdef \CTEX@prechapter { \CTEX@preappendix }
1392     \gdef \CTEX@thechapter { \CTEX@appendix@number }
1393     \gdef \CTEX@postchapter { \CTEX@postappendix }
1394     \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
1395     </book|report>
1396   }

```

15.7.2.5 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。

```

1397 \cs_new_protected:Npn \CTEX@makeanchor #1
1398   { }

```

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

```

1399 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
<article> 1400 { part , spart , sect , ssect }
<book|report> 1401 { part , spart , chapter , schapter , sect , ssect }
1402 \seq_map_inline:Nn \c__ctex_headings_cs_seq
1403 {
1404   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
1405   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
1406 }

```

`\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 C_T_E_X 的定义。

```

1407 \cs_new_protected:Npn \CTEX@hyperheadinghook
1408 {
1409   \group_begin:
1410   \legacy_if:nTF { Hy@implicit }
1411   {
1412     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
1413     \seq_map_inline:Nn \c__ctex_headings_cs_seq
1414     {
1415       \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
1416       {
1417         \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
1418         \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
1419       }
1420     }
1421   }
1422   {
1423     \seq_map_inline:Nn \c__ctex_headings_cs_seq
1424     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
1425   }
1426   \group_end:
1427 }

1428 \ctex_at_end_package:nn { hyperref }
1429 {
1430   \cs_gset_protected:Npn \CTEX@makeanchor #1
1431   {
1432     \Hy@MakeCurrentHrefAuto {#1}
1433     \Hy@raisedlink
1434     {
1435       \hyper@anchorstart { \@currentHref }
1436       \hyper@anchorend
1437     }
1438   }
1439   \CTEX@hyperheadinghook
1440 }

```

15.7.2.6 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。

```

1441 \cs_new_protected:Npn \CTEX@getttitle #1
1442 { }
1443 \ctex_at_end_package:nn { nameref }
1444 {
1445   \cs_gset_protected:Npn \CTEX@getttitle { \NR@getttitle }
1446   \seq_map_inline:Nn \c__ctex_headings_cs_seq
1447   {
1448     \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
1449     { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
1450   }
1451 }

```

15.7.2.7 兼容 titlesec 宏包

我们修改了 `\@startsection` 的定义, 它的第四个(`\before`)和第五个(`\after`)参数的符号不再有特殊意义, 改由相应的选项 `afterindent` 和 `runin` 来控制。

引入 `titlesec` 宏包, 并且未设置它的 `loadonly` 选项时, `titlesec` 会展开 `section` 类标题获取它们的参数, 进行初始设置。我们需要进行一些调整。

`\ctex_titlesec_hook`: `\titleformat` 的设置保存在名为 `\ttl@<section>` 的宏中备用, 它的内容是

```
\ttl@<shape>{\format}{\label}{\sep}{\before}{\after}
```

我们这里的 `<shape>` 为 `hang` 或者 `runin`。 `\titlespacing` 的设置保存在 `\ttls@<section>` 之中, 它的内容是

```
{\left}{\right}{\before}{\after}{\afterindent}
```

其中 `<afterindent>` 为 1 或 0, 分别对应是否保留段首缩进。我们需要根据 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 `\ttls@<section>`。

```
1452 \cs_new_protected:Npn \ctex_titlesec_hook:
1453 {
1454   \@ifpackagewith { titlesec } { explicit }
1455   {
1456     \cs_set_eq:NN \__ctex_titlesec_format:Nn
1457     \__ctex_titlesec_format_explicit:Nn
1458   }
1459   { }
1460   \clist_map_inline:nn
1461   { indentafter , noindentafter , indentfirst , nonindentfirst }
1462   {
1463     \@ifpackagewith { titlesec } { ##1 }
1464     {
1465       \clist_map_break:n
1466       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
1467     }
1468     { }
1469   }
1470   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
1471 }
1472 \cs_new_protected:Npn \__ctex_titlesec_hook:n #1
1473 {
1474   \__ctex_titlesec_format:n {#1}
1475   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
1476 }
1477 \cs_new_protected:Npn \__ctex_titlesec_format:n #1
1478 {
1479   \cs_if_free:cF { ttl@#1 }
1480   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttl@#1 } {#1} }
1481 }
1482 \cs_new_protected:Npn \__ctex_titlesec_format:Nn #1#2
1483 {
1484   \tl_set:Ne #1
1485   {
1486     \bool_if:cTF { CTEX@#2@runin }
1487     { \exp_not:N \ttl@runin }
1488     { \exp_not:N \ttl@hang }
1489     \tl_tail:N #1
1490   }
1491 }
```

```

1492 \cs_new_protected:Npn \__ctex_titlesec_format_explicit:Nn #1#2
1493 {
1494   \cs_set_nopar:Npx #1 ##1
1495   {
1496     \bool_if:cTF { CTEX@#2@runin }
1497     { \exp_not:N \ttlh@runin }
1498     { \exp_not:N \ttlh@hang }
1499     \exp_args:No \tl_tail:n { #1 { } }
1500   }
1501 }
1502 \cs_new_protected:Npn \__ctex_titlesec_spacing:Nn #1#2
1503 { \tl_set:Ne #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
1504 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
1505 {
1506   \exp_not:n { {#1} {#2} {#3} {#4} }
1507   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
1508 }

1509 \@ifpackageloaded { titlesec }
1510 { }
1511 {
1512   \ctex_at_end_package:nn { titlesec }
1513   {
1514     \@ifpackagewith { titlesec } { loadonly }
1515     { }
1516     { \ctex_titlesec_hook: }
1517   }
1518 }

```

让编译时终端显示 \CTEXthechapter, 目录使用 \CTEXtheXXX 编号。

```

1519 \ctex_at_end_package:nn { titlesec }
1520 {
1521   <*book|report>
1522   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
1523   </book|report>
1524   \cs_if_free:NF \ttl@tocpart
1525   {
1526     \cs_set_protected:Npn \ttl@tocpart
1527     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
1528   }
1529   \seq_map_inline:Nn \c__ctex_headings_seq
1530   {
1531     \cs_if_exist:cF { \ttl@toc #1 }
1532     {
1533       \cs_new_protected:cpx { \ttl@toc #1 }
1534       {
1535         \tl_set:Nn \exp_not:N \ttl@a
1536         {
1537           \exp_not:N \protect
1538           \exp_not:N \numberline { \exp_not:c { CTEXthe #1 } }
1539         }
1540       }
1541     }
1542   }
1543 }

```

在 titlesec 包定义的标题中更新 \CTEXifname。

```

1544 \ctex_at_end_package:nn { titlesec }
1545 {
1546   \ctex_patch_cmd:Nnn \ttl@labelling
1547   { \let \ifttl@toclabel \ifttl@label }
1548   {
1549     \let \ifttl@toclabel \ifttl@label
1550     \CTEX@updatettlifname
1551   }

```

```

1552 \cs_new_protected:Npn \CTEX@updatettlifname
1553 { \legacy_if:nTF { ttl@label } { \CTEX@ifnametrue } { \CTEX@ifnamefalse } }
1554 }

```

15.7.2.8 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¹⁸。

\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。

```

1555 \group_begin:
1556 \char_set_catcode_other:N \#
1557 \cs_new_protected:Npn \ctex_titleps_hook:
1558 {
1559   \ctex_patch_cmd:Nnn \ttl@settopmark
1560   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
1561   {
1562     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
1563     \CTEX@titlepslabel@set {#1}
1564   }
1565   \ctex_patch_cmd:Nnn \ttl@setsubmark
1566   { \protect \@namedef { the#1 } { } }
1567   {
1568     \protect \@namedef { the#1 } { }
1569     \CTEX@titlepslabel@clear {#1}
1570   }
1571   \ctex_patch_cmd:Nnn \ttl@setsubmark
1572   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
1573   {
1574     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
1575     \CTEX@titlepslabel@set {#2}
1576   }
1577 }
1578 \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。

```

1579 \cs_new:Npn \CTEX@titlepslabel@set #1
1580 {
1581   \cs_if_free:cF { CTEXthe#1 }
1582   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
1583 }
1584 \cs_new:Npn \CTEX@titlepslabel@clear #1
1585 {
1586   \cs_if_free:cF { CTEXthe#1 }
1587   { \protect \@namedef { CTEXthe#1 } { } }
1588 }

```

titleps 宏包的功能可以由 titlesec 的选项 pagestyles 引入。

```

1589 \ctex_at_end_package:nn { titlesec }
1590 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
1591 \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 }

```

¹⁸<https://github.com/CTeX-org/ctex-kit/issues/217>

但 `\newtitlemark` 不包含章节间的层次信息, 功能上不及修改内部命令完整。

`\ttl@setifthe` 使 `\iftheXXX` 等命令在页眉设置中可用。

```

1592 \ctex_at_end_package:nn { titleps }
1593 {
1594   \cs_set_protected:Npn \ttl@setifthe #1
1595   {
1596     \exp_args:Nco \cs_set:Npn { ifthe #1 }
1597     {
1598       \CTEXifname
1599       { \protect \@firstoftwo }
1600       { \protect \@secondoftwo }
1601     }
1602   }
1603   \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
1604 }

```

15.7.3 目录标签的宽度

```

\CTEX@toc@width@n 1605 \cs_new_protected:Npn \CTEX@toc@width@n #1
1606 {
1607   \hbox_set:Nn \l__ctex_tmp_box {#1}
1608   \dim_set:Nn \@tempdima
1609   {
1610     \dim_max:nn { \@tempdima }
1611     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
1612   }
1613 }

```

`\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。

```

1614 \group_begin:
1615 \char_set_catcode_other:N \#
1616 \use:n
1617 {
1618   \group_end:
1619   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
1620   { \CTEX@toc@width@n {#1} }
1621   { }
1622   { \ctex_patch_failure:N \numberline }
1623   \cs_new_protected:Npn \__ctex_patch_toc_width:n \c_parameter_token 1
1624   {
1625     \ifpackageloaded { \c_parameter_token 1 }
1626     { }
1627     {
1628       \ctex_at_end_package:nn { \c_parameter_token 1 }
1629       {
1630         \ctex_preto_cmd:NnnTF \numberline
1631         { \char_set_catcode_letter:n { 64 } }
1632         { \CTEX@toc@width@n {#1} }
1633         { }
1634         { \ctex_patch_failure:N \numberline }
1635       }
1636     }
1637   }
1638 }
1639 \__ctex_patch_toc_width:n { tocloft }
1640 \__ctex_patch_toc_width:n { titletoc }

```

15.7.4 页眉信息的修改

```

\ps@headings 1641 <*article>
1642 \legacy_if:nTF { @twoside }
1643 {
1644   \ctex_patch_cmd:Nnn \ps@headings
1645     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
1646     { \CTEXifname { \CTEXthesection \quad } { } }
1647   \ctex_patch_cmd:Nnn \ps@headings
1648     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
1649     { \CTEXifname { \CTEXthesubsection \quad } { } }
1650 }

```

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。

```

1651 {
1652   \ctex_patch_cmd:Nnn \ps@headings
1653     { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
1654     { \CTEXifname { \CTEXthesection \quad } { } }
1655 }
1656 </article>
1657 <*book|report>
1658 \ctex_patch_cmd:Nnn \ps@headings
1659 {
1660   <book> \ifnum \c@secnumdepth > \m@ne \if@mainmatter
1661   <report> \ifnum \c@secnumdepth > \m@ne
1662     \@chapapp \thechapter . ~ \ %
1663   <report> \fi
1664   <book> \fi \fi
1665 }
1666 { \CTEXifname { \CTEXthechapter \quad } { } }
1667 \legacy_if:nTF { @twoside }
1668 {
1669   \ctex_patch_cmd:Nnn \ps@headings
1670     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
1671     { \CTEXifname { \CTEXthesection \quad } { } }
1672 }
1673 </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。

```

1674 \ctex_at_end_package:nn { fancyhdr }
1675 {
1676   \ctex_patch_cmd:Nnn \f@nch@initialise
1677     { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
1678     { \CTEXifname { \CTEXthesection \quad } { } }
1679   \ctex_patch_cmd:Nnn \f@nch@initialise
1680     { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
1681     { \CTEXifname { \CTEXthesubsection \quad } { } }
1682   \ctex_patch_cmd:Nnn \f@nch@initialise
1683     { \ifnum \c@secnumdepth > \m@ne \@chapapp \thechapter . ~ \ \fi }
1684     { \CTEXifname { \CTEXthechapter \quad } { } }
1685   \ctex_patch_cmd:Nnn \f@nch@initialise
1686     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
1687     { \CTEXifname { \CTEXthesection \quad } { } }
1688   \f@nch@initialise

```

`fancyhdr` 的 `headings` 选项会重定义 `\ps@headings`, 这里也要打补丁。

```

1689   \@ifpackagewith { fancyhdr } { headings }
1690   {
1691     <*article>
1692     \legacy_if:nTF { @twoside }
1693     {
1694       \ctex_patch_cmd:Nnn \ps@headings
1695         { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }

```

```

1696         { \CTEXifname { \CTEXthesection \quad } { } }
1697     \ctex_patch_cmd:Nnn \ps@headings
1698     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
1699     { \CTEXifname { \CTEXthesubsection \quad } { } }
1700 }
1701 {
1702     \ctex_patch_cmd:Nnn \ps@headings
1703     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
1704     { \CTEXifname { \CTEXthesection \quad } { } }
1705 }
1706 </article>
1707 <*book|report>
1708     \ctex_patch_cmd:Nnn \ps@headings
1709     {
1710         \ifnum \c@secnumdepth > \m@ne \if@mainmatter
1711         \ifnum \c@secnumdepth > \m@ne
1712             \@chapapp \thechapter . ~ \ %
1713         \fi
1714         \fi \fi
1715     }
1716     { \CTEXifname { \CTEXthechapter \quad } { } }
1717     \legacy_if:nT { @twoside }
1718     {
1719         \ctex_patch_cmd:Nnn \ps@headings
1720         { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
1721         { \CTEXifname { \CTEXthesection \quad } { } }
1722     }
1723 </book|report>
1724 }
1725 { }
1726 }
1727 </article|book|report>

```

15.7.5 beamer 标题页模板的修改

```

1728 <*beamer>
1729 \ExplSyntaxOff

```

对应 \partpage。

```

1730 \defbeamertemplate*{part page}{CTEX}[1][ ]{%
1731     \begingroup
1732     % \centering
1733     % {\usebeamerfont{part name}%
1734     % \usebeamerfont{part name}%
1735     % \vskip1em\par
1736     \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
1737     \CTEX@heading@format@initial
1738     \CTEX@part@format{%
1739         \CTEX@indentbox{\CTEX@part@indent}%
1740         \ifodd \CTEX@part@numbering
1741             \CTEX@partname \CTEX@part@aftername
1742         \fi
1743         \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
1744         % \usebeamerfont{part title}\insertpart\par
1745         \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
1746         \end{beamercolorbox}}%
1747     \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
1748     \endgroup}

```

对应 \sectionpage。

```

1749 \defbeamertemplate*{section page}{CTEX}[1][ ]{%
1750     \begingroup
1751     % \centering
1752     % {\usebeamerfont{section name}%

```



```

1753 % \usebeamerfont{section name}\sectionname~\insertsectionnumber}
1754 % \vskip1em\par
1755 \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
1756 \CTEX@heading@format@initial
1757 \CTEX@section@format{%
1758 \CTEX@indentbox{\CTEX@section@indent}%
1759 \ifodd \CTEX@section@numbering
1760 \CTEX@sectionname \CTEX@section@aftername
1761 \fi
1762 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
1763 % \usebeamerfont{section title}\insertsection\par
1764 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
1765 \end{beamercolorbox}}%
1766 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
1767 \endgroup}

```

对应 \subsectionpage。

```

1768 \defbeamertemplate*{subsection page}{CTEX}[1][]{%
1769 \begin{group}
1770 % \centering
1771 % {\usebeamerfont{subsection name}%
1772 % \usebeamerfont{subsection name}\subsectionname~\insertsubsectionnumber}
1773 % \vskip1em\par
1774 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
1775 \CTEX@heading@format@initial
1776 \CTEX@subsection@format{%
1777 \CTEX@indentbox{\CTEX@subsection@indent}%
1778 \ifodd \CTEX@subsection@numbering
1779 \CTEX@subsectionname \CTEX@subsection@aftername
1780 \fi
1781 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
1782 % \usebeamerfont{subsection title}\insertsubsection\par
1783 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
1784 \end{beamercolorbox}}%
1785 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
1786 \endgroup}

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

1787 \defbeamertemplatealias{part page}{default}{CTEX}
1788 \defbeamertemplatealias{section page}{default}{CTEX}
1789 \defbeamertemplatealias{subsection page}{default}{CTEX}

1790 \ExplSyntaxOn
1791 </beamer>

```

15.7.6 标题编号和目录的层次设置

secnumdepth secnumdepth 在 beamer 下无意义。
tocdepth

```

1792 \ctex_define:n
1793 {
1794 <!*beamer>
1795 secnumdepth .code:n = \ctex_heading_depth:ne { secnumdepth } {#1} ,
1796 secnumdepth .value_required:n = true ,
1797 <!/beamer>
1798 tocdepth .code:n = \ctex_heading_depth:ne { tocdepth } {#1} ,
1799 tocdepth .value_required:n = true
1800 }

```

\ctex_heading_depth:nn 注意此处 \setcounter 的赋值是全局的。

```

1801 \cs_new_protected:Npn \ctex_heading_depth:nn #1#2
1802 {
1803 \prop_get:NnNTF \c__ctex_heading_level_prop {#2} \l__ctex_tmp_tl
1804 { \setcounter {#1} { \l__ctex_tmp_tl } }

```

```

1805     { \setcounter {#1} { \int_eval:n {#2} } }
1806   }
1807 \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。

```

1808 \prop_const_from_keyval:Nn \c__ctex_heading_level_prop
1809   {
1810     <*article|beamer>
1811     part = 0 ,
1812     </article|beamer>
1813     <*book|report>
1814     part = -1 ,
1815     chapter = 0 ,
1816     </book|report>
1817     section = 1 ,
1818     subsection = 2 ,
1819     subsubsection = 3 ,
1820     paragraph = 4 ,
1821     subparagraph = 5
1822   }

```

15.7.7 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。

```

1823 \cs_new_protected:Npn \CTEX@setcurrentlabel@n #1
1824   {
1825     \protected@edef \@currentlabel
1826     {
1827       \cs_if_exist:cTF { CTEX@the#1 }
1828       { \exp_args:cc { p@#1 } { CTEX@the#1 } }
1829       { \exp_not:o { \@currentlabel } }
1830     }
1831   }

```

\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 CT_EX 之后被载入, 我们之前的修改将会被覆盖。较新版 L^AT_EX 内核已经包含 \labelformat, 可以直接使用。

```

1832 \cs_new_protected:Npn \ctex_varioref_hook:
1833   {
1834     \seq_map_inline:Nn \c__ctex_headings_seq
1835     { \ctex_fix_varioref_label:n { ##1 } }
1836     \ctex_at_end_package:nn { cleveref } { \ctex_cleveref_hook: }
1837   }

```

\ctex_fix_varioref_label:n varioref 宏包的 \labelformat 实际上是定义一个以 \the<#1> 为参数的宏 \p@<#1>。L^AT_EX 在定义计数器 <#1> 时, 都会将 \p@<#1> 初始化为 \@empty。如果这个宏非空, 说明用户自定义了标签格式, 我们就不再修改。这里不能使用 \exp_args:Nnc, 因为 c 这种展开格式不会将参数放在花括号内。而 \labelformat 的定义是

```
\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
```

它的第二个参数必须放在花括号内, 否则将会被作为宏的定界符号。

```

1838 \cs_new_protected:Npn \ctex_fix_varioref_label:n #1
1839   {
1840     \tl_if_empty:cT { p@#1 }
1841     { \exp_args:Nne \labelformat {#1} { \exp_not:c { CTEX@the#1 } } }
1842   }

```

`patch/cleveref` `cleveref` 宏包长期未维护, 与较新的 L^AT_EX 内核和 `hyperref` 存在兼容问题。CT_EX 提供了对 `cleveref` 的补丁, 但用户也可以选择关闭它。

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

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

1843 \ctex_define:n
1844 {
1845   patch / cleveref .bool_set:N = \l__ctex_patch_cleveref_bool ,
1846   patch / cleveref .initial:n = true
1847 }
```

`\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

1848 \cs_new_protected:Npn \ctex_cleveref_hook:
1849 {
1850   \bool_if:NT \l__ctex_patch_cleveref_bool
1851   {
1852     \@ifpackageloaded { hyperref }
1853     {
1854       \@ifpackagewith { hyperref } { implicit = false }
1855       { }
1856       { \__ctex_cleveref_hook_aux:N \H@refstepcounter }
1857     }
1858     {
1859       \__ctex_cleveref_hook_aux:N \refstepcounter@noarg
1860       \__ctex_cleveref_hook_aux:N \refstepcounter@optarg
1861     }
1862     \__ctex_cleveref_hook_aux:N \appendix
1863   }
1864 }
1865 \cs_new_protected:Npn \__ctex_cleveref_hook_aux:N #1
1866 {
1867   \ctex_patch_cmd_all:NnnnTF #1
1868   {
1869     \ExplSyntaxOff
1870     \char_set_catcode_letter:n { 64 }
1871   }
1872   { \endcsname \csname the }
1873   { \expandafter \endcsname \csname the }
1874   { }
1875   { \ctex_patch_failure:N #1 }
1876 }
```

如果 `varioref` 已经被载入, 则使用它来设置。

```
1877 \cs_if_exist:NTF \labelformat
1878 { \ctex_varioref_hook: }
1879 {
1880   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
1881   \RenewDocumentCommand \refstepcounter { m }
1882   {
1883     \CTEX@save@refstepcounter {#1}
1884     \CTEX@setcurrentlabel@n {#1}
1885   }
1886   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
1887 }
```

15.7.8 载入 $\langle scheme \rangle$ 文件

```
1888 \ctex_scheme_input:o { \l__ctex_scheme_tl }
1889 </class|heading>
```

15.7.9 ctex.sty 的 heading 选项

```
1890 <*ctex|ctexheading>
```

$\backslash c_ctex_std_class_tl$ 用于记录被引入的标准文档类。

```
1891 \clist_map_inline:nn { article , book , report , beamer }
1892 {
1893   \@ifclassloaded {#1}
1894     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
1895     { }
1896 }
```

若标准文档类被引入,则载入对应的标题定义文件。否则视 $\backslash chapter$ 是否有定义来引入 book 或者 article。

```
1897 \msg_new:nnn { ctex } { not-standard-class }
1898 {
1899   None~of~the~standard~document~classes~was~loaded.\\
1900   Heading~`#1'~is~selected.\\
1901   ctex~may~not~work~as~expected.
1902 }
<ctex> 1903 \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 1904 \use:n
1905 {
1906   \tl_if_exist:NTF \c__ctex_std_class_tl
1907     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
1908     {
1909       \cs_if_exist:NTF \chapter
1910         {
1911           \cs_if_exist:NF \if@mainmatter
1912             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
1913             \tl_const:Nn \c__ctex_class_tl { book }
1914         }
1915         { \tl_const:Nn \c__ctex_class_tl { article } }
1916       \msg_warning:nne { ctex } { not-standard-class } { \c__ctex_class_tl }
1917     }
1918   \ctex_file_input:n { ctex-heading- \c__ctex_class_tl .def }
1919 }
<ctex> 1920 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
1921 </ctex|ctexheading>
```

15.8 中文字号

```
1922 <*class|ctex|ctexsize>
```

```
<zihao> 1923 \NewDocumentCommand \zihao { m }
1924 { \exp_args:Ne \ctex_zihao:n {#1} \tex_ignorespaces:D }
\ctex_zihao:n 1925 \cs_new_protected:Npn \ctex_zihao:n #1
1926 {
1927   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
1928   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
1929   { \msg_error:nnn { ctex } { fontsize } {#1} }
1930 }
1931 \msg_new:nnnn { ctex } { fontsize }
1932 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
1933 {
1934   The~old~font~size~is~used~if~you~continue.\\
```

```

1935 The~available~font~sizes~are~listed~as~follow.\\
1936 \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
1937 }

```

15.8.1 定义中文字号

`\c__ctex_font_size_prop` 基础行距是字号的 1.2 倍, 采用 ϵ -TeX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确¹⁹。

```

1938 \prop_new:N \l__ctex_font_size_temp_prop
1939 \seq_new:N \l__ctex_font_size_temp_seq
1940 \cs_new_protected:Npn \__ctex_save_font_size:nn #1#2
1941 {
1942   \prop_put:Nne \l__ctex_font_size_temp_prop {#1}
1943   {
1944     { \dim_to_decimal:n {#2} }
1945     { \dim_to_decimal:n { (#2) * 6 / 5 } }
1946   }
1947   \seq_put_right:Nn \l__ctex_font_size_temp_seq {#1}
1948 }
1949 \cs_new_eq:NN \ctex_save_font_size:nn \__ctex_save_font_size:nn
1950 \group_begin:
1951   \prop_clear:N \l__ctex_font_size_temp_prop
1952   \seq_clear:N \l__ctex_font_size_temp_seq
1953   \str_case:onF { \g__ctex_font_size_system_tl }
1954   {
1955     { word }
1956     {
1957       \clist_map_inline:nn
1958       {
1959         { 8 } { 5 bp } ,
1960         { 7 } { 5.5 bp } ,
1961         { -6 } { 6.5 bp } ,
1962         { 6 } { 7.5 bp } ,
1963         { -5 } { 9 bp } ,
1964         { 5 } { 10.5 bp } ,
1965         { -4 } { 12 bp } ,
1966         { 4 } { 14 bp } ,
1967         { -3 } { 15 bp } ,
1968         { 3 } { 16 bp } ,
1969         { -2 } { 18 bp } ,
1970         { 2 } { 22 bp } ,
1971         { -1 } { 24 bp } ,
1972         { 1 } { 26 bp } ,
1973         { -0 } { 36 bp } ,
1974         { 0 } { 42 bp }
1975       }
1976       { \__ctex_save_font_size:nn #1 }
1977     }
1978   } { letterpress }
1979   {
1980     \clist_map_inline:nn
1981     {
1982       { 8 } { 4 bp } ,
1983       { 7 } { 5.25 bp } ,
1984       { -6 } { 7.5 bp } ,
1985       { 6 } { 8 bp } ,
1986       { -5 } { 9 bp } ,
1987       { 5 } { 10.5 bp } ,
1988       { -4 } { 12 bp } ,
1989       { 4 } { 14 bp } ,
1990       { -3 } { 15 bp } ,
1991       { 3 } { 16 bp } ,
1992       { -2 } { 18 bp } ,
1993       { 2 } { 21 bp } ,
1994       { -1 } { 24 bp } ,

```

```

1995         { 1 } { 28 bp } ,
1996         { -0 } { 36 bp } ,
1997         { 0 } { 42 bp }
1998     }
1999     { \__ctex_save_font_size:nn #1 }
2000 }
2001 }
2002 {
2003     \file_if_exist:nTF { ctex-fontsize- \g__ctex_font_size_system_tl .def }
2004     {
2005         \ctex_file_input:n
2006         { ctex-fontsize- \g__ctex_font_size_system_tl .def }
2007     }
2008     {
2009         \msg_error:nne { ctex } { fontsize-system-not-found }
2010         { \g__ctex_font_size_system_tl }
2011         \clist_map_inline:nn
2012         {
2013             { 8 } { 5 bp } ,
2014             { 7 } { 5.5 bp } ,
2015             { -6 } { 6.5 bp } ,
2016             { 6 } { 7.5 bp } ,
2017             { -5 } { 9 bp } ,
2018             { 5 } { 10.5 bp } ,
2019             { -4 } { 12 bp } ,
2020             { 4 } { 14 bp } ,
2021             { -3 } { 15 bp } ,
2022             { 3 } { 16 bp } ,
2023             { -2 } { 18 bp } ,
2024             { 2 } { 22 bp } ,
2025             { -1 } { 24 bp } ,
2026             { 1 } { 26 bp } ,
2027             { -0 } { 36 bp } ,
2028             { 0 } { 42 bp }
2029         }
2030         { \__ctex_save_font_size:nn #1 }
2031     }
2032 }
2033 \exp_args:NNe \prop_const_from_keyval:Nn \c__ctex_font_size_prop
2034 { \prop_to_keyval:N \l__ctex_font_size_temp_prop }
2035 \exp_args:NNe \seq_const_from_clist:Nn \c__ctex_font_size_seq
2036 { \seq_use:Nn \l__ctex_font_size_temp_seq { , } }
2037 \group_end:
2038 \msg_new:nnnn { ctex } { fontsize-system-not-found }
2039 {
2040     CTEX~font~size~system~`#1'~could~not~be~found.
2041     \l The~`word'~system~will~be~used~instead.
2042 }
2043 { You~may~need~to~create~`ctex-fontsize-#1.def'. }

\ctex_declare_math_sizes:nnnn 2044 \cs_new_protected:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
2045 {
2046     \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
2047     \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
2048 }

\__ctex_get_font_sizes:Nn 2049 \cs_new_protected:Npn \__ctex_get_font_sizes:Nn #1#2
2050 {
2051     \tl_clear:N #1
2052     \tl_map_inline:nn {#2}
2053     {
2054         \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
2055         { \tl_put_right:Ne #1 { { \tl_head:N \l__ctex_tmp_tl } } }
2056         { \tl_put_right:Ne #1 { { \dim_to_decimal:n { ##1 } } } }
2057     }

```

¹⁹<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

2058 }

2059 \clist_map_inline:nn
2060 {
2061   { 8 }{ 8 }{ 5pt }{ 5pt } ,
2062   { 7 }{ 7 }{ 5pt }{ 5pt } ,
2063   { -6 }{ -6 }{ 5pt }{ 5pt } ,
2064   { 6 }{ 6 }{ 5pt }{ 5pt } ,
2065   { -5 }{ -5 }{ 6pt }{ 5pt } ,
2066   { 5 }{ 5 }{ 7pt }{ 5pt } ,
2067   { -4 }{ -4 }{ 8pt }{ 6pt } ,
2068   { 4 }{ 4 }{ 5 }{ 6 } ,
2069   { -3 }{ -3 }{ -4 }{ -5 } ,
2070   { 3 }{ 3 }{ 4 }{ 5 } ,
2071   { -2 }{ -2 }{ -3 }{ -4 } ,
2072   { 2 }{ 2 }{ 3 }{ 4 } ,
2073   { -1 }{ -1 }{ -2 }{ -3 } ,
2074   { 1 }{ 1 }{ 2 }{ 3 } ,
2075   { -0 }{ -0 }{ -1 }{ -2 } ,
2076   { 0 }{ 0 }{ 1 }{ 2 }
2077 }
2078 { \ctex_declare_math_sizes:nnnn #1 }

```

15.8.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 2079 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
2080 {
2081   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
2082   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
2083   { \msg_error:nnn { ctex } { fontsize } {#2} }
2084 }
2085 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
2086 { \cs_set_protected:Npn #3 { \setfontsize #3 {#1} {#2} #4 } }

2087 \int_case:nn { \g__ctex_font_size_int }
2088 {
2089   { 0 } { \file_input:n { ctex-c5size.clo } }
2090   { 1 } { \file_input:n { ctex-cs4size.clo } }
2091 }

2092 </class|ctex|ctexsize>
2093 <*c5size>
2094 \ctex_set_font_size:Nnn \normalsize { 5 }
2095 {
2096   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2097   \abovedisplayshortskip \z@ \@plus3\p@
2098   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2099   \belowdisplayskip \abovedisplayskip
2100   \let\@listi\@listI
2101 }
2102 \ctex_set_font_size:Nnn \small { -5 }
2103 {
2104   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
2105   \abovedisplayshortskip \z@ \@plus2\p@
2106   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
2107   \def\@listi{\leftmargin\leftmarginI
2108             \topsep 4\p@ \@plus2\p@ \@minus2\p@
2109             \parsep 2\p@ \@plus\p@ \@minus\p@
2110             \itemsep \parsep}
2111   \belowdisplayskip \abovedisplayskip
2112 }
2113 \ctex_set_font_size:Nnn \footnotesize { 6 }
2114 {
2115   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
2116   \abovedisplayshortskip \z@ \@plus\p@
2117   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@

```

```

2118 \def\@listi{\leftmargin\leftmargini
2119 \topsep 3\p@ \@plus\p@ \@minus\p@
2120 \parsep 2\p@ \@plus\p@ \@minus\p@
2121 \itemsep \parsep}
2122 \belowdisplayskip \abovedisplayskip
2123 }
2124 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
2125 \ctex_set_font_size:Nnn \tiny { 7 } { }
2126 \ctex_set_font_size:Nnn \large { -4 } { }
2127 \ctex_set_font_size:Nnn \Large { -3 } { }
2128 \ctex_set_font_size:Nnn \LARGE { -2 } { }
2129 \ctex_set_font_size:Nnn \huge { 2 } { }
2130 \ctex_set_font_size:Nnn \Huge { 1 } { }
2131 </cs5size>
2132 <*cs4size>
2133 \ctex_set_font_size:Nnn \normalsize { -4 }
2134 {
2135 \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
2136 \abovedisplayshortskip \z@ \@plus3\p@
2137 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
2138 \belowdisplayskip \abovedisplayskip
2139 \let\@listi\@listI
2140 }
2141 \ctex_set_font_size:Nnn \small { 5 }
2142 {
2143 \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
2144 \abovedisplayshortskip \z@ \@plus3\p@
2145 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
2146 \def\@listi{\leftmargin\leftmargini
2147 \topsep 9\p@ \@plus3\p@ \@minus5\p@
2148 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
2149 \itemsep \parsep}
2150 \belowdisplayskip \abovedisplayskip
2151 }
2152 \ctex_set_font_size:Nnn \footnotesize { -5 }
2153 {
2154 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2155 \abovedisplayshortskip \z@ \@plus3\p@
2156 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2157 \def\@listi{\leftmargin\leftmargini
2158 \topsep 6\p@ \@plus2\p@ \@minus2\p@
2159 \parsep 3\p@ \@plus2\p@ \@minus\p@
2160 \itemsep \parsep}
2161 \belowdisplayskip \abovedisplayskip
2162 }
2163 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
2164 \ctex_set_font_size:Nnn \tiny { -6 } { }
2165 \ctex_set_font_size:Nnn \large { -3 } { }
2166 \ctex_set_font_size:Nnn \Large { -2 } { }
2167 \ctex_set_font_size:Nnn \LARGE { 2 } { }
2168 \ctex_set_font_size:Nnn \huge { -1 } { }
2169 \ctex_set_font_size:Nnn \Huge { 1 } { }
2170 </cs4size>
<ctexsize> 2171 \normalsize
2172 <*class|ctex>

```

15.9 更新行距

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

```

2173 \fp_if_nan:nF { \l__ctex_line_spread_fp }
2174 {
2175 \exp_args:Ne \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

`\footnotesep` 我们调整了行距, 可能导致脚注的间距与行距不协调, 需要调整 `\footnotesep`。标准文档类

对 `\footnotesep` 的设置是, 字体大小为 `\footnotesize` 时 `\strutbox` 的高度 (默认值是 `.7\baselineskip`)。我们沿用这个设置方法, 只需要更新具体的大小。

```
2176 \group_begin: \footnotesize \exp_args:NNNo \group_end:
2177 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
2178 }
```

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

```
2179 \normalsize
```

15.10 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。CT_EX 可以直接用在 PDF 书签中。

```
2180 \NewDocumentCommand \CTeX { }
2181 { C \TeX }
2182 \ctex_at_end_package:nn { hyperref }
2183 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

`captiondelimater` 过时选项。

```
2184 \ctex_define:n
2185 {
2186   captiondelimater .code:n =
2187   {
2188     \ctex_deprecated_option:n
2189     { You~can~load~the~package~`caption'~to~get~its~functionality. }
2190   }
2191 }
```

15.11 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。

```
2192 \cs_new_protected:Npn \ctex_fontset_error:n #1
2193 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
2194 \msg_new:nnn { ctex } { fontset-unavailable }
2195 { CTeX'fontset~`#1'~is~unavailable~in~current~mode. }
```

`\ctex_fontset_case:nnn` 3 个参数依次为 pdf_TE_X、up_TE_X 和 X_Y_TE_X/Lua_TE_X。

```
2196 \cs_new:Npx \ctex_fontset_case:nnn #1#2#3
2197 {
2198   \sys_if_engine_pdftex:TF
2199   {#1}
2200   { \sys_if_engine_uptex:TF {#2} {#3} }
2201 }
```

`\ctex_fontset_case:nnnn` 4 个参数依次为 pdf_TE_X (生成 PDF)、pdf_TE_X (生成 DVI)、up_TE_X 和 X_Y_TE_X/Lua_TE_X。

```
2202 \cs_new:Npx \ctex_fontset_case:nnnn #1#2#3#4
2203 {
2204   \sys_if_engine_pdftex:TF
2205   { \sys_if_output_pdf:TF {#1} {#2} }
2206   { \sys_if_engine_uptex:TF {#3} {#4} }
2207 }
```

`\ctex_detect_platform:` 根据操作系统判断默认字体配置。

```

2208 \cs_new_protected:Npn \ctex_detect_platform:
2209 {
2210   \sys_if_platform_windows:TF
2211     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
2212     {
2213       \ctex_if_platform_macos:TF
2214         { \tl_gset:Nn \g__ctex_fontset_tl { mac } }
2215         { \tl_gset:Nn \g__ctex_fontset_tl { fandol } }
2216     }
2217 }
```

`\ctex_if_platform_macos:TF` 以特定字体判断 macOS 系统。

```

2218 \cs_new_protected:Npn \ctex_if_platform_macos:TF
2219 { \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。

```

2220 \cs_new_protected:Npn \ctex_load_fontset:
2221 {
2222   \tl_if_empty:NTF \g__ctex_fontset_tl
2223     { \ctex_detect_platform: }
2224     {
2225       \bool_lazy_or:nnTF
2226         { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsnew } }
2227         { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsold } }
2228         {
2229           \msg_warning:nnee { ctex } { deprecated-fontset }
2230           { \g__ctex_fontset_tl } { windows }
2231           \tl_gset:Nn \g__ctex_fontset_tl { windows }
2232         }
2233         {
2234           \file_if_exist:nF { ctex-fontset- \g__ctex_fontset_tl .def }
2235           {
2236             \use:e
2237             {
2238               \ctex_detect_platform:
2239               \msg_error:nnee { ctex } { fontset-not-found }
2240               { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
2241             }
2242           }
2243         }
2244     }
2245   \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2246 }
2247 \msg_new:nnn { ctex } { deprecated-fontset }
2248 { CTeX~fontset~`#1'~is~deprecated.\\ Fontset~`#2'~will~be~used~instead. }
2249 \msg_new:nnnn { ctex } { fontset-not-found }
2250 { CTeX~fontset~`#1'~could~not~be~found.\\ Fontset~`#2'~will~be~used~instead. }
2251 { You~may~run~`mktexlsr'~firstly. }
2252 \@onlypreamble \ctex_load_fontset:
```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

2253 \ctex_define:n
2254 {
2255   fontset .code:n =
2256   {
2257     \ctex_if_preamble:TF
2258     {
2259       \str_if_eq:eeTF {#1} { none }
2260       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
2261       {
2262         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
```

```

2263         {
2264             \tl_gset:Nx \g__ctex_fontset_tl {#1}
2265             \ctex_load_fontset:
2266         }
2267         {
2268             \msg_error:nnee { ctex } { fontset-loaded }
2269             { \g__ctex_fontset_tl } {#1}
2270         }
2271     }
2272 }
2273 { \msg_error:nn { ctex } { fontset-only-preamble } }
2274 }
2275 }
2276 \msg_new:nnnn { ctex } { fontset-loaded }
2277 {
2278     CTeX~fontset~`#1'~has~been~loaded.
2279     \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
2280 }
2281 { Only~one~fontset~can~be~loaded~in~the~preamble. }
2282 \msg_new:nnn { ctex } { fontset-only-preamble }
2283 { The~fontset~option~can~be~used~only~in~preamble. }

```

载入中文字库。

```

2284 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
2285 { \ctex_load_fontset: }

```

15.12 宏包配置文件

15.12.1 ctex.cfg

```

2286 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
2287 </class>ctex>
2288 <@@=>

```

ctex-engine.dtx

```

2289 <@@=ctex>

```

15.13 特定引擎支持与设置

15.13.1 ctexbackend.cfg

对于 X_YLaTeX/pdfLaTeX/LuaLaTeX 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 LaTeX 和 upLaTeX 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 CTeX 宏集对中文支持的默认方式, 我们在用户使用 CTeX 系列文档类时, 将默认的输出驱动改为 DVIPDFMX。

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

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

```

2290 <*backend>
2291 \begingroup
2292 \expandafter\ifx\csname Umathchardef\endcsname\relax
2293 \else\expandafter\endgroup\expandafter\endinput\fi

```

```

2294 \ifodd
2295 \expandafter\ifx\csname pdfoutput\endcsname\relax
2296 \expandafter\ifx\csname enablecjktoken\endcsname\relax 0\else 1\fi
2297 \else\ifnum\pdfoutput>0 0\else 1\fi\fi\space
2298 \def\x#1{%
2299 \if\relax\detokenize{#1}\relax
2300 \gdef\@classoptionslist{dvipdfmx}%
2301 \else
2302 \let\CTEX@add\@ne
2303 \@tfor\x:={dvips}{dvipdfmx}{dvisvgm}\do{%
2304 \expandafter\in@\expandafter{\expandafter,\x,}{, #1,}%
2305 \ifin@ \let\CTEX@add\tw@ \@break@tfor \fi}%
2306 \ifodd\CTEX@add \gdef\@classoptionslist{dvipdfmx, #1}\fi
2307 \fi}
2308 \expandafter\x\expandafter{\@classoptionslist}
2309 \fi
2310 \endgroup
2311 </backend>

```

15.13.2 ctex-engine-pdfTeX.def

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

可以使用 $\text{\texttt{L^{A}T\textsubscript{E}X}}$ 2020/10/01 的钩子机制来统一设置。

```

2312 <*pdfTeX|uptex|aptex>
2313 \cs_new_protected:Npn \ctex_set_zhmap:n
2314 { \tl_gput_right:Ne \g__ctex_zhmap_tl }
2315 \cs_new_protected:Npn \ctex_use_zhmap:
2316 { \tl_use:N \g__ctex_zhmap_tl }
2317 \cs_if_exist:NTF \ctex_gadd_ltxhook:nn
2318 {
2319 \cs_new_protected:Npn \ctex_at_shipout_first:n
2320 { \ctex_gadd_ltxhook:nn { shipout/firstpage } }
2321 \ctex_at_shipout_first:n { \ctex_use_zhmap: }
2322 }
2323 {
2324 \cs_new_protected:Npn \ctex_add_dvi_zhmap:
2325 { \AtBeginDvi { \ctex_use_zhmap: } }
2326 \ctex_after_end_preamble:n { \ctex_add_dvi_zhmap: }
2327 \ctex_at_end_package:nn { atbegshi }
2328 {
2329 \cs_new_protected:Npn \ctex_at_shipout_first:n
2330 { \AtBeginShipoutFirst }
2331 \ctex_at_shipout_first:n { \ctex_use_zhmap: }
2332 \cs_gset_eq:NN \ctex_add_dvi_zhmap: \prg_do_nothing:
2333 }
2334 }
2335 \tl_new:N \g__ctex_zhmap_tl
2336 \@onlypreamble \ctex_set_zhmap:n
2337 </pdfTeX|uptex|aptex>
2338 <*pdfTeX>

```

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

```

2339 \seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
2340 { C19 , C10 , C00 , C09 , C40 , C60 }

```

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

```

2341 \cs_new_protected:Npn \ctex_family_cmap:nn #1#2

```

```

2342 {
2343   \cs_if_free:cF { #1 + #2 }
2344   {
2345     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
2346     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
2347   }
2348 }
2349 \cs_generate_variant:Nn \ctex_family_cmap:nn { ee }
2350 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:ee

```

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

```

2351 \cs_new_protected:Npn \ctex_add_cmap:n #1
2352 {
2353   \cs_if_free:NF \CJK@plane
2354   { \exp_args:Ne \__ctex_add_cmap_auxi:n { #1 \CJK@plane } }
2355 }
2356 \cs_new_protected:Npn \__ctex_add_cmap_auxi:n #1
2357 { \exp_args:Nc \__ctex_add_cmap_auxii:Nn { __ctex_add_cmap_ #1 : } {#1} }
2358 \cs_new_protected:Npn \__ctex_add_cmap_auxii:Nn #1#2
2359 {
2360   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
2361   #1
2362 }
2363 \cs_new_protected:Npn \__ctex_save_cmap:Nn #1#2
2364 {
2365   \exp_args:Ne \file_get_full_name:nNTF
2366   { \str_lowercase:n {#2} .cmap } \l__ctex_cmap_file_tl
2367   {
2368     \pdf_object_unnamed_write:ne
2369     { fstream }
2370     { { } { \l__ctex_cmap_file_tl } }
2371     \cs_new_protected:Npx #1
2372     {
2373       \pdfnobluiltintounicode \tex_font:D
2374       \tex_pdffontattr:D \tex_font:D
2375       { /ToUnicode ~ \pdf_object_ref_last: }
2376     }
2377   }
2378   { \cs_new_eq:NN #1 \prg_do_nothing: }
2379 }
2380 \tl_new:N \l__ctex_cmap_file_tl

```

`\DeclareFontFamily` 只在 pdf¹TeX 下加 CMap。如 cmap 宏包被引入, 则不重复设置。

```

2381 \sys_if_output_pdf:T
2382 {
2383   \exp_args:Nno \use:n
2384   { \cs_gset_protected:Npn \DeclareFontFamily #1#2#3 }
2385   {
2386     \DeclareFontFamily {#1} {#2} {#3}
2387     \CTEX@Family@CMap {#1} {#2}
2388   }
2389   \ctex_at_end_package:nn { cmap }
2390   {
2391     \cs_gset_eq:NN \ctex_add_cmap:n \use_none:n
2392     \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn
2393   }
2394 }

```

首先检查选项, 决定是否载入 zhmCJK 宏包。

```

2395 \bool_if:NTF \g__ctex_zhmCJK_bool
2396 {
2397   \exp_args:Ne \PassOptionsToPackage
2398   { encoding = \g__ctex_encoding_tl }

```

```

2399     { zhmCJK }
2400     \RequirePackage { zhmCJK }
2401 }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

2402 {
2403   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
2404   { \RequirePackage { CJK } }
2405   { \RequirePackage { CJKutf8 } }
2406   \RequirePackage { CJKpunct , CJKspace }

```

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

```

2407   \cs_new_protected:Npn \ctex_load_zhmap:nnnn #1#2#3#4
2408   {
2409     \tl_set:Nn \CJKrmdefault {#1}
2410     \tl_set:Nn \CJKsfdefault {#2}
2411     \tl_set:Nn \CJKttdefault {#3}
2412     \ctex_set_zhmap:n { \ctex_file_input:n { ctex-zhmap- #4 .tex } }
2413   }
2414   \@onlypreamble \ctex_load_zhmap:nnnn
2415 }

```

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

```

2416 \cs_new_protected:Npn \ctex_undecare_unicode_character:n #1
2417 { \cs_set_eq:cN { u8 : \codepoint_str_generate:n {#1} } \tex_undefined:D }
2418 \CJKaddEncHook { UTF8 }
2419 {
2420   \ctex_undecare_unicode_character:n { "3008 }
2421   \ctex_undecare_unicode_character:n { "3009 }
2422 }

```

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

```

2423 \cs_new_protected:Npn \ctex_CJK_input:n #1
2424 {
2425   \ctex_push_file:
2426   \int_set:Nn \tex_endlinechar:D { -1 }
2427   \file_input:n {#1}
2428   \ctex_pop_file:
2429 }
2430 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

`\ctex_plane_to_utfxvibe:Nn` fancyhdr 宏包的 `\nouppercase` 会将 `\uppercase` 定义为 `\relax`,而 `\CJK@surr` 需要用它
`\CJK@surr` 将 `\CJK@plane` 转化成大写字母,这就造成了冲突²⁰。我们在这里给出 `\CJK@surr` 的一个不
依赖 `\uppercase` 的实现。

```

2431 \cs_if_free:NF \CJK@surr
2432 {
2433   \cs_new_protected:Npn \ctex_plane_to_utfxvibe:Nn #1#2
2434   {
2435     \int_set:Nn \l__ctex_tmp_int
2436     { \exp_args:Ne \int_from_hex:n {#2} }
2437     \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
2438     { \tl_gset:Ne #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
2439     {
2440       \int_sub:Nn \l__ctex_tmp_int { 256 }

```

²⁰<https://github.com/CTeX-org/ctex-kit/issues/146>

```

2441         \tl_gset:Ne #1
2442         {
2443             \int_to_Hex:n
2444             { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
2445             \int_to_Hex:n
2446             { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
2447         }
2448     }
2449 }
2450 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
2451 }

```

\CJK@addcmap L^AT_EX 2021-06-01 默认载入 glyphtounicode.tex, 我们对 CJK 字体禁用这一内建设置。

```

2452 \cs_if_free:NF \CJK@addcmap
2453 {
2454     \ctex_patch_cmd:Nnn \CJK@addcmap
2455     { \pdffontattr \font@name }
2456     {
2457         \pdfnombuiltintounicode \font@name
2458         \pdffontattr \font@name
2459     }
2460 }

```

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

```

2461 \AtBeginDocument
2462 {
2463     \use:e
2464     {
2465         \str_if_eq:nnF { \l__ctex_punct_tl } { quanjiao }
2466         { \exp_not:N \punctstyle { \l__ctex_punct_tl } }
2467     }
2468 }

```

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

```

2469 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

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

```

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

```

\CJK@upperReset 可能会有一定风险, 因此我们直到导言区末尾才使用 \CJK@envStart。这样可以避免将 CJK 环境内置入 document 环境的最里层, 最后也就不需要 \clearpage。zhmCJK

已经提供类似功能。注意先使用 `\ctex_update_default_family`: 更新 `\CJKfamilydefault`。

```

2470 \bool_if:NF \g__ctex_zhmCJK_bool
2471 {
2472   \str_if_eq:ontF { \g__ctex_encoding_tl } { UTF8 }
2473   { \CJK@loadBinding { UTF8 } }
2474   { \CJK@loadBinding { standard } }
2475   \exp_args:Ne \ctex_at_end_preamble:n
2476   {
2477     \exp_not:N \CJK@envStart
2478     { } { \g__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
2479     \exp_not:N \CJKtilde
2480   }
2481 }

```

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

```

2482 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces

```

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

```

2483 \cs_new:Npn \__ctex_char:N #1
2484 { \exp_args:Ne \CTEX@char@n { \int_value:w `#1 } }
2485 \cs_new:Npn \__ctex_char:NN #1#2
2486 {
2487   \token_if_eq_meaning:NNTF #2 \protect
2488   { \__ctex_char_aux:NN #1 }
2489   { \__ctex_char_aux:NN #1#2 }
2490 }
2491 \cs_new:Npn \__ctex_char_aux:NN #1#2
2492 {
2493   \exp_last_unbraced:Ne \CTEX@char@nn
2494   {
2495     { \int_value:w `#1 }
2496     { \int_value:w `#2 }
2497   }
2498 }
2499 \cs_new:Npn \__ctex_char:NNN #1#2#3
2500 {
2501   \token_if_eq_meaning:NNTF #2 \protect
2502   { \__ctex_char_auxi:NNNN #1#3 }
2503   { \__ctex_char_auxii:NNN #1#2#3 }
2504 }
2505 \cs_new:Npn \__ctex_char_auxi:NNNN #1#2#3#4
2506 { \__ctex_char_auxii:NNN #1#2#4 }
2507 \cs_new:Npn \__ctex_char_auxii:NNN #1#2#3
2508 {
2509   \exp_last_unbraced:Ne \CTEX@char@nnn
2510   {
2511     { \int_value:w `#1 }
2512     { \int_value:w `#2 }
2513     { \int_value:w `#3 }
2514   }
2515 }
2516 \cs_new:Npn \__ctex_char:NNNN #1#2#3#4
2517 {
2518   \token_if_eq_meaning:NNTF #2 \protect
2519   { \__ctex_char_auxi:NNNN #1#3 }
2520   { \__ctex_char_auxii:NNNN #1#2#3#4 }
2521 }

```



```

2522 \cs_new:Npn \__ctex_char_auxi:NNNN #1#2#3#4#5
2523 { \__ctex_char_auxii:NNNN #1#2#3#5 }
2524 \cs_new:Npn \__ctex_char_auxii:NNNN #1#2#3#4
2525 {
2526   \exp_last_unbraced:Ne \CTEX@char@nnnn
2527   {
2528     { \int_value:w `#1 }
2529     { \int_value:w `#2 }
2530     { \int_value:w `#3 }
2531     { \int_value:w `#4 }
2532   }
2533 }
2534 \cs_gset_eq:NN \CJK@X \__ctex_char:N
2535 \cs_gset_eq:NN \CJK@XX \__ctex_char:NN
2536 \cs_gset_eq:NN \CJK@XXX \__ctex_char:NNN
2537 \cs_gset_eq:NN \CJK@XXXX \__ctex_char:NNNN

```

CJKutf8 包优先使用 \u8:xxx/\u8:xxxx 的 L^AT_EX 定义。

```

2538 \str_if_eq:onF { \g__ctex_encoding_tl } { GBK }
2539 {
2540   \exp_args:Nne \use:n
2541   { \cs_gset:Npn \__ctex_char_aux:NN #1#2 }
2542   {
2543     \exp_not:N \cs_if_exist_use:cF
2544     { u8: \exp_not:N \tl_to_str:n { #1#2 } }
2545     { \exp_not:o { \__ctex_char_aux:NN {#1} {#2} } }
2546   }
2547   \exp_args:Nne \use:n
2548   { \cs_gset:Npn \__ctex_char_auxii:NNN #1#2#3 }
2549   {
2550     \exp_not:N \cs_if_exist_use:cF
2551     { u8: \exp_not:N \tl_to_str:n { #1#2#3 } }
2552     { \exp_not:o { \__ctex_char_auxii:NNN {#1} {#2} {#3} } }
2553   }
2554   \exp_args:Nne \use:n
2555   { \cs_gset:Npn \__ctex_char_auxii:NNNN #1#2#3#4 }
2556   {
2557     \exp_not:N \cs_if_exist_use:cF
2558     { u8: \exp_not:N \tl_to_str:n { #1#2#3#4 } }
2559     { \exp_not:o { \__ctex_char_auxii:NNNN {#1} {#2} {#3} {#4} } }
2560   }
2561 }

```

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

```

2562 \group_begin:
2563 \char_set_catcode_active:n { "7F }
2564 \cs_new:Npn \__ctex_char_preproc:w #1 ^^7f #2 ^^7f
2565 { \exp_args:No \CTEX@char@nn { \int_value:w `#1 } {#2} }
2566 \cs_gset_eq:NN \mule@arg \__ctex_char_preproc:w
2567 \group_end:
2568 \exp_args:Ne \ctex_at_end:n
2569 {
2570   \char_set_catcode:nn
2571   { \int_value:w "7F }
2572   { \char_value_catcode:n { "7F } }
2573 }

```

\CTEX@char@n 额外加入 \mode_leave_vertical: 是为了解决某些历史遗留问题。

```

\CTEX@char@nn
\CTEX@char@nnnn 2574 \cs_new_protected:Npn \CTEX@char@n #1
\CTEX@char@nnnn 2575 {
2576   \mode_leave_vertical:
2577   \use:c { CJK@ #1 }
2578   \CJK@ignorespaces

```

```

2579 }
2580 \cs_new_protected:Npn \CTEX@char@nn #1#2
2581 {
2582   \mode_leave_vertical:
2583   \use:c { CJK@ #1 } {#2}
2584   \CJK@ignorespaces
2585 }
2586 \cs_new_protected:Npn \CTEX@char@nnn #1#2#3
2587 {
2588   \mode_leave_vertical:
2589   \use:c { CJK@ #1 } {#2} {#3}
2590   \CJK@ignorespaces
2591 }
2592 \cs_new_protected:Npn \CTEX@char@nnnn #1#2#3#4
2593 {
2594   \mode_leave_vertical:
2595   \use:c { CJK@ #1 } {#2} {#3} {#4}
2596   \CJK@ignorespaces
2597 }

```

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

```

2598 \cs_new_protected:Npn \CTEX@char@nn@n #1 { \CTEX@char@nn #1 }
2599 \cs_new_protected:Npn \CTEX@char@nnn@n #1 { \CTEX@char@nnn #1 }
2600 \cs_new_protected:Npn \CTEX@char@nnnn@n #1 { \CTEX@char@nnnn #1 }
2601 \cs_new:Npn \__ctex_char_wrap:nn #1#2
2602 { \CTEX@char@nn@n { {#1} {#2} } }
2603 \cs_new:Npn \__ctex_char_wrap:nnn #1#2#3
2604 { \CTEX@char@nnn@n { {#1} {#2} {#3} } }
2605 \cs_new:Npn \__ctex_char_wrap:nnnn #1#2#3#4
2606 { \CTEX@char@nnnn@n { {#1} {#2} {#3} {#4} } }
2607 \tl_put_right:Nn \l_text_case_exclude_arg_tl
2608 {
2609   \CTEX@char@n
2610   \CTEX@char@nn@n
2611   \CTEX@char@nnn@n
2612   \CTEX@char@nnnn@n
2613 }
2614 \text_declare_expand_equivalent:Nn \CTEX@char@nn { \__ctex_char_wrap:nn }
2615 \text_declare_expand_equivalent:Nn \CTEX@char@nnn { \__ctex_char_wrap:nnn }
2616 \text_declare_expand_equivalent:Nn \CTEX@char@nnnn { \__ctex_char_wrap:nnnn }

```

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

```

2617 \cs_new:Npn \__ctex_char_raw:n #1
2618 { \char_generate:nn {#1} { 12 } }
2619 \cs_new:Npn \__ctex_char_raw:nn #1#2
2620 {
2621   \char_generate:nn {#1} { 12 }
2622   \char_generate:nn {#2} { 12 }
2623 }
2624 \cs_new:Npn \__ctex_char_raw:nnn #1#2#3
2625 {
2626   \char_generate:nn {#1} { 12 }
2627   \char_generate:nn {#2} { 12 }
2628   \char_generate:nn {#3} { 12 }
2629 }
2630 \cs_new:Npn \__ctex_char_raw:nnnn #1#2#3#4
2631 {
2632   \char_generate:nn {#1} { 12 }
2633   \char_generate:nn {#2} { 12 }
2634   \char_generate:nn {#3} { 12 }
2635   \char_generate:nn {#4} { 12 }

```

```

2636 }
2637 \cs_new:Npn \__ctex_char_raw_nn:n #1 { \use:e { \__ctex_char_raw:nn #1 } }
2638 \cs_new:Npn \__ctex_char_raw_nnn:n #1 { \use:e { \__ctex_char_raw:nnn #1 } }
2639 \cs_new:Npn \__ctex_char_raw_nnnn:n #1 { \use:e { \__ctex_char_raw:nnnn #1 } }
2640 \text_declare_purify_equivalent:Nn \CTEX@char@n { \__ctex_char_raw:n }
2641 \text_declare_purify_equivalent:Nn \CTEX@char@nn@n { \__ctex_char_raw:nn:n }
2642 \text_declare_purify_equivalent:Nn \CTEX@char@nnn@n { \__ctex_char_raw:nnn:n }
2643 \text_declare_purify_equivalent:Nn \CTEX@char@nnnn@n { \__ctex_char_raw:nnnn:n }

```

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

```

2644 \cs_new_protected:Npn \ctex_ignorespaces_case:N #1
2645 {
2646   \cs_set_protected:Npn \ctex_set_ignorespaces:
2647   { \cs_set_eq:NN \CJK@ignorespaces #1 }
2648   \ctex_set_ignorespaces:
2649 }
2650 \cs_new_protected:Npn \ctex_set_ignorespaces:
2651 { \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。

```

2652 \tl_if_exist:NF \CJKhook { \tl_new:N \CJKhook }
2653 \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。

```

2654 \cs_new_protected:Npn \ctex_punct_set:n #1
2655 {
2656   \clist_map_inline:Nn \c__ctex_punct_family_clist
2657   {
2658     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
2659     {
2660       \cs_set_eq:cc
2661       { CJKpunct@ ##1 @spaces }
2662       { c__ctex_ #1 ##1 _punct_spaces_tl }
2663     }
2664   }
2665 }
2666 \clist_const:Nn \c__ctex_punct_family_clist
2667 {
2668   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
2669   zhsongb , zhheil , zhheib , zhyoub ,
2670   zhyahei , zhyaheib , zhpfb , zhpfb
2671 }

```

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

```

2672 \cs_new_protected:Npn \ctex_punct_map_family:nn #1#2
2673 {
2674   \cs_if_free:cF { CJKpunct@ #2 @spaces }
2675   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
2676 }

```

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

```

2677 \cs_new_protected:Npn \ctex_punct_map_bfseries:nn #1#2
2678 {
2679   \clist_map_inline:nn {#1}
2680   {
2681     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
2682     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
2683   }

```

```

2684 }
2685 \cs_new_protected:Npn \ctex_punct_map_series:nnn #1#2#3
2686 {
2687   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
2688   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
2689   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
2690   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
2691   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
2692   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
2693 }

```

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

```

2694 \cs_new_protected:Npn \ctex_punct_map_itshape:nn #1#2
2695 {
2696   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
2697   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
2698   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
2699   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
2700   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
2701   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
2702 }

```

`\ctex_punct_space:nn` 定义标点的边界信息。

```

\ctexspadef
2703 \cs_new_protected:Npn \ctex_punct_space:nn #1#2
2704 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
2705 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

2706 \ctex_file_input:n { ctexspa.def }

2707 </pdfTeX>

```

15.13.3 ctex-engine-xetex.def

```

2708 <*xetex>

2709 \RequirePackage { xeCJK }
2710 \exp_args:Ne \xeCJKsetup
2711 {
2712   LoadFandol = false ,
2713   PunctStyle = \l__ctex_punct_tl
2714 }

```

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`。

```

2715 \@ifpackagelater { fontspec } { 2014/05/25 } { }
2716 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

2717 </xetex>

```

15.13.4 ctex-engine-luatex.def

```

2718 <*luatex>

```

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

```

2719 \msg_new:nnn { ctex } { luatexja-loaded }
2720 {
2721   Package~`luatexja'~cannot~be~loaded~before~`ctex'.\\
2722   Loading~file~`#1'~will~abort!
2723 }

```

```

2724 \ifpackageloaded { luatexja }
2725 { \msg_critical:nne { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
2726 {
2727   \ctex_at_begin_package:nn { luatexja }
2728   { \msg_redirect_name:nnn { ctexhook } { disable-package } { info } }
2729   \ctex_at_end_package:nn { luatexja }
2730   { \msg_redirect_name:nnn { ctexhook } { disable-package } { } }
2731   \ctex_disable_package:n { ltj-latex }
2732 }

```

CJKfntef 会载入传统 CJK 宏包，覆盖 LuaTeX-ja 与 CTeX 共用的字体族接口，使 `\setCJKmainfont` 的结果依赖宏包加载顺序。两套字体协议并不兼容，因此在 LuaTeX 后端直接禁止载入，并引导用户使用基于 LuaTeX 节点回调的 lua-ul。若 CJKfntef 已经载入，字体状态可能已经被污染，只能中止加载。

```

2733 \msg_new:nnn { ctex } { cjkfntef-luatex }
2734 {
2735   Package~`#1'~is~incompatible~with~the~ctex~LuaTeX~backend.\\
2736   Use~the~lua-ul'~package~instead;~try~
2737   \token_to_str:N \underline,~
2738   \token_to_str:N \strikeThrough,~or~
2739   \token_to_str:N \highLight.
2740 }
2741 \ifpackageloaded { CJKfntef }
2742 { \msg_critical:nnn { ctex } { cjkfntef-luatex } { CJKfntef } }
2743 {
2744   % 复用通用禁载机制，但改用包含迁移路径的专用消息。
2745   \__ctex_disable_package_aux:nnnn
2746   { ctex } { cjkfntef-luatex } { CJKfntef } { }
2747 }
2748 \RequirePackage { luatexja }
2749 \ifpackagelater { luatexja } { 2020/04/12 } { }
2750 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
2751 \RequirePackage { fontspec }
2752 \ifpackagelater { fontspec } { 2020/02/21 } { }
2753 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

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

```

2754 \ctex_at_end:n { \char_set_catcode_comment:n { \ltjlineendcomment } }

```

15.13.4.1 LuaTeX-ja 的默认设置

```

2755 \ExplSyntaxOff

```

以下设置抄录自 `lltjdefs.sty`，略有改动。

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

```

2756 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
2757 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
2758 \ltjdefcharrange{3}{%
2759   "2000-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
2760 \ltjdefcharrange{4}{%
2761   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
2762   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A95F, "A980-"ABFF, "E000-"F8FF,
2763   "FB00-"FE0F, "FE20-"FE2F, "FE70-"FEFF, "10000-"1AFF, "1B170-"1F0FF,
2764   "1F300-"1FFFF, "2000-"206F}
2765 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
2766 \ltjdefcharrange{6}{%
2767   "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
2768   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE1F, "FE30-"FE6F, "FF00-"FFEF,

```

```

2769 "1B000-"1B16F, "1F100-"1F2FF, "20000-"3FFFF, "E0100-"E01EF}
2770 \ltjdefcharrange{7}{%
2771 "1100-"11FF, "2F00-"2FFF, "3100-"318F, "31A0-"31EF, "A000-"A4CF,
2772 "A960-"A97F, "AC00-"D7FF}
2773 \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。

```

2774 \ltjdefcharrange{9}{%
2775 "00B7, "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}

```

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

```

2776 \ltjsetparameter{jacharrange={-1, -2, -3, -4, -5, +6, +7, -8, +9}}
2777 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

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

```

2778 \@ifpackagelater{luatexja}{2020/08/08}
2779 { \ltjsetparameter { autospacing, autoxspacing, differentjfm = paverage } }
2780 {
2781   \directlua{
2782     local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
2783     luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
2784   }
2785   \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
2786     xkanjiskip=.25\zw plus 1pt minus 1pt,
2787     autospacing, autoxspacing, jacharrange={-1},
2788     yalbaselineshift=\z@, yjabaselineshift=\z@,
2789     jcharwidowpenalty=500, differentjfm=paverage
2790   }
2791 }
2792 \ExplSyntaxOn

```

15.13.4.2 LuaTeX-ja 的补丁

```

2793 <@@-ctex_ltj>

```

\CTEX@alchar 分组中的字符都是 ALchar 类。

```

\CTEX@beginallalchar
\CTEX@endallalchar
2794 \cs_new_protected:Npn \CTEX@alchar #1
2795 { \CTEX@beginallalchar #1 \CTEX@endallalchar }
2796 \cs_new_protected:Npn \CTEX@beginallalchar
2797 {
2798   \group_begin:
2799   \ctex_ltj_zero_globaldefs:
2800   \ltj@allalchar
2801 }
2802 \cs_new_protected:Npn \CTEX@endallalchar
2803 { \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。

```

2804 \cs_new_protected:Npn \CTEX@chardef@text@cmd #1
2805 {
2806   \cs_set_eq:NN \@ifdefinable \@ifdefinable
2807   \tl_set:Nn \l__ctex_ltj_cmd_tl {#1}
2808   \tex_afterassignment:D \__ctex_ltj_chardef_text_cmd:
2809   \tex_chardef:D #1
2810 }
2811 \tl_new:N \l__ctex_ltj_cmd_tl
2812 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd:
2813 { \exp_after:wN \__ctex_ltj_chardef_text_cmd_aux:N \l__ctex_ltj_cmd_tl }
2814 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd_aux:N #1
2815 {
2816   \int_compare:nNf {#1} < { "80 }

```

```

2817     { \cs_set_protected:Npx #1 { \CTEX@alchar { \tex_Uchar:D #1 } } }
2818   }

```

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

```

2819 \cs_new_protected:Npn \CTEX@text@composite@x #1#2
2820 {
2821   \CTEX@beginallalchar
2822   \cs_if_exist_use:NF #1 {#2}
2823   \CTEX@endallalchar
2824 }

```

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

```

2825 \cs_new_protected:Npx \CTEX@add@unicode@accent #1#2
2826 {
2827   \CTEX@beginallalchar
2828   \exp_not:N \tl_if_blank:nTF {#2} { \tex_Uchar:D "A0 ~ } {#2}
2829   \exp_not:N \tex_Uchar:D \tex_numexpr:D #1 \scan_stop:
2830   \CTEX@endallalchar
2831 }

```

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

```

2832 \cs_new_protected:Npn \CTEX@patch@text@cmd #1
2833 {
2834   \exp_args:Nnc \__ctex_ltj_patch_text_cmd:NN #1
2835   { \UnicodeEncodingName \token_to_str:N #1 }
2836 }
2837 \cs_new_protected:Npn \__ctex_ltj_patch_text_cmd:NN #1#2
2838 {
2839   \cs_set_eq:NN \CTEX@textcmd #2
2840   \ctex_preto_cmd:NnnTF \CTEX@textcmd
2841   { \ExplSyntaxOff \makeatletter }
2842   { \CTEX@beginallalchar }
2843   {
2844     \ctex_appto_cmd:NnnTF \CTEX@textcmd
2845     { \ExplSyntaxOff \makeatletter }
2846     { \CTEX@endallalchar }
2847     { \cs_set_eq:NN #2 \CTEX@textcmd }
2848     { \ctex_patch_failure:N #1 }
2849   }
2850   { \ctex_patch_failure:N #1 }
2851 }

```

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

```

2852 \cs_new_protected:Npn \CTEX@patch@tunec
2853 {
2854   \cs_set_eq:NN \chardef@text@cmd \CTEX@chardef@text@cmd
2855   \ctex_file_input:n { tuenc.def }
2856   \cs_set_eq:NN \@text@composite@x \CTEX@text@composite@x
2857   \cs_set_eq:NN \add@unicode@accent \CTEX@add@unicode@accent
2858   \CTEX@patch@text@cmd \textasteriskcentered
2859 }
2860 \@ifpackageloaded { xunicode }
2861 { }
2862 { \CTEX@patch@tunec }

```

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

```

2863 \ctex_at_end_package:nn { xunicode }
2864 {

```

```

2865 \RequirePackage { xunicode-addon }
2866 \AtBeginUTFCommand { \CTEX@beginallalchar }
2867 \AtEndUTFCommand { \CTEX@endallalchar }
2868 }

```

对 listings 的补丁是让代码环境支持 JAchar 类。

```

2869 \ctex_at_end_package:nn { listings }
2870 { \RequirePackage { lltjp-listings } }

```

15.13.4.3 Lua 函数

直接用 token.set_lua 定义, 不是传统意义上的 TeX 宏。

```

2871 \group_begin:
2872 \char_set_catcode_space:n { 32 }
2873 \lua_now:e
2874 {

```

ctex.newluacmd 定义新的 Lua 函数。

```

2875 ctex = ctex or { }
2876 local ctex = ctex
2877 local functions = lua.get_functions_table()
2878 local new_luafunction = luatexbase.new_luafunction
2879 local create, set_lua = token.create, token.set_lua
2880 local lua_cmds = {
2881     lua_call = true ,
2882     lua_expandable_call = true ,
2883 }
2884 local newluacmd = function (name, func, ...)
2885     local tok = create(name)
2886     local id = lua_cmds[tok.cmdname] and tok.index
2887     local id = id or new_luafunction(name)
2888     set_lua(name, id, ...)
2889     functions[id] = func
2890 end
2891 ctex.newluacmd = newluacmd

2892 local ltjfont = luatexja.jfont
2893 local getattribute = tex.getattribute
2894 local tex_set, sprint = tex.set, tex.sprint
2895 local scan_int, scan_arg = token.scan_int, token.scan_argument

```

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

```

2896 newluacmd("ctex_ltj_add_kyenc:n", ltjfont.add_kyenc_list, "global", "protected")

```

\ctex_ltj_is_kenc:n 判断编码是否属于 jfont。

```

2897 newluacmd("ctex_ltj_is_kenc:n", ltjfont.is_kenc, "global")

```

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

```

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

```

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

```

2898 newluacmd("ctex_ltj_patch_external_font:n", function ()
2899     local s = scan_arg()
2900     local is_braced, is_quoted

```



```

2901     if s:sub(1,1) == '{' and s:sub(-1)=='}' then
2902         is_braced = true; s = s:sub(2,-2)
2903     end
2904     if s:sub(1,1) == '"' and s:sub(-1) == '"' then
2905         is_quoted = true; s = s:sub(2,-2)
2906     end
2907     s = s .. ltjfont.print_aftl_address()
2908     if is_braced then s = '{...s...}'
2909     elseif is_quoted then s = '"...s..."'
2910     end
2911     sprint(-2, s)
2912 end, "global")

```

`\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。

```

2913 newluacmd("ctex_ltj_set_alt_font:nnnn", function ()
2914     local b = tonumber(scan_arg())
2915     local e = tonumber(scan_arg())
2916     local alt = scan_arg()
2917     local base = scan_arg()
2918     ltjfont.set_alt_font_latex(b, e, alt, base)
2919 end, "global", "protected")

```

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

```

2920 newluacmd("ctex_ltj_clear_alt_font:n", function ()
2921     local base = scan_arg()
2922     ltjfont.clear_alt_font_latex(base)
2923 end, "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`。

```

2924 newluacmd("ctex_ltj_pickup_alt_font:nn", function ()
2925     local base = scan_arg()
2926     local size = scan_arg()
2927     ltjfont.output_alt_font_cmd("y", base)
2928     ltjfont.pickup_alt_font_a(size)
2929 end, "global", "protected")

```

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

```

2930 newluacmd("__ctex_ltj_pickup_alt_font:Nn", function ()
2931     local num = scan_int()
2932     local base = scan_arg()
2933     ltjfont.pickup_alt_font_b(num, base)
2934 end, "global", "protected")

```

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

```

2935 newluacmd("__ctex_ltj_if_alt_set:nT", ltjfont.does_alt_set, "global")

```

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

```

2936 newluacmd("ctex_ltj_zero_globaldefs:", function ()
2937     tex_set("globaldefs", 0)
2938     end, "global", "protected")
2939 }
2940 \group_end:

```

15.13.4.4 字体切换方式

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

```

2941 \cs_new_protected:Npn \ctex_ljt_select_font:
2942 {
2943   \group_begin: \exp_args:Nnc \group_end:
2944   \cs_if_exist_use:NF { \l__ctex_ljt_current_font_tl }
2945   { \tl_if_empty:NF \CJK@family { \__ctex_ljt_select_font_aux: } }
2946 }
2947 \tl_new:N \CJK@family
2948 \tl_new:N \l__ctex_ljt_current_font_tl
2949 \tl_set:Nn \l__ctex_ljt_current_font_tl
2950 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

`\__ctex_ljt_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。

```

2951 \cs_new_protected:Npn \__ctex_ljt_select_font_aux:
2952 {
2953   \group_begin:
2954   \tl_set_eq:NN \f@encoding \CJK@encoding
2955   \tl_set_eq:NN \f@family \CJK@family
2956   \cs_set_eq:NN \pickup@font \ctex_ljt_pickup_font:
2957   \__ctex_ljt_push_fontname:n { \curr@fontshape / \f@size }
2958   \ctex_ljt_pickup_font:
2959   \group_end:
2960   \font@name
2961   \__ctex_ljt_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_ljt_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。

```

2962 \cs_if_exist:cF { \l__ctex_ljt_current_font_tl }
2963 { \__ctex_ljt_select_font_aux: }
2964 }
2965 \cs_new_protected:Npn \__ctex_ljt_push_fontname:n #1
2966 {
2967   \seq_gpush:No \g__ctex_ljt_fontname_seq { \font@name }
2968   \tl_gset:Ne \font@name { \exp_not:c {#1} }
2969 }
2970 \cs_new_protected:Npn \__ctex_ljt_pop_fontname:
2971 {
2972   \seq_gpop:NNT \g__ctex_ljt_fontname_seq \l__ctex_ljt_tmp_tl
2973   { \tl_gset_eq:NN \font@name \l__ctex_ljt_tmp_tl }
2974 }
2975 \seq_new:N \g__ctex_ljt_fontname_seq

```

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

```

2976 \cs_new_protected:Npn \ctex_ljt_pickup_font:
2977 {
2978   \exp_after:wN \cs_if_exist:NF \font@name
2979   {
2980     \group_begin:
2981     \cs_set_eq:NN \extract@font \ctex_ljt_extract_font:
2982     \cs_set_eq:NN \do@subst@correction \ctex_ljt_subst_font:
2983     \define@newfont
2984     \group_end:
2985   }
2986 }
2987 \cs_new_eq:NN \pickup@font \ctex_ljt_pickup_font:

```

`\ctex_ljt_extract_font:` LuaTeX-j_a 的 `\globaljfont` 在 `luatexja-core` 中定义:

```

%%\font\CS={...:.;jfm=metric;...}, \global\font
\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。

```

2988 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
2989 {
2990   \get@external@font
2991   \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
2992   {
2993     \tl_set:Nx \external@font
2994     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
2995   }
2996   \exp_after:wN \global\font \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。

```

2997   \font@name
2998   \ctex_ltj_use_jfont:
2999   \use:c { \f@encoding + \f@family }
3000   \use:c { \curr@fontshape }
3001 }

```

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

```

3002 \cs_new_protected_nopar:Npn \ctex_ltj_use_jfont:
3003 { \tex_setfontid:D \ltj@curjfont }

```

\ctex_ltj_subst_font: \do@subst@correction 在设置通过 sub 或者 ssup 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。

```

3004 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:

```

```

3005 {
3006   \ctex_ltj_if_alterate_shape_exist:nF { \curr@fontshape }
3007   {
3008     \group_begin:
3009     \tl_set_eq:NN \CJK@family \f@family
3010     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
3011     {
3012       \cs_gset_protected_nopar:Npx \subst@correction
3013       {
3014         \cs_new_eq:NN
3015         \exp_not:c { \l__ctex_ltj_current_font_tl }
3016         \font@name
3017       }
3018       \group_insert_after:N \group_insert_after:N
3019       \group_insert_after:N \subst@correction
3020     }
3021     \group_end:
3022   }
3023 }

```

`\ctex_ltj_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。

```

3024 \prg_new_conditional:Npnn \ctex_ltj_if_alterate_shape_exist:n #1 { T , F , TF }
3025 {
3026   \__ctex_ltj_if_alt_set:nT {#1} { \prg_return_true: \use_none:n }
3027   \prg_return_false:
3028 }

```

`\__ctex_ltj_patch_external_font:w` 3029 `\cs_new:Npn \__ctex_ltj_patch_external_font:w #1 ~ at`
 3030 `{ \ctex_ltj_patch_external_font:n {#1} ~ at }`

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

```

3031 \cs_new_protected:Npn \ctex_ltj_select_alterate_font:
3032 {
3033   \ctex_ltj_if_alterate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
3034   {
3035     \ctex_ltj_pickup_alt_font:nn
3036     { \l__ctex_ltj_current_shape_tl } { \f@size }
3037   }
3038 }
3039 \tl_new:N \l__ctex_ltj_current_shape_tl
3040 \tl_set:Nn \l__ctex_ltj_current_shape_tl
3041 { \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。

```

3042 \cs_new_protected:Npn \ltj@pickup@altfont@auxy #1
3043 {
3044   \cs_if_exist:cF { #1/\f@size }
3045   {
3046     \group_begin:
3047     \use:e { \exp_not:N \split@name #1 / \f@size } \@nil
3048     \__ctex_ltj_push_fontname:n { \curr@fontshape / \f@size }
3049     \ctex_ltj_pickup_font:
3050     \group_end:
3051     \__ctex_ltj_pop_fontname:
3052   }
3053 }

```

`\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。

```

3054 \cs_new_protected:Npn \ltj@pickup@altfont@copy #1#2
3055 {

```

```

3056 \ltj@@getjfontnumber #1
3057 \__ctex_ltj_pickup_alt_font:Nn \ltj@tempcntc {#2}
3058 }

```

15.13.4.5 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`。

```

3059 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
3060 {
3061   \ctex_ltj_is_kenc:n { \__ctex_ltj_ltj_if_jfont:w #1 / \q_stop }
3062   \legacy_if:nTF { in@ }
3063 }
3064 \cs_new:Npn \__ctex_ltj_ltj_if_jfont:w #1 / #2 \q_stop
3065 {#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。

```

3066 \cs_new:Npn \ctex_ltj_if_jfont_math:nTF #1
3067 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
3068 \group_begin:
3069   \char_set_catcode_other:N M
3070   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
3071     { \ctex_ltj_if_jfont:nTF {#3} }
3072 \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`。

```

3073 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
3074 {
3075   \exp_args:No \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
3076   { \ctex_ltj_get_and_define_fonts_ja:nN }
3077   { \ctex_ltj_get_and_define_fonts_al:nN }
3078   {#1} #2
3079 }
3080 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
3081 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
3082 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
3083 {
3084   \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
3085   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
3086   \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
3087   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
3088   \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
3089   \ctex_ltj_pickup_font:
3090   \tl_put_right:Ne \math@fonts
3091   {
3092     \ltj@setpar@global
3093     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
3094     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
3095     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
3096   }
3097 }

```

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

`\ctex_ltj_use_math_group:Nn`

```

3098 \cs_new_protected:Npn \ctex_ltj_use_math_group:Nn #1#2
3099 {
3100     \mode_if_math:T
3101     {
3102         \math@bgroup
3103         \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
3104         \ctex_ltj_math_group_hook:
3105         \ctex_ltj_if_jfont_math:NTF #1
3106         { \jfam } { \mathgroup } #2 \scan_stop:
3107         \math@egroup
3108     }
3109 }
3110 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
3111 \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。

```

3112 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
3113 {
3114     \group_begin:
3115     \cs_set_protected:Npn #1 ##1##2##3
3116     { \ltjsetmathletter { ##1 } }
3117     #2
3118     \group_end:
3119 }
3120 \ctex_at_end_package:nn { unicode-math }
3121 {
3122     \cs_if_exist:NTF \um_input_math_symbol_table:
3123     {
3124         \ctex_ltj_set_math_letter:NN
3125         \um_sym:nnn
3126         \um_input_math_symbol_table:
3127     }
3128     {
3129         \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
3130         \cs_set_protected:Npn \ctex_ltj_math_group_hook:
3131         { \__um_switchto_literal: }
3132         \ctex_ltj_set_math_letter:NN
3133         \__um_sym:nnn
3134         \__um_input_math_symbol_table:
3135     }
3136 }

```

15.13.4.6 字体族的定义与使用

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

`\l__ctex_ltj_jfm_tl`

`\l__ctex_ltj_redirect_jfm_prop`

```

3137 \cs_new_protected:Npn \ctex_set_jfm:n #1
3138 {
3139     \prop_get:NnNF \l__ctex_ltj_redirect_jfm_prop {#1} \l__ctex_ltj_jfm_tl
3140     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
3141 }
3142 \cs_generate_variant:Nn \ctex_set_jfm:n { o }
3143 \prop_new:N \l__ctex_ltj_redirect_jfm_prop
3144 \prop_set_from_keyval:Nn \l__ctex_ltj_redirect_jfm_prop
3145 {
3146     plain      = mono ,
3147     quanjiao   = zh_CN / quanjiao ,
3148     banjiao    = zh_CN / banjiao ,
3149     kaiming     = zh_CN / kaiming
3150 }

```

```

3151 \keys_define:nn { ctex_ltj / fontspec }
3152 {
3153     JFM .code:n = \ctex_set_jfm:n {#1} ,
3154     JFM .value_required:n = true
3155 }
3156 \tl_new:N \l__ctex_ltj_jfm_tl

```

注意这里的 `@@=ctex_ltj`。

```

3157 \ctex_set_jfm:o { \l__ctex_punct_tl }

```

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

```

3158 \tl_const:Nn \CJK@encoding { LTJY3 }
3159 \DeclareFontEncoding { \CJK@encoding } { } { }
3160 \use:e
3161 {
3162     \exp_not:N \DeclareFontSubstitution
3163     { \CJK@encoding } { song } { \mddefault } { \shapedefault }
3164 }
3165 \ctex_ltj_add_kyenc:n { \CJK@encoding }
3166 \DeclareFontFamily { \CJK@encoding } { song } { }
3167 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \shapedefault }
3168 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
3169 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
3170 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
3171 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
3172 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
3173 { \CJK@encoding } { song } { \mddefault } { \shapedefault }
3174 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
3175 { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
3176 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
3177 \jfam \c__ctex_ltj_math_fam_int

```

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

3178 \newfontfeature { CID } { cid = #1 }
3179 \newfontfeature { JFM } { jfm = #1 }
3180 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

3181 \keys_define:nn { fontspec-prepare-external }
3182 {
3183     NoEmbed .code:n =
3184     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
3185 }
3186 \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。

```

3187 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
3188 {
3189     \group_begin:
3190     \clist_clear:N \l__ctex_ltj_char_range_clist
3191     \prop_clear:N \l__ctex_ltj_alternate_prop
3192     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
3193     \keys_set_known:nnn { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
3194     \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
3195     \ctex_ltj_set_alternate_family:nnF {#1} {#3}
3196     {
3197         \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}

```

```

3198     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
3199     {#1} { \l__ctex_ltj_font_options_clist }
3200     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
3201     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
3202     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
3203   }
3204   \group_end:
3205 }
3206 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
3207 \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。

```

3208 \cs_new_protected:Npn \__ctex_ltj_use_global_options:N #1
3209 {
3210   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
3211   \clist_put_left:Ne #1
3212   { NFSSEncoding = \CJK@encoding , JFM = \l__ctex_ltj_jfm_tl }
3213 }

```

\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_name_prop
\g__ctex_ltj_family_font_name_prop
\g__ctex_ltj_family_font_options_prop
3214 \prop_new:N \g__ctex_ltj_family_name_prop
3215 \prop_new:N \g__ctex_ltj_family_font_name_prop
3216 \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。

```

3217 \cs_new_protected:Npn \__ctex_ltj_check_family:n #1
3218 {
3219   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
3220   {
3221     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
3222     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
3223     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
3224     {
3225       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
3226       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
3227       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
3228       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
3229     }
3230     \msg_warning:nnee { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
3231   }
3232 }
3233 \tl_new:N \l__ctex_ltj_tmp_tl
3234 \msg_new:nnn { ctex } { redefine-family }
3235 { 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。

```

3236 \cs_new_protected:Npn \__ctex_ltj_gset_family_cs:nn #1#2
3237 {
3238   \cs_gset_protected:cpx { \__ctex_ltj_family_csname:n {#1} }
3239   {
3240     \group_begin:
3241     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
3242     \exp_not:n { \fontspec_gset_family:Nnn \g__ctex_ltj_fontspec_family_tl }
3243     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
3244     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
3245     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
3246     \__ctex_ltj_set_alternate_family:n {#1}
3247     \group_end:
3248   }
3249 }

```



```

3250 \tl_new:N \l__ctex_ltj_base_family_tl
3251 \tl_new:N \g__ctex_ltj_fontspec_family_tl
3252 \cs_new:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
3253 \cs_new_protected:Npn \__ctex_ltj_set_altername_family:n #1
3254 {
3255   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
3256   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
3257   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
3258   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n {#1} }
3259 }
3260 \cs_new:Npn \__ctex_ltj_altername_cs:n #1 { ctex_ltj/altername_family/#1 }

```

\CJKfamily 切换字体。

```

3261 \NewDocumentCommand \CJKfamily { m }
3262 { \ctex_ltj_switch_family:e {#1} \tex_ignorespaces:D }
3263 \cs_new_protected:Npn \ctex_ltj_switch_family:n #1
3264 {
3265   \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
3266   {
3267     \tl_set:Nn \l__ctex_ltj_family_tl {#1}
3268     \selectfont
3269   }
3270   { \__ctex_ltj_family_unknown_warning:n {#1} }
3271 }
3272 \tl_new:N \l__ctex_ltj_family_tl
3273 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { e }

```

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

```

3274 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
3275 {
3276   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
3277   { \prg_return_true: }
3278   {
3279     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
3280     {
3281       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
3282       \prg_return_true:
3283     }
3284     { \prg_return_false: }
3285   }
3286 }
3287 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { e } { T , F , TF }

```

```

\__ctex_ltj_family_unknown_warning:n 3288 \cs_new_protected:Npn \__ctex_ltj_family_unknown_warning:n #1
3289 {
3290   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
3291   {
3292     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
3293     {
3294       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
3295       \msg_warning:nnn { ctex } { family-unknown } {#1}
3296     }
3297   }
3298 }
3299 \seq_new:N \g__ctex_ltj_unknown_family_seq
3300 \msg_new:nnn { ctex } { family-unknown }
3301 {
3302   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
3303   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
3304 }
3305 \cs_new:Npn \__ctex_ltj_msg_def_family_map:n #1
3306 {
3307   \str_case_e:nnF {#1}
3308   {
3309     \CJKrmdefault { \token_to_str:N \setCJKmainfont }

```

```

3310      \CJKsfdefault { \token_to_str:N \setCJKsansfont }
3311      \CJKttdefault { \token_to_str:N \setCJKmonofont }
3312    }
3313    { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
3314    [...] \{...\}
3315  }
3316 \cs_new:Npn \__ctex_ltj_msg_family_map:n #1
3317 {
3318   \str_case:e:nnF {#1}
3319   {
3320     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
3321     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
3322     \CJKttdefault { \token_to_str:N \CJKttdefault }
3323   }
3324   {#1}
3325 }

\ctex_ltj_fontspec:nn 3326 \cs_new_protected:Npn \ctex_ltj_fontspec:nn #1#2
3327 {
3328   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
3329   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
3330   { \ctex_ltj_switch_family:e { \l_ctex_ltj_family_tl } }
3331   {
3332     \int_gincr:N \g__ctex_ltj_family_int
3333     \__ctex_ltj_fontspec:enn
3334     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
3335     {#1} {#2}
3336   }
3337 }
3338 \cs_new_protected:Npn \ctex_ltj_fontspec:ee #1#2
3339 { \use:e { \ctex_ltj_fontspec:nn {#1} {#2} } }
3340 \cs_new_protected:Npn \__ctex_ltj_fontspec:nnn #1#2#3
3341 {
3342   \bool_if:NT \l__ctex_ltj_add_alterate_bool
3343   {
3344     \cs_if_free:cF
3345     { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
3346     {
3347       \cs_gset_eq:cc
3348       { \__ctex_ltj_alterate_cs:n { reset / #1 } }
3349       { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
3350       \cs_gset_eq:cc
3351       { \__ctex_ltj_alterate_cs:n { clear / #1 } }
3352       { \__ctex_ltj_alterate_cs:n { clear / \l_ctex_ltj_family_tl } }
3353     }
3354     \bool_set_false:N \l__ctex_ltj_add_alterate_bool
3355   }
3356   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
3357   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
3358   \ctex_ltj_switch_family:n {#1}
3359 }
3360 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { e }
3361 \prop_new:N \g__ctex_ltj_fontspec_prop

\ctex_ltj_add_font_features:n 3362 \cs_new_protected:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 3363 { \ctex_ltj_add_font_features:en { \l_ctex_ltj_family_tl } {#1} }
3364 \cs_new_protected:Npn \ctex_ltj_add_font_features:nn #1#2
3365 {
3366   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
3367   {#1} \l__ctex_ltj_tmp_tl
3368   {
3369     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
3370     {#1} \l__ctex_ltj_font_options_clist
3371     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
3372     \bool_set_true:N \l__ctex_ltj_add_alterate_bool
3373     \ctex_ltj_fontspec:ee
3374     { \exp_not:o { \l__ctex_ltj_font_options_clist } }

```

```

3375         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
3376     }
3377     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
3378 }
3379 \bool_new:N \l__ctex_ltj_add_alterate_bool
3380 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { e }
3381 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { e }
3382 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
3383 {
3384     \token_to_str:N \addCJKfontfeature (s)~ignored.\
3385     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
3386 }

```

`\__ctex_ltj_pass_args:nnnn` 为了支持字体属性可选项在前在后两种语法, 给出两个辅助工具, 自带展开功能。

```

3387 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
3388 {
3389     \tl_if_novalue:nTF {#2}
3390     { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
3391     {
3392         \use:e { #1 {#2} {#3} }
3393         #4
3394     }
3395 }
3396 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
3397 {
3398     \use:e { #1 {#4} {#2} }
3399     #3
3400 }

\setCJKfamilyfont 3401 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 3402 {
\CJKfontspec 3403     \__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 3404     { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
3405     { }
3406 }
3407 \msg_new:nnn { ctex } { command-already-defined }
3408 { Control~sequence~#1~is~already~defined. }
3409 \NewDocumentCommand \newCJKfontfamily { o m o m }
3410 {
3411     \tl_set:Nx \l__ctex_ltj_tmp_tl
3412     { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
3413     \cs_if_exist:NTF #2
3414     { \msg_error:nne { ctex } { command-already-defined }
3415       { \token_to_str:N #2 } }
3416     {
3417         \cs_set_protected:Npx #2
3418         { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
3419     }
3420     \__ctex_ltj_pass_args:nnnn
3421     { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
3422     { }
3423 }
3424 \NewDocumentCommand \CJKfontspec { o m }
3425 {
3426     \__ctex_ltj_pass_args:nnnn
3427     { \ctex_ltj_fontspec:nn {#1} {#2}
3428       { \tex_ignorespaces:D }
3429     }
3430 \NewDocumentCommand \addCJKfontfeatures { m }
3431 {
3432     \ctex_ltj_add_font_features:e {#1}
3433     \tex_ignorespaces:D
3434 }
3435 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

```

\setCJKmainfont 3436 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures

```

```

3437 {
3438   \__ctex_ltj_pass_args:nnnn
3439   { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
3440   { \normalfont }
3441 }
3442 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
3443 \NewDocumentCommand \setCJKsansfont { o m }
3444 {
3445   \__ctex_ltj_pass_args:nnnn
3446   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
3447   { \normalfont }
3448 }
3449 \NewDocumentCommand \setCJKmonofont { o m }
3450 {
3451   \__ctex_ltj_pass_args:nnnn
3452   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
3453   { \normalfont }
3454 }
3455 \NewDocumentCommand \setCJKmathfont { o m }
3456 {
3457   \__ctex_ltj_pass_args:nnnn
3458   { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
3459   { }
3460 }
3461 \NewDocumentCommand \defaultCJKfontfeatures { m }
3462 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
3463 \clist_new:N \g__ctex_ltj_default_features_clist
3464 \@onlypreamble \setCJKmainfont
3465 \@onlypreamble \setCJKsansfont
3466 \@onlypreamble \setCJKmonofont
3467 \@onlypreamble \setCJKmathfont
3468 \@onlypreamble \setCJKromanfont
3469 \@onlypreamble \defaultCJKfontfeatures

```

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

```

3470 \cs_new_protected:Npn \ctex_ltj_ensure_default_family:
3471 {
3472   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
3473   {
3474     \ctex_ltj_family_if_exist:eNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
3475     {
3476       \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
3477       { \use:n }
3478       {
3479         \ctex_ltj_family_if_exist:eNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
3480         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
3481         { \use:n }
3482       }
3483       {
3484         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
3485         {
3486           \prop_map_break:n
3487           { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
3488         }
3489       }
3490     }
3491     \normalfont
3492     \ctex_ltj_update_mathfont:
3493   }
3494 }

```

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

```

3495 \cs_new_protected:Npn \ctex_ltj_update_mathfont:
3496 {

```

```

3497 \ctex_ltj_family_if_exist:eNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
3498 { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
3499 {
3500 \ctex_ltj_family_if_exist:eNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
3501 { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
3502 }
3503 }
3504 \cs_new_protected:Npn \ctex_ltj_update_mathfont:n #1
3505 {
3506 \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
3507 \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
3508 { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
3509 \cs_if_free:cTF
3510 { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\shapedefault }
3511 {
3512 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
3513 { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
3514 }
3515 {
3516 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
3517 { \c__ctex_ltj_math_family_tl } { \bfdefault } { \shapedefault }
3518 }
3519 }

```

15.13.4.7 替代字体的设置

AlternateFont 设置替代字体的选项。
CharRange

```

3520 \keys_define:nn { ctex_ltj / fontspec }
3521 {
3522 AlternateFont .code:n = \ctex_ltj_set_alternate_prop:n {#1} ,
3523 AlternateFont .value_required:n = true ,
3524 CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
3525 CharRange .value_required:n = true
3526 }

```

`\ctex_ltj_set_alternate_prop:n` 保存替代字体序列。

```

3527 \cs_new_protected:Npn \ctex_ltj_set_alternate_prop:n #1
3528 { \clist_map_function:nN {#1} \__ctex_ltj_push_alternate_prop:n }
3529 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:n #1
3530 {
3531 \clist_set:Nx \l__ctex_ltj_tmp_clist { \tl_head:n {#1} }
3532 \tl_remove_all:Nn \l__ctex_ltj_tmp_clist { ~ }
3533 \exp_args:No \__ctex_ltj_push_alternate_prop:nn
3534 { \l__ctex_ltj_tmp_clist } {#1}
3535 }
3536 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:nn #1
3537 {
3538 \prop_remove:Nn \l__ctex_ltj_alternate_prop {#1}
3539 \prop_put:Nnn \l__ctex_ltj_alternate_prop {#1}
3540 }
3541 \clist_new:N \l__ctex_ltj_tmp_clist
3542 \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。

```

3543 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnF
3544 {
3545 \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
3546 { \__ctex_ltj_set_family_aux:nnn }
3547 { \__ctex_ltj_set_alternate_family_aux:nnn }
3548 }
3549 \cs_new_protected:Npn \__ctex_ltj_set_family_aux:nnn #1#2#3
3550 {

```

```

3551 \__ctex_ltj_check_family:n {#1}
3552 \prop_if_empty:NF \l__ctex_ltj_altername_prop
3553 { \ctex_ltj_save_altername_seq:cn { \__ctex_ltj_altername_cs:n {#1} } {#2} }
3554 #3
3555 }
3556 \cs_new_protected:Npn \__ctex_ltj_set_altername_family_aux:nnn #1#2#3
3557 { \ctex_ltj_set_altername_family:nn {#1} {#2} }

```

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

```

3558 \cs_new_protected:Npn \ctex_ltj_save_altername_seq:Nn #1#2
3559 {
3560   \prop_map_inline:Nn \l__ctex_ltj_altername_prop
3561   { \__ctex_ltj_save_altername_auxi:w ##2 { } \q_mark #1 {#2} }
3562 }
3563 \cs_new_protected:Npn \__ctex_ltj_save_altername_auxi:w #1#2#
3564 {
3565   \tl_if_blank:nTF {#2}
3566   { \__ctex_ltj_save_altername_auxii:w {#1} }
3567   { \__ctex_ltj_save_altername_auxii:w {#1} {#2} }
3568 }
3569 \cs_new_protected:Npn \__ctex_ltj_save_altername_auxii:w #1#2#3 #4 \q_mark #5#6
3570 {
3571   \clist_set:Nn \l__ctex_ltj_char_range_clist {#1}
3572   \clist_set:Nn \l__ctex_ltj_altername_options_clist {#3}
3573   \__ctex_ltj_use_global_options:N \l__ctex_ltj_altername_options_clist
3574   \tl_if_blank:nTF {#2}
3575   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#6} }
3576   {
3577     \tl_set:Nn \l__ctex_ltj_tmp_tl { \tl_trim_spaces:n {#2} }
3578     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#6}
3579   }
3580   \use:e
3581   {
3582     \ctex_ltj_save_altername_family:Nnnn \exp_not:N #5
3583     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
3584     { \exp_not:o { \l__ctex_ltj_altername_options_clist } }
3585     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
3586   }
3587 }
3588 \clist_new:N \l__ctex_ltj_altername_options_clist
3589 \cs_generate_variant:Nn \ctex_ltj_save_altername_seq:Nn { c }

```

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

```

3590 \cs_new_protected:Npn \ctex_ltj_set_altername_family:nn #1#2
3591 {
3592   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
3593   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
3594   \ctex_ltj_set_altername_family:coonn
3595   { \__ctex_ltj_altername_cs:n {#1} }
3596   { \l__ctex_ltj_char_range_clist }
3597   { \l__ctex_ltj_font_options_clist } {#2} {#1}
3598 }
3599 \cs_new_protected:Npn \ctex_ltj_set_altername_family:Nnnnn #1#2#3#4#5
3600 {
3601   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
3602   { \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} }
3603   \ctex_ltj_save_altername_family:Nnnn #1 {#2} {#3} {#4}
3604 }
3605 \cs_generate_variant:Nn \ctex_ltj_set_altername_family:Nnnnn { coo }

```

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

```

3606 \cs_new_protected:Npn \ctex_ltj_save_altername_family:Nnnn #1#2#3#4

```

```

3607 {
3608   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
3609   \cs_gset_protected:Npx #1
3610     { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
3611 }

```

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

```

3612 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
3613 {
3614   \group_begin:
3615   \cs_set_eq:NN \CJKfamily \use_none:n
3616   \ctex_ltj_swap_cs:NN
3617     \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
3618     \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
3619     \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
3620   \group_end:
3621 }
3622 \tl_new:N \l__ctex_ltj_alternate_family_tl

```

\ctex_ltj_swap_cs:NN 交换两个控制序列的意义。

```

3623 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
3624 {
3625   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
3626   \cs_set_eq:NN #1 #2
3627   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
3628   \cs_undefine:N \__ctex_ltj_tmp:w
3629 }

```

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

```

3630 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
3631 \cs_new_protected:Npn \__ctex_ltj_update_family_uid:N #1
3632 {
3633   \int_gincr:N \g__ctex_ltj_family_int
3634   \clist_put_right:Ne #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
3635 }
3636 \int_new:N \g__ctex_ltj_family_int

```

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

```

3637 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
3638 {
3639   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
3640   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
3641     { \l__ctex_ltj_base_family_tl } {#3} {#4} {#2} {#3} {#4}
3642 }

```

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

```

3643 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
3644 {
3645   \clist_map_inline:Nn #1
3646     {
3647       \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
3648       {
3649         \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
3650         \l__ctex_ltj_char_range_tl
3651       }

```

```

3652         { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
3653     }
3654     \__ctex_ltj_save_alternate_shape:cn
3655     { \__ctex_ltj_alternate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
3656     { \ctex_ltj_clear_alt_font:n { \CJK@encoding/#2/#3/#4 } }
3657 }

```

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

```

3658 \NewDocumentCommand \ctex_ltj_set_alternate_shape:nnn
3659 { m m > { \SplitArgument { 1 } { -> } } m }
3660 { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
3661 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
3662 {
3663     \ctex_ltj_set_alternate_shape:e
3664     {
3665         \__ctex_ltj_range_normalization:nn {#3} {#4}
3666         { \CJK@encoding / \exp_not:n {#2} }
3667         { \CJK@encoding / \exp_not:n {#1} }
3668     }
3669 }
3670 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:n #1
3671 {
3672     \ctex_ltj_set_alt_font:nnnn #1
3673     \__ctex_ltj_save_alternate_shape:cn
3674     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
3675     { \ctex_ltj_set_alt_font:nnnn #1 }
3676 }
3677 \cs_generate_variant:Nn \ctex_ltj_set_alternate_shape:n { e }

```

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

```

3678 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
3679 {
3680     \tl_map_inline:Nn #3
3681     {
3682         \ctex_ltj_set_alternate_shape:n
3683         {
3684             ##1
3685             { \CJK@encoding/#2 }
3686             { \CJK@encoding/#1 }
3687         }
3688     }
3689 }

```

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

```

3690 \cs_new_protected:Npn \__ctex_ltj_save_alternate_shape:Nn #1#2
3691 {
3692     \group_begin:
3693     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
3694     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
3695     \cs_gset_protected:Npx #1 { \exp_not:o {#1} #2 }
3696     \group_end:
3697 }
3698 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nn { c }

```

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

```

3699 \ctex_define:n

```



```

3700 {
3701   clearalternatefont .code:n =
3702     { \clist_map_function:eN {#1} \ctex_ltj_clear_altername_font:n } ,
3703   resetalternatefont .code:n =
3704     { \clist_map_function:eN {#1} \ctex_ltj_reset_altername_font:n } ,
3705   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
3706   resetalternatefont .default:n = \l_ctex_ltj_family_tl
3707 }
3708 \cs_new_protected:Npn \ctex_ltj_clear_altername_font:n #1
3709 {
3710   \group_begin:
3711     \ctex_ltj_family_if_exist:eNTF {#1} \l__ctex_ltj_base_family_tl
3712     {
3713       \cs_if_exist_use:cT
3714       { \__ctex_ltj_altername_cs:n { clear / #1 } }
3715       {
3716         \prop_gput:Nno \g__ctex_ltj_reset_altername_prop
3717           {#1} { \l__ctex_ltj_base_family_tl }
3718         \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
3719         \selectfont
3720       }
3721     }
3722     { \__ctex_ltj_family_unknown_warning:n {#1} }
3723   \group_end:
3724 }
3725 \cs_new_protected:Npn \ctex_ltj_reset_altername_font:n #1
3726 {
3727   \group_begin:
3728     \prop_gpop:NnNT \g__ctex_ltj_reset_altername_prop {#1} \CJK@family
3729     {
3730       \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
3731       \use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
3732       \selectfont
3733     }
3734   \group_end:
3735 }
3736 \prop_new:N \g__ctex_ltj_reset_altername_prop
3737 \cs_generate_variant:Nn \clist_map_function:nN { e }

```

declarecharrange 预先声明字符范围。

```

3738 \ctex_define:n
3739 {
3740   declarecharrange .code:n = \ctex_ltj_declare_char_range:e {#1} ,
3741   declarecharrange .value_required:n = true
3742 }
3743 \cs_new_protected:Npn \ctex_ltj_declare_char_range:n #1
3744 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
3745 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { e }
3746 \cs_new_protected:Npn \__ctex_ltj_declare_char_range:nn #1
3747 { \tl_trim_spaces_apply:nN {#1} \ctex_ltj_declare_char_range:nn }

```

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

`\g__ctex_ltj_char_range_prop`

```

3748 \cs_new_protected:Npn \ctex_ltj_declare_char_range:nn #1#2
3749 {
3750   \tl_clear:N \l__ctex_ltj_char_range_tl
3751   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
3752   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
3753   \ctex_ltj_def_char_range_key:n {#1}
3754   \tl_clear:N \l__ctex_ltj_char_range_tl
3755 }
3756 \tl_new:N \l__ctex_ltj_char_range_tl
3757 \prop_new:N \g__ctex_ltj_char_range_prop

```

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

```

3758 \NewDocumentCommand \ctex_ltj_save_char_range:n
3759 { > { \SplitArgument { 1 } { -> } } m }
3760 { \ctex_ltj_save_char_range:nn #1 }
3761 \cs_new_protected:Npn \ctex_ltj_save_char_range:nn #1#2
3762 {
3763   \tl_put_right:Ne \l__ctex_ltj_char_range_tl
3764     { { \__ctex_ltj_range_normalization:nn {#1} {#2} } }
3765 }
3766 \cs_new:Npn \__ctex_ltj_range_normalization:nn #1#2
3767 {
3768   \tl_if_novalue:nTF {#2}
3769   {
3770     { \int_eval:n {#1} }
3771     { \int_eval:n {#1} }
3772   }
3773   {
3774     { \int_eval:n { \tl_if_blank:nTF {#1} { \c__ctex_ltj_range_min_int } {#1} } }
3775     { \int_eval:n { \tl_if_blank:nTF {#2} { \c__ctex_ltj_range_max_int } {#2} } }
3776   }
3777 }
3778 \int_const:Nn \c__ctex_ltj_range_min_int { "80 }
3779 \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。

```

3780 \cs_new_protected:Npn \ctex_ltj_def_char_range_key:n #1
3781 {
3782   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
3783   {
3784     \keys_define:nn { ctex_ltj / fontspec }
3785       { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
3786   }
3787 }

```

`\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。

```

3788 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
3789 {
3790   \tl_if_blank:nTF {#2}
3791   { \clist_set:Nn \l__ctex_ltj_char_range_clist {#1} }
3792   { \__ctex_ltj_push_alternate_prop:nn {#1} { {#1} #2 } }
3793 }

```

15.13.4.8 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
}

3794 \AtBeginDocument
3795 {

```

```

3796 \ctex_appto_cmd:NnnTF \verbatim@font
3797 { \char_set_catcode_letter:n { 64 } }
3798 { \CTEX@verbatim@font@hook }
3799 { }
3800 { \ctex_patch_failure:N \verbatim@font }
3801 }
3802 \cs_new_protected:Npn \CTEX@verbatim@font@hook
3803 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

```

LuaTeX-ja 的 `lltjcore.sty` 会将 $\text{\LaTeX}$ 的 `\verb` 中 `\null` (即 `\hbox{}`) 替换为 `\vadjust{}`, 因为空 `\hbox{}` 会阻断 `xkanjiskip` 的自动插入。由于 `ctex` 禁用了 `ltj-latex`, 这里需要自行复制该修复。

```

3804 \@namedef { ver@lltjcore.sty } { }
3805 \if@compatibility \else
3806 \def \verb
3807 {
3808 \relax \ifmmode \hbox \else \leavevmode \vadjust { } \fi
3809 \bgroup
3810 \verb@eol@error \let \do \@makeother \dospecials
3811 \verbatim@font \@noligs
3812 \language \l@nohyphenation
3813 \@ifstar \sverb \@verb
3814 }
3815 \fi
3816 \patchcmd { \do@noligs } { \kern \z@ } { \vadjust { } } { } { }

```

`\@italiccorr` $\text{\LaTeX}$ 的倾斜校正也要重新定义。

```

3817 \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`。

```

3818 \cs_new_protected:Npn \ctex_ltj_set_kanjiskip:N
3819 { \ltj@setpar@global \ltjsetkanjiskip }
3820 \cs_new_protected:Npn \ctex_ltj_set_xkanjiskip:N
3821 { \ltj@setpar@global \ltjsetxkanjiskip }

3822 <@@=ctex>
3823 </luatex>

```

15.13.5 ctex-engine-uptex.def

```

3824 <*uptex|aptex>

```

按 CJK 的命名习惯模拟 `\CJKfamily`。

```

3825 \NewDocumentCommand \CJKfamily { m }
3826 { \kanjifamily {#1} \selectfont }

```

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

```

3827 \DeclareErrorKanjiFont {JY2}{zhrm}{m}{n}{10}
3828 \DeclareKanjiSubstitution {JY2}{zhrm}{m}{n}
3829 \DeclareKanjiSubstitution {JT2}{zhrm}{m}{n}
3830 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
3831 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
3832 \jfam \symmincho

```

`\em` 取消 $\text{up\LaTeX}$ 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。 $\text{up\LaTeX}$ 2016/05/07u00 的定义有所变化, 这一行为可以由用户通过 `platexrelease` 包改变, 需要分支处理。

```

3833 \ctex_patch_cmd_once:NnnnTF \em

```

```

3834 { \ExplSyntaxOff }
3835 { \eminnershape \else \gtfamily \itshape }
3836 { \eminnershape \else \itshape }
3837 { }
3838 {
3839   \ctex_patch_cmd:Nnn \em
3840     { \mcfamily \upshape \else \gtfamily \itshape }
3841     { \eminnershape \else \itshape }
3842 }
3843 \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。

```

3844 \cs_new_protected:Npn \ctex_set_upfamily:nnn #1#2#3
3845 {
3846   \DeclareKanjiFamily{JY2}{#1}{ }
3847   \DeclareKanjiFamily{JT2}{#1}{ }
3848   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{ }
3849   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{ }
3850   \tl_if_empty:nF {#3}
3851   {
3852     \DeclareFontShape{JY2}{#1}{b}{n}{<->~ #3-h}{ }
3853     \DeclareFontShape{JT2}{#1}{b}{n}{<->~ #3-v}{ }
3854     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{ }
3855     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{ }
3856   }
3857 }

```

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

```

3858 \cs_new_protected:Npn \ctex_set_upmap:nnn #1#2#3
3859 {
3860   \ctex_set_zhmap:n
3861   {
3862     \special { pdf:mapline~#1-h~UniGB-UTF16-H~#2 }
3863     \special { pdf:mapline~#1-v~UniGB-UTF16-V~#2 }
3864     \tl_if_empty:nF {#3}
3865     {
3866       \special { pdf:mapline~#1b-h~UniGB-UTF16-H~#3 }
3867       \special { pdf:mapline~#1b-v~UniGB-UTF16-V~#3 }
3868     }
3869   }
3870 }

```

`\ctex_set_upmap_unicode:nnn` 设置 $\text{\TeX}$ 字体映射, 使用 unicode CMap。参数同上。

```

3871 \cs_new_protected:Npn \ctex_set_upmap_unicode:nnn #1#2#3
3872 {
3873   \ctex_set_zhmap:n
3874   {
3875     \special { pdf:mapline~#1-h~unicode~#2 }
3876     \special { pdf:mapline~#1-v~unicode~#2 }
3877     \tl_if_empty:nF {#3}
3878     {
3879       \special { pdf:mapline~#1b-h~unicode~#3 }
3880       \special { pdf:mapline~#1b-v~unicode~#3 }
3881     }
3882   }
3883 }

```

`\ctex_set_upfonts:nnnnnn` 设置 $\text{\LaTeX}$ 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。

```
3884 \cs_new_protected:Npn \ctex_set_upfonts:nnnnnn #1#2#3#4#5#6
3885 {
3886   \ctex_set_upmap:nnn { upserif } {#1} {#2}
3887   \ctex_set_upmap:nnn { upserifit } {#3} {}
3888   \ctex_set_upmap:nnn { upsans } {#4} {#5}
3889   \ctex_set_upmap:nnn { upmono } {#6} {}
3890 }
```

以下命令只能在导言区使用。

```
3891 \@onlypreamble \ctex_set_upfamily:nnn
3892 \@onlypreamble \ctex_set_upmap:nnn
3893 \@onlypreamble \ctex_set_upmap_unicode:nnn
3894 \@onlypreamble \ctex_set_upfonts:nnnnnn
```

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

```
3895 \bool_if:NT \c__ctex_everyysel_loaded_bool
3896 { \RequirePackage { pxeveryysel } }

3897 <uptex|aptex>
```

15.13.6 修改主要字体命令

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

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

```
3898 <*pdf|luatex|uptex|aptex>
<pdf|tex> 3899 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
```

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

```
\CTEX@rmfamilyhook
3900 \cs_new_protected:Npn \ctex_provide_font_hook:NNN #1#2
3901 {
3902   \exp_args:Nc \__ctex_provide_font_hook_aux:NNNN
3903   { CTEX \cs_to_str:N #2 } #1#2
3904 }
3905 \cs_new_protected:Npn \__ctex_provide_font_hook_aux:NNNN #1#2#3#4
3906 {
3907   \tl_new:N #1
3908   \exp_args:Ne \ctex_gadd_ltxhook:nn { \cs_to_str:N #2 } {#1}
3909 }
3910 \ctex_provide_font_hook:NNN \rmfamily \@rmfamilyhook \selectfont
3911 \ctex_provide_font_hook:NNN \sffamily \@sffamilyhook \selectfont
3912 \ctex_provide_font_hook:NNN \ttfamily \@ttfamilyhook \selectfont
<pdf|tex|luatex> 3913 \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。

```
3914 \tl_if_exist:NF \CJKfamilydefault
3915 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
3916 <*pdf|luatex>
3917 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
3918 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
3919 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
3920 \tl_gput_right:Nn \CTEX@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
3921 \tl_gput_right:Nn \CTEX@sffamilyhook { \CJKfamily { \CJKsfdefault } }
```

```

3922 \tl_gput_right:Nn \CTEX@ttfamilyhook { \CJKfamily { \CJKttdefault } }
3923 \tl_gput_right:Nn \CTEX@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }
3924 <pdf|u|u|u>

```

up $\text{\LaTeX}$ 不需要补丁 `\normalfont`, 只需要修改 `\kanjifamilydefault`。

```

3925 <*uptex|aptex>
3926 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { zhrm } }
3927 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { zhsf } }
3928 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { zhst } }
3929 \tl_gput_right:Nn \CTEX@rmfamilyhook { \kanjifamily { \CJKrmdefault } }
3930 \tl_gput_right:Nn \CTEX@sffamilyhook { \kanjifamily { \CJKsfdefault } }
3931 \tl_gput_right:Nn \CTEX@ttfamilyhook { \kanjifamily { \CJKttdefault } }
3932 \tl_gset:Nn \kanjifamilydefault { \CJKfamilydefault }
3933 </uptex|aptex>

```

zhmCJK 判断结束。

```
<pdf|u|u|u> 3934 \fi:
```

使修改立刻生效, 保证导言区字体族正确。

```
3935 \normalfont
```

在导言区末尾更新 `\CJKfamilydefault`, pdf $\text{\TeX}$ 已经在之前使用过此处代码。

```
<pdf|u|u|u> 3936 \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。

```

3937 \cs_new_protected:Npn \ctex_update_default_family:
3938 {
3939   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
3940   {
3941     \group_begin:
3942     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
3943     \tl_gset:Ne \CJKfamilydefault
3944     {
3945       \str_case:onF { \familydefault }
3946       {
3947         { \rmdefault } { \exp_not:N \CJKrmdefault }
3948         { \sfdefault } { \exp_not:N \CJKsfdefault }
3949         { \ttdefault } { \exp_not:N \CJKttdefault }
3950       }
3951       { \CJKfamilydefault }
3952     }
3953     \group_end:
3954   }

```

使用 Lua $\text{\TeX}$ 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 $\text{\LaTeX}$ 时, C19rm、JY2rm 等总是有定义的, 不需要确认。

```

<luatex> 3955   \ctex_ltj_ensure_default_family:
3956   }

```

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

```

3957 \tl_new:N \l__ctex_family_default_init_tl
3958 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
3959 \tl_set:Ne \l__ctex_family_default_init_tl
3960 {
3961   \exp_not:N \__ctex_family_default_wrap:n
3962   { \exp_not:o { \CJKfamilydefault } }
3963 }
3964 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl

```

```
3965 </pdfTeX|luatex|uptex|aptex>
```

15.13.7 hyperref 兼容性处理

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。hyperref 2021-02-04 版开始默认设置 unicode 为 true, 对于DVIPDFMx 驱动, 我们需要禁用这个设置, 为此设置 pdfencoding 为 pdfdoc。

driverfallback 是 hyperref 的加载选项, 加载后即被禁用, 不能通过 `\hypersetup` 设置。因此当 hyperref 已被其他宏包(如 beamer)加载时, 只能跳过此选项。

```
3966 <*pdfTeX>
3967 \ifpackageloaded { hyperref }
3968 {
3969   \PassOptionsToPackage { driverfallback = dvipdfmx } { hyperref } }
3970 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3971 {
3972   \ctex_hypersetup:n { CJKbookmarks = true }
3973   \sys_if_output_pdf:TF
3974     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
3975     {
3976       \ctex_hypersetup:n { pdfencoding = pdfdoc }
3977       \ctex_at_end_package:nn { hyperref }
3978       {
3979         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
3980         {
3981           \ctex_at_shipout_first:n
3982             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
3983         }
3984         { \RequirePackage { xCJK2uni } }
3985       }
3986     }
3987 }
3988 { \ctex_hypersetup:n { pdfencoding = unicode } }
3989 </pdfTeX>
```

X_gTeX 和LuaTeX 统一设置 pdfencoding 为 unicode。

```
3990 <*xetex|luatex>
3991 \ctex_hypersetup:n { pdfencoding = unicode }
3992 </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。

```
3993 <*uptex|aptex>
3994 \ifpackageloaded { hyperref }
3995 { \ctex_hypersetup:n { pdfencoding = pdfdoc } }
3996 {
3997   \ctex_hypersetup:n
3998     {
3999       driverfallback = dvipdfmx ,
4000       pdfencoding     = pdfdoc
4001     }
4002 }
4003 \ctex_at_end_package:nn { hyperref }
4004 {
4005   \ctex_at_shipout_first:n
4006     { \special { pdf:tounicode~UTF8-UTF16 } }
4007 }
4008 </uptex|aptex>
```

4009 `<*pdf|*xetex|*luatex|*uptex|*aptex>`

15.13.8 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。

```

4010 <*pdf|*xetex>
4011 \ctex_at_end_package:nn { CJKfntef }
4012 {
4013   \normalem
4014   \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
4015     { \tl_clear:c { CJK#1color } }
4016 </pdf|*xetex>
4017 <*xetex>
4018 \ctex_at_end_package:nn { xeCJKfntef }
4019 {
4020   \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
4021   {
4022     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
4023       { \xeCJKsetup { #1 / format = { } } }
4024   }
4025   {
4026     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
4027       { \tl_clear:c { CJK#1color } }
4028   }
4029 </xetex>
4030 <*pdf|*xetex>
4031   \clist_map_inline:nn
4032     { underdot , underline , underdblline , underwave , sout , xout }
4033     { \__ctex_clear_fntef_color:n {#1} }
4034 }
4035 </pdf|*xetex>

```

15.13.9 \ccwd 的更新

```

\ctex_update_ccwd: 4036 \cs_new_protected:Npn \ctex_update_ccwd:
\ccwd 4037 <*pdf|*xetex>
4038 {
4039   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
4040   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
4041 }
4042 </pdf|*xetex>
4043 <*luatex>
4044 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
4045 </luatex>
4046 <*uptex|*aptex>
4047 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
4048 </uptex|*aptex>
4049 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

4050 \cs_new_protected:Npn \ctex_update_ccglue:
4051 <*pdf|*xetex>
4052 {
4053   \cs_set_protected:Npn \CJKglue
4054     { \skip_horizontal:N \l__ctex_ccglue_skip }
4055   \cs_if_exist:NT \__ctex_ccglue:
4056     { \cs_set_eq:NN \__ctex_ccglue: \CJKglue }
4057 }
4058 </pdf|*xetex>
4059 <*luatex>
4060 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }

```



```

4061 </luatex>
4062 <*uptex|aptex>
4063   { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
4064 </uptex|aptex>
4065 \skip_new:N \l__ctex_ccglue_skip

```

检查用户是否修改过汉字间距。

```

\ctex_if_ccglue_touched_p:
\ctex_if_ccglue_touched:TF
4066 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
4067   {
4068 <*pdf|etex|xetex>
4069     \if_meaning:w \CJKglue \l__ctex_ccglue:
4070     \prg_return_false: \else: \prg_return_true: \fi:
4071 </pdf|etex|xetex>
4072 <*luatex>
4073     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
4074     { \prg_return_false: } { \prg_return_true: }
4075 </luatex>
4076 <*uptex|aptex>
4077     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
4078     { \prg_return_false: } { \prg_return_true: }
4079 </uptex|aptex>
4080   }
<pdf|etex|xetex> 4081 \ctex_at_end:n { \cs_new_eq:NN \l__ctex_ccglue: \CJKglue }
4082 <*pdf|etex|xetex>
4083 \ctex_at_end:n
4084   {
4085     \cs_set_protected:Npn \l__ctex_update_stretch_auxii:
4086     {
4087       \ctex_if_ccglue_touched:TF
4088       { \ctex_update_ccwd: }
4089       { \l__ctex_update_stretch_auxiii: }
4090     }
4091   }
4092 </pdf|etex|xetex>

```

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

```

4093 \cs_new_protected:Npn \ctex_update_em_unit:
<pdf|etex|xetex> 4094   { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 4095   { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 4096   { \dim_set:Nn \ccwd { 1zw } }

```

15.13.10 其他

```

\ctex_add_to_selectfont:n 4097 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
\CTEX@selectfont@hook 4098   {
4099     \cs_set_protected:Npx \CTEX@selectfont@hook
4100     { \exp_not:o { \CTEX@selectfont@hook #1 } }
4101   }
4102 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:

```

使用 every_{sel} 包的情况。EverySelectfont 直到文档开始时才有效。为了 \ccwd 和 LuaTeX-ja 的字体设置在导言区也可用, 我们还需要在这里手工修改 \selectfont。every_{sel} 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。pxevery_{sel} 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。

```

4103 \bool_if:NTF \c__ctex_everysel_loaded_bool
4104 {
4105   \cs_if_free:NF \@EverySelectfont@Init
4106   {
4107     \group_begin:
4108     \cs_set:Npn \__ctex_tmp:N #1
4109     {
4110       \tl_set:Nn \l__ctex_tmp_tl {#1}
4111       \cs_new_eq:NN \CTEX@selectfont@save #1
4112       \cs_new_protected:Npn \__ctex_restore_selectfont:
4113       {
4114         \cs_if_free:NF \scr@new@selectfont
4115         {

```

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

```

4116 <*pdfTeX|uTeX|apTeX>
4117   \cs_if_free:NF \par@update
4118 <*uTeX|apTeX>
4119   {
4120     \tl_put_right:Nn \@EverySelectfont@Init
4121     { \tl_put_right:Nn #1 { \par@update } }
4122   }
4123 </uTeX|apTeX>
4124 <*pdfTeX>
4125   { \tl_put_right:Nn #1 { \par@update } }
4126   \cs_set_eq:NN \scr@selectfont \CTEX@selectfont@save
4127 </pdfTeX>
4128 </pdfTeX|uTeX|apTeX>
4129   \cs_set_eq:NN \scr@new@selectfont #1
<!pdfTeX> 4130   \cs_set_eq:NN \CTEX@selectfont@save \scr@selectfont
4131   }
4132   \tl_put_left:Nn \@EverySelectfont@Init
4133   { \cs_set_eq:NN #1 \CTEX@selectfont@save }
4134   \cs_undefine:N \__ctex_restore_selectfont:
4135   }
4136   }
4137   \ctex_parse_name:NN \__ctex_tmp:N \selectfont
4138   \exp_last_unbraced:NNo \group_end:
4139   \ctex_patch_cmd_once:NnnTF { \l__ctex_tmp_tl }
4140   { \ExplSyntaxOff }
4141   { \size@update }
4142   { \CTEX@selectfont@hook \size@update }
4143   { \__ctex_restore_selectfont: }
4144   { \ctex_patch_failure:N \selectfont }
4145   }
4146   \cs_new_protected:Npn \ctex_gadd_selectfont_hook:n
4147   { \EverySelectfont }
4148 }

```

使用 L^AT_EX 2021-06-01 的新钩子,不使用 `everysel` 包的情况。

```

4149 {
4150   \cs_new_protected:Npn \ctex_gadd_selectfont_hook:n
4151   { \ctex_gadd_ltxhook:nn { selectfont } }
4152 }

```

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

```

4153 <*pdfTeX>
4154 \ctex_gadd_selectfont_hook:n
4155 { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
4156 </pdfTeX>
4157 <*xetex|uatex|uTeX|apTeX>
4158 \ctex_gadd_selectfont_hook:n { \CTEX@selectfont@hook }
4159 </xetex|uatex|uTeX|apTeX>

```

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

```

4160 <lua>
4161 \ctex_add_to_selectfont:n
4162 {
4163   \ctex_ltj_select_font:
4164   \ctex_ltj_select_alterate_font:
4165 }
4166 \tl_set:Nn \CJK@family { song } \selectfont
4167 \tl_clear:N \CJK@family
4168 </lua>

```

`\ctex_update_xkanjiskip:` `upTeX` 和 `LuaTeX-ja` 对 `\xkanjiskip` 都是即时赋值。单位 `zw` 与字体相关, 因此需要每次 `\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。

```

4169 <lua>
4170 \cs_new_protected:Npn \ctex_update_xkanjiskip:
4171 {
4172   \skip_if_eq:nnT
4173   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
4174   { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
4175   {
4176     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
4177     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
4178     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
4179   }
4180 }
4181 \tl_new:N \l__ctex_xkanjiskip_tl
4182 \tl_set:Nn \l__ctex_xkanjiskip_tl
4183 { .25\zw plus 1pt minus 1pt }
4184 { .25zw plus 1pt minus 1pt }
4185 \skip_new:N \l__ctex_xkanjiskip_skip
4186 \skip_set:Nn \l__ctex_xkanjiskip_skip
4187 { \ltjgetparameter { xkanjiskip } }
4188 { \tex_xkanjiskip:D }

4189 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
4190 </lua>

```

`\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
\cwg
4191 <lua>
4192 \dim_new:N \cht
4193 \dim_new:N \cdp
4194 \dim_new:N \cwg
4195 \group_begin:
4196 \char_set_catcode_space:n { 32 }
4197 \lua_now:e
4198 {
4199   local nulltable = { }
4200   local fmt = luatexja.jfont.font_metric_table
4201   local getattribute = tex.getattribute
4202   local setdimen = tex.setdimen
4203   ctex.newluacmd("ctex_update_kanjisize:", function ()
4204     local ft = fmt[getattribute("ltj@curjfont")] or nulltable
4205     local ft = ft and ft.char_type or nulltable
4206     local fk = ft and ft[0] or nulltable
4207     setdimen("cht", fk.height or 0)
4208     setdimen("cdp", fk.depth or 0)
4209     setdimen("cwg", fk.width or ft.zw or 0)
4210     end, "global", "protected")
4211 }

```

```

4212 \group_end:
4213 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
4214 \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。

```

4215 \ctex_define:n
4216 {
4217   <pdf|xetex>
4218     space .choice: ,
4219     space / true .code:n =
4220     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
4221     <xetex> { \xeCJKsetup { CJKspace = true } } ,
4222     space / auto .code:n =
4223     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
4224     <xetex> { \xeCJKsetup { CJKspace = false } } ,
4225     space / false .code:n =
4226     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
4227     <xetex> { \xeCJKsetup { CJKspace = false } } ,
4228     space .default:n = { true } ,
4229     space .initial:n = { auto }
4230   </pdf|xetex>
4231   <lua|uptex|aptex>
4232     space .code:n =
4233     { \msg_warning:nn { ctex } { invalid-option } }
4234   </lua|uptex|aptex>
4235 }

```

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-jan 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。

```

4236 \ctex_define:n
4237 {
4238   punct .code:n =
4239   {
4240     \tl_set:Nx \l__ctex_punct_tl {#1}
4241     <pdf|xetex> \punctstyle { \l__ctex_punct_tl }
4242     <xetex> \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
4243     <lua> \ctex_set_jfm:o { \l__ctex_punct_tl }
4244     <uptex|aptex> \msg_warning:nn { ctex } { invalid-option }
4245   } ,
4246   punct .default:n = { quanjiao } ,
4247 }

```

experiment 实验性选项。experiment/CJKecglue 用于统一设置 CJK 文字与西文之间的间距。XeTeX 下转发给 xeCJK, LuaTeX 和 upTeX 下设置 xkanjiskip, pdfTeX 下不支持。

```

4248 \ctex_define:n
4249 { experiment .meta:nn = { ctex / experiment } {#1} }
4250 \keys_define:nn { ctex / experiment }
4251 {
4252   <pdf|xetex>
4253     CJKecglue .code:n =
4254     { \xeCJKsetup { CJKecglue = { \skip_horizontal:n {#1} } } } ,
4255   </pdf|xetex>
4256   <lua|uptex|aptex>
4257     CJKecglue .code:n =
4258     {
4259       \tl_set:Nx \l__ctex_xkanjiskip_tl {#1}
4260       \skip_set:Nx \l__ctex_xkanjiskip_skip {#1}
4261       <lua> \ctex_lj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
4262       <uptex|aptex> \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
4263     } ,

```

```

4264 </luatex|uptex|aptex>
4265 <*pdfTeX>
4266     CJKecglue .code:n =
4267     { \msg_warning:nn { ctex } { CJKecglue-unsupported } } ,
4268 </pdfTeX>
4269     CJKecglue .value_required:n = true
4270 }

4271 </pdfTeX|xetex|luatex|uptex|aptex>

4272 <@@=>

```

ctex-auxpkg.dtx

```

4273 <@@=ctex>

```

15.14 translator 宏包的中文字典

```

4274 <*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。

```

4275 <*theorem>
4276 \providetranslation{Comments}{评论}
4277 \providetranslation{comments}{评论}
4278 \providetranslation{Comment}{评论}
4279 \providetranslation{comment}{评论}
4280 \providetranslation{Corollaries}{推论}
4281 \providetranslation{corollaries}{推论}
4282 \providetranslation{Corollary}{推论}
4283 \providetranslation{corollary}{推论}
4284 \providetranslation{Definitions}{定义}
4285 \providetranslation{definitions}{定义}
4286 \providetranslation{Definition}{定义}
4287 \providetranslation{definition}{定义}
4288 \providetranslation{Examples}{例}
4289 \providetranslation{examples}{例}
4290 \providetranslation{Example}{例}
4291 \providetranslation{example}{例}
4292 \providetranslation{Exercises}{练习}
4293 \providetranslation{exercises}{练习}
4294 \providetranslation{Exercise}{练习}
4295 \providetranslation{exercise}{练习}
4296 \providetranslation{Facts}{事实}
4297 \providetranslation{facts}{事实}
4298 \providetranslation{Fact}{事实}
4299 \providetranslation{fact}{事实}
4300 \providetranslation{Key Lemmas}{关键引理}
4301 \providetranslation{key lemmas}{关键引理}
4302 \providetranslation{Key Lemma}{关键引理}
4303 \providetranslation{key lemma}{关键引理}
4304 \providetranslation{Key Observations}{关键观察}
4305 \providetranslation{key observations}{关键观察}
4306 \providetranslation{Key Observation}{关键观察}
4307 \providetranslation{key observation}{关键观察}
4308 \providetranslation{Lemmas}{引理}
4309 \providetranslation{lemmas}{引理}
4310 \providetranslation{Lemma}{引理}
4311 \providetranslation{lemma}{引理}
4312 \providetranslation{Main Theorems}{主要定理}
4313 \providetranslation{main theorems}{主要定理}
4314 \providetranslation{Main Theorem}{主要定理}
4315 \providetranslation{main theorem}{主要定理}
4316 \providetranslation{Observations}{观察}
4317 \providetranslation{observations}{观察}

```

```

4318 \providetranslation{Observation}{观察}
4319 \providetranslation{observation}{观察}
4320 \providetranslation{Problems}{问题}
4321 \providetranslation{problems}{问题}
4322 \providetranslation{Problem}{问题}
4323 \providetranslation{problem}{问题}
4324 \providetranslation{Proofs}{证明}
4325 \providetranslation{proofs}{证明}
4326 \providetranslation{Proof}{证明}
4327 \providetranslation{proof}{证明}
4328 \providetranslation{Proof Sketch}{证明提要}
4329 \providetranslation{Proof sketch}{证明提要}
4330 \providetranslation{proof sketch}{证明提要}
4331 \providetranslation{Proof Sketches}{证明提要}
4332 \providetranslation{Proof sketches}{证明提要}
4333 \providetranslation{proof sketches}{证明提要}
4334 \providetranslation{Sketch of Proof}{证明提要}
4335 \providetranslation{Sketch of Proofs}{证明提要}
4336 \providetranslation{Sketch of proof}{证明提要}
4337 \providetranslation{Sketch of proofs}{证明提要}
4338 \providetranslation{sketch of proof}{证明提要}
4339 \providetranslation{sketch of proofs}{证明提要}
4340 \providetranslation{Propositions}{命题}
4341 \providetranslation{propositions}{命题}
4342 \providetranslation{Proposition}{命题}
4343 \providetranslation{proposition}{命题}
4344 \providetranslation{Remarks}{注}
4345 \providetranslation{remarks}{注}
4346 \providetranslation{Remark}{注}
4347 \providetranslation{remark}{注}
4348 \providetranslation{Solutions}{解}
4349 \providetranslation{solutions}{解}
4350 \providetranslation{Solution}{解}
4351 \providetranslation{solution}{解}
4352 \providetranslation{Theorems}{定理}
4353 \providetranslation{theorems}{定理}
4354 \providetranslation{Theorem}{定理}
4355 \providetranslation{theorem}{定理}
4356 </theorem>
4357 </dict>

```

15.15 ctexcap 宏包

4358 <*ctexcap>

ctexcap 是过时宏包。

```

4359 \clist_new:N \l__ctex_ctexcap_options_clist
4360 \clist_set:Ne \l__ctex_ctexcap_options_clist
4361 { \exp_not:v { opt@ \@currname . \@currentext } , heading }
4362 \msg_new:nnn { ctexcap } { deprecated }
4363 {
4364   Package~`ctexcap'~is~deprecated.\\
4365   Please~use~package~`ctex'~with~option~`#1'~instead: \\\
4366   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4367 }
4368 \msg_warning:nne { ctexcap } { deprecated }
4369 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

4370 \PassOptionsToPackage { heading = true } { ctexcap }
4371 \RequirePackageWithOptions { ctex }
4372 </ctexcap>

```

15.16 ctexhook 宏包

4373 <*ctexhook>

\ctex_if_format_at_least:nTF 与 \IfFormatAtLeastTF 同义。

```
4374 \cs_new:Npn \ctex_if_format_at_least:nTF
4375   { \@ifl@t@r \fmtversion }
```

\ctex_file_input:n 输入文件, 关闭 L^AT_EX3 语法环境, 并设置 @ 为字母类, 利用 l3cctab 实现。我们使用 \file_-
 \ctex_push_file: input:n 而不是 L^AT_EX2_ε 的 \input 或者 \InputIfFileExists 载入文件, 因此 L^AT_EX2_ε 的文
 \ctex_pop_file: 件钩子都无效。

```
4376 \cs_new_protected:Npn \ctex_file_input:n #1
4377   {
4378     \ctex_push_file:
4379     \file_input:n {#1}
4380     \ctex_pop_file:
4381   }
4382 \bool_if_exist:NTF \l__kernel_expl_bool
4383   {
4384     \cs_new_protected:Npn \ctex_push_file:
4385     {
4386       \seq_gpush:Nx \g__ctex_expl_status_seq
4387       { \bool_if:NTF \l__kernel_expl_bool { 1 } { 0 } }
4388       \bool_set_false:N \l__kernel_expl_bool
4389       \cctab_begin:N \c__ctex_package_cctab
4390     }
4391     \cs_new_protected:Npn \ctex_pop_file:
4392     {
4393       \cctab_end:
4394       \seq_gpop:NN \g__ctex_expl_status_seq \l__ctex_expl_status_tl
4395       \int_if_odd:nTF { \l__ctex_expl_status_tl }
4396       { \bool_set_true:N \l__kernel_expl_bool }
4397       { \bool_set_false:N \l__kernel_expl_bool }
4398     }
4399     \tl_new:N \l__ctex_expl_status_tl
4400     \seq_new:N \g__ctex_expl_status_seq
4401   }
4402   {
4403     \cs_new_protected:Npn \ctex_push_file:
4404     { \cctab_begin:N \c__ctex_package_cctab }
4405     \cs_new_protected:Npn \ctex_pop_file:
4406     { \cctab_end: }
4407   }
4408 \cctab_const:Nn \c__ctex_package_cctab
4409   {
4410     \cctab_select:N \c_document_cctab
4411     \char_set_catcode_letter:n { 64 }
4412   }
```

\ctex_disable_package:n 禁止宏包载入。采用 L^AT_EX 2020-10-01 提供的 \disable@package@load 实现, 否则采用传统
 方式: 预定义 \ver@<package>.sty 标识符。

```
4413 \cs_new_protected:Npn \ctex_disable_package:n #1
4414   {
4415     \@ifpackageloaded {#1}
4416     { \msg_error:nnee }
4417     { \__ctex_disable_package_aux:nnnn }
4418     { ctexhook } { disable-package } {#1} { \@currname }
4419   }
4420 \cs_new_protected:Npn \__ctex_disable_package_aux:nnnn #1#2#3#4
4421   {
4422     \cs_if_exist:NTF \disable@package@load
```

```

4423 {
4424   \exp_args:Nne \disable@package@load {#3}
4425   { \msg_warning:nnnn {#1} {#2} {#3} {#4} }
4426 }
4427 { \tl_const:cn { ver@ #3 . \@pkgextension } { 9999/99/99 } }
4428 }
4429 \msg_new:nnn { ctexhook } { disable-package }
4430 { Package~`#1'~can~not~be~loaded~with~`#2'. }

```

`\ctex_replace_package:nn` 替换宏包。采用 L^AT_EX 2020-10-01 提供的 `\declare@file@substitution` 实现, 否则给出无效警告。

```

4431 \ctex_if_format_at_least:nTF { 2020/10/01 }
4432 {
4433   \cs_new_protected:Npn \ctex_replace_package:nn #1#2
4434   {
4435     \declare@file@substitution
4436     { #1 . \@pkgextension }
4437     { #2 . \@pkgextension }
4438   }
4439 }
4440 {
4441   \cs_new_protected:Npn \ctex_replace_package:nn
4442   { \msg_warning:nnnn { ctexhook } { replace-package-invalid } }
4443   \msg_new:nnn { ctexhook } { replace-package-invalid }
4444   {
4445     \token_to_str:N \ctex_replace_package:nn \{#1\}\{#2\}~is~invalid~
4446     before~LaTeX~2020-10-01.
4447   }
4448 }

```

`\ctex_at_begin_package:nn` 如果宏包已经被载入, 则钩子无效, 给出警告。

```

4449 \cs_new_protected:Npn \ctex_at_begin_package:nn #1
4450 {
4451   \@ifpackageloaded {#1}
4452   { \__ctex_package_loaded_warning:nn {#1} }
4453   { \ctex_gadd_package_hook:nnn { before } {#1} }
4454 }
4455 \cs_new_protected:Npn \__ctex_package_loaded_warning:nn #1#2
4456 { \msg_warning:nne { ctexhook } { invalid-hook } {#1} }
4457 \msg_new:nnn { ctexhook } { invalid-hook }
4458 {
4459   Package~`#1'~is~loaded. \\
4460   \token_to_str:N \ctex_at_begin_package:nn \{#1\}\{...\}~is~invalid.
4461 }

```

`\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。

```

4462 \cs_new_protected:Npn \ctex_at_end_package:nn #1
4463 {
4464   \@ifpackageloaded {#1}
4465   { \use:n }
4466   { \ctex_gadd_package_hook:nnn { after } {#1} }
4467 }

```

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。

```

4468 \ctex_if_format_at_least:nTF { 2020/10/01 }
4469 {
4470   \cs_new_protected:Npx \ctex_gadd_ltxhook:nn #1
4471   { \hook_gput_code:nnn {#1} { \c_novalue_tl } }
4472   \cs_new_protected:Npn \ctex_at_end_preamble:n
4473   { \ctex_gadd_ltxhook:nn { begindocument/before } }

```



```

4474 \cs_new_protected:Npn \ctex_after_end_preamble:n
4475 { \ctex_gadd_ltxhook:nn { begindocument/end } }
4476 \cs_new_protected:Npx \ctex_gadd_package_hook:nnn #1#2
4477 {
4478   \ctex_if_format_at_least:nTF { 2021/11/15 }
4479   { \ctex_gadd_ltxhook:nn { package/#2/#1 } }
4480   { \ctex_gadd_ltxhook:nn { package/#1/#2 } }
4481 }
4482 \file_input_stop:
4483 }
4484 { }

```

对于 L^AT_EX 2020/10/01 之前的版本, 需要自行补丁。

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4485 \cs_new_protected:Npn \ctex_at_end_preamble:n
4486 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl }
4487 \cs_new_protected:Npn \ctex_after_end_preamble:n
4488 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl }
4489 \cs_new_protected:Npn \CTEX@document@left@hook
4490 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4491 \cs_new_protected:Npn \CTEX@document@right@hook
4492 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4493 \cs_set_nopar:Npx \document
4494 {
4495   \CTEX@document@left@hook
4496   \exp_not:o { \document }
4497   \CTEX@document@right@hook
4498 }
4499 \tl_new:N \g__ctex_end_preamble_hook_tl
4500 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

给钩子附加内容。

```

\ctex_gadd_package_hook:nnn
\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn
4501 \cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
4502 { \ctex_gadd_hook:cn { g__ctex_at_ #1 _ #2 _hook_tl } }
4503 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1
4504 {
4505   \tl_if_exist:NF #1 { \tl_new:N #1 }
4506   \tl_gput_right:Nn #1
4507 }
4508 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

宏包钩子, 只执行一次, 用后清除。

```

\ctex_use_package_hook:nn
4509 \cs_new_protected:Npn \ctex_use_package_hook:nn #1#2
4510 {
4511   \group_begin: \exp_args:Nnc \group_end:
4512   \__ctex_use_package_hook_aux:N { g__ctex_at_ #1 _ #2 _hook_tl }
4513 }
4514 \cs_new_protected:Npn \__ctex_use_package_hook_aux:N #1
4515 { \cs_if_exist_use:NT #1 { \cs_undefine:N #1 } }

```

\@resetOptions \@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。

```

\@resetOptions
\CTEX@resetOptions@hook
4516 \tl_put_right:Nn \@resetOptions { \CTEX@resetOptions@hook }
4517 \cs_new_protected:Npn \CTEX@resetOptions@hook
4518 {
4519   \cs_if_eq:NNT \@currentx \@pkgextension
4520   { \ctex_use_package_hook:nn { before } { \@currname } }
4521 }

```

```

\@popfilename 对 \@popfilename 做补丁来实现 \ctex_at_end_package:nn 的功能。
\CTEX@popfilename@hook
4522 \tl_put_left:Nn \@popfilename { \CTEX@popfilename@hook }
4523 \cs_new_protected:Npn \CTEX@popfilename@hook
4524 {
4525   \cs_if_eq:NNT \@currentx \@pkgextension
4526   { \ctex_use_package_hook:nn { after } { \@currname } }
4527 }
4528 </ctexhook>

```

15.17 ctexpatch 宏包

```
4529 <*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。

```

4530 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
4531 {
4532   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4533   { once } {#2} { \use_i:nn } { \use_ii:nn }
4534 }

```

`\ctex_patch_cmd_all:NnnnTF` 替换所有匹配到的文本。

```

4535 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
4536 {
4537   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4538   { all } {#2} { \use_i:nn } { \use_ii:nn }
4539 }

```

`\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。

```

4540 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
4541 {
4542   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4543   { once }
4544   {
4545     \ExplSyntaxOff
4546     \char_set_catcode_letter:n { 64 }
4547   }
4548   { }
4549   { \ctex_patch_failure:N #1 }
4550 }
4551 \cs_new_protected:Npn \ctex_patch_failure:N #1
4552 { \msg_warning:nne { ctexpatch } { patch-failure } { \token_to_str:N #1 } }
4553 \msg_new:nnn { ctexpatch } { patch-failure }
4554 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

`\ctex_preto_cmd:NnnTF` 在宏的原本定义前面增加钩子。

```

4555 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
4556 {
4557   \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
4558   { left } {#2} { \use_i:nn } { \use_ii:nn }
4559 }

```

`\ctex_appto_cmd:NnnTF` 在宏的原本定义后面追加钩子。

```

4560 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
4561 {
4562   \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
4563   { right } {#2} { \use_i:nn } { \use_ii:nn }
4564 }

```

`\ctex_patch_boot:NnnTF` 参数记号 # 作为宏的参数被读入时,总是会双写,会影响随后的字符串替换。需要先将它转换为普通符号。

```

4565 \cs_new_protected:Npn \ctex_patch_boot:NnnTF #1#2#3#4#5#6
4566 {
4567   \cs_set_protected:Npx \__ctex_patch_true:w { \exp_not:n {#5} }
4568   \cs_set_protected:Npx \__ctex_patch_false:w { \exp_not:n {#6} }
4569   \group_begin:
4570     \char_set_catcode_other:n { 35 }
4571     \ctex_parse_name:NN #1 #2 {#3} {#4}
4572   }
4573 \cs_new_eq:NN \__ctex_patch_true:w \use_i:nn
4574 \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。

```

4575 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
4576 { \ctex_parse_name:NNe #1#2 { \cs_to_str:N #2 } }
4577 \group_begin:
4578 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
4579 {
4580   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
4581   {
4582     \bool_lazy_or:nnTF
4583     { \cs_if_exist_p:c { ##3 ~ } }
4584     { \cs_if_exist_p:c { #1##3 } }
4585     {
4586       \group_begin:
4587       \use:e
4588       {
4589         \group_end:
4590         \__ctex_parse_name:nNNnN
4591         { \cs_replacement_spec:N ##2 }
4592         \exp_not:n ##2
4593         \exp_not:c { ##3 ~ }
4594         \exp_not:c { #1##3 }
4595       } { ##3 } ##1
4596     }
4597     { ##1##2 }
4598   }
4599 \cs_new_protected:Npn \__ctex_parse_name:nNNnN ##1##2##3##4##5##6
4600 {
4601   \exp_args:Nc ##6
4602   {
4603     \str_case:nnTF {##1}
4604     {
4605       { \protect ##3 } { }
4606       { \x@protect ##2 \protect ##3 } { }
4607     }
4608     {
4609       \str_if_eq:eeTF

```

```

4610         { \exp_not:n { #1@protected@ ##3 #1##3 } }
4611         {
4612             \exp_last_unbraced:Ne \__ctex_parse_name:w
4613             { \cs_replacement_spec:N ##3 } #3 ~ #2 \q_stop
4614         }
4615         { #1##5 ~ } { ##5 ~ }
4616     }
4617     {
4618         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
4619         {
4620             { #1@protected@ ##2 ##4 } { }
4621             { #1@ ##4 } { }
4622         }
4623         { #1##5 } { ##5 }
4624     }
4625 }
4626 }
4627 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
4628 }
4629 \use:e
4630 {
4631     \__ctex_tmp:w
4632     { \c_backslash_str }
4633     { \c_left_brace_str }
4634     { \tl_to_str:n { testopt } }
4635 }
4636 \group_end:
4637 \cs_generate_variant:Nn \ctex_parse_name:Nn { NNe }

```

\l__ctex_prefix_str 分别保存宏的 \meaning 中的前缀、参数文本和替换文本。
 \l__ctex_parameter_str
 \l__ctex_replacement_str

```

4638 \str_new:N \l__ctex_prefix_str
4639 \str_new:N \l__ctex_parameter_str
4640 \str_new:N \l__ctex_replacement_str

```

\ctex_get_macro_meaning:NnTF 解构待补丁宏的 \meaning。若命令不是宏,则走向 false 分支。
 __ctex_get_macro_meaning:w

```

4641 \group_begin:
4642 \cs_set_protected:Npn \__ctex_tmp:w #1
4643 {
4644     \prg_new_protected_conditional:Npnn
4645     \ctex_get_macro_meaning:N ##1 { TF }
4646     {
4647         \exp_after:wN \__ctex_get_macro_meaning:w
4648         \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
4649     }
4650     \cs_new_protected:Npn \__ctex_get_macro_meaning:w
4651     ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
4652     {
4653         \tl_if_empty:nTF { ##4 }
4654         { \prg_return_false: }
4655         {
4656             \str_set:Nn \l__ctex_prefix_str { ##1 }
4657             \str_set:Nn \l__ctex_parameter_str { ##2 }
4658             \str_set:Nn \l__ctex_replacement_str { ##3 }
4659             \prg_return_true:
4660         }
4661     }
4662 }
4663 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
4664 \group_end:

```

\ctex_if_rescanable:NnTF 检查宏是否可以重建。

```

4665 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
4666 {

```

```

4667 \ctex_get_macro_meaning:NTF #1
4668 {
4669   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
4670   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
4671 }
4672 {#4}
4673 }
4674 \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。

```

4675 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
4676 {
4677   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
4678   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
4679   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
4680   \use:e
4681   {
4682     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
4683     \exp_not:o { \l__ctex_parameter_tl }
4684     { \exp_not:o { \l__ctex_replacement_tl } }
4685   }
4686 }
4687 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
4688 {
4689   \str_if_empty:NTF #2
4690   { \tl_clear:N #1 }
4691   { \tl_set_rescan:Nno #1 {#3} {#2} }
4692 }
4693 \tl_new:N \l__ctex_prefix_tl
4694 \tl_new:N \l__ctex_parameter_tl
4695 \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。

```

4696 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
4697 {
4698   \group_end:
4699   \ctex_if_rescanable:NnTF #1 {#3}
4700   {
4701     \use:e
4702     {
4703       \__ctex_patch_replace:nnnTF {#2}
4704       { \tl_to_str:n {#4} }
4705       { \tl_to_str:n {#5} }
4706     }
4707     {
4708       \__ctex_patch_rebuild:Nn #1 {#3}
4709       \__ctex_patch_true:w
4710     }
4711     { \__ctex_patch_false:w }
4712   }
4713   { \__ctex_patch_false:w }
4714 }

```

__ctex_patch_replace:nnnTF 替换前先检查原文本是否存在。

```

4715 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
4716 {
4717   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
4718   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
4719 }

```

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

4720 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4

```

```

4721 {
4722   \group_end:
4723   \ctex_get_macro_meaning:NTF #1
4724   {
4725     \str_if_empty:NTF \l__ctex_parameter_str
4726     { \__ctex_hookto_cmd_parameterless:Nnnnw }
4727     { \__ctex_hookto_cmd_parameter:Nnnnw }
4728     #1 {#2} {#3} {#4}
4729   }
4730   { \__ctex_patch_false:w }
4731 }

```

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。

```

4732 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
4733 {
4734   \str_if_empty:NF \l__ctex_prefix_str
4735   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
4736   \tex_edef:D #1
4737   {
4738     \use:c { __ctex_ #2 _hook_aux:nn }
4739     { \exp_not:o {#1} }
4740     { \exp_not:n {#4} }
4741   }
4742   \__ctex_patch_true:w
4743 }
4744 \cs_generate_variant:Nn \tl_rescan:nn { no }
4745 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
4746 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

如果宏有参数,需要在字符串中进行附加,然后再重建。

```

4747 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
4748 {
4749   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
4750   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
4751   {
4752     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
4753     \__ctex_patch_rebuild:Nn #1 {#3}
4754     \__ctex_patch_true:w
4755   }
4756   { \__ctex_patch_false:w }
4757 }

```

4758 <ctexpatch>

4759 <@@= >

ctex-scheme.dtx

4760 <@@=ctex>

15.17.1 标题配置文件

4761 <*name>

```

4762 \keys_set_known:nn { ctex }
4763 {
4764   contentsname = 目录 ,
4765   listfigurename = 插图 ,
4766   listtablename = 表格 ,
4767   figurename = 图 ,
4768   tablename = 表 ,
4769   abstractname = 摘要 ,
4770   indexname = 索引 ,
4771   bibname = 参考文献 ,

```

```

4772     appendixname = 附录 ,
4773     proofname    = 证明 ,
4774     algorithmname = 算法 ,
4775     refname      = 参考文献 ,
4776     continuation = (续) ,
4777     part / name = { 第 , 部分 } ,
4778     chapter / name = { 第 , 章 }
4779 }
4780 </name>

```

15.17.2 ctexopts.cfg

这里仅为配置文件示例:使用 Windows 下的字体设置。

```

4781 <*ctexopts>
4782 %%
4783 %% \ctex_set:nn { option } { fontset = windows }
4784 </ctexopts>

```

15.17.3 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。

```

4785 <*scheme&(article|book|report|beamer)>

4786 \ctex_set:nn { part }
4787 {
4788     aftertitle = \par ,
4789 <*article|book|report>
4790     hang      = false ,
4791 </article|book|report>
4792 <*plain>
4793     name      = \partname \space ,
4794 <*article|book|report>
4795     number    = \thepart ,
4796 </article|book|report>
4797 <*beamer>
4798     number    = \insertromanpartnumber ,
4799 </beamer>
4800 </plain>
4801 <*chinese>
4802     number    = \chinese { part } ,
4803 </chinese>
4804 <*article>
4805     beforekip = 4ex ,
4806     afterskip = 3ex ,
4807 <*plain>
4808     format    = \raggedright ,
4809     nameformat = \Large \bfseries ,
4810     aftername  = \par \nobreak ,
4811     titleformat = \huge \bfseries ,
4812     afterindent = false
4813 </plain>
4814 <*chinese>
4815     format    = \Large \bfseries \centering ,
4816     aftername  = \quad ,
4817     afterindent = true
4818 </chinese>
4819 </article>
4820 <*book|report>
4821     aftername  = \par \vskip 20 \p@ ,
4822     beforekip  = 0pt \@plus 1fil ,
4823     afterskip  = 0pt \@plus 1fil ,

```

```

4824     pagestyle      = plain ,
4825     break          = \if@openright \cleardoublepage \else \clearpage \fi ,
4826 <*plain>
4827     format         = \centering ,
4828     nameformat     = \huge \bfseries ,
4829     titleformat    = \Huge \bfseries
4830 </plain>
4831 <*chinese>
4832     format         = \huge \bfseries \centering
4833 </chinese>
4834 </book|report>
4835 <*beamer>
4836     format         = \centering ,
4837     nameformat     = \usebeamerfont { part ~ name }
4838                   \usebeamercolor [fg] { part ~ name } ,
4839     aftername      = \vskip 1em \par ,
4840     titleformat    = \usebeamerfont { part ~ title }
4841 </beamer>
4842 }

4843 <*book|report>
4844 \ctex_set:nn { chapter }
4845 {
4846     pagestyle      = plain ,
4847     aftertitle     = \par ,
4848     hang           = false ,
4849     beforeskip     = 50 \p@ ,
4850     afterskip      = 40 \p@ ,
4851     lofskip        = 10 \p@ ,
4852     lotskip        = 10 \p@ ,
4853     break          = \if@openright \cleardoublepage \else \clearpage \fi ,
4854 <*plain>
4855     name           = \chaptername \space ,
4856     number         = \thechapter ,
4857     format         = \raggedright ,
4858     nameformat     = \huge \bfseries ,
4859     aftername      = \par \nobreak \vskip 20 \p@ ,
4860     titleformat    = \Huge \bfseries ,
4861     afterindent    = false ,
4862     tocline        = \CTEXnumberline {#1} #2
4863 </plain>
4864 <*chinese>
4865     number         = \chinese { chapter } ,
4866     format         = \huge \bfseries \centering ,
4867     aftername      = \quad ,
4868     afterindent    = true
4869 </chinese>
4870 }
4871 </book|report>

4872 \ctex_set:nn { section }
4873 {
4874 <*article|book|report>
4875     number         = \thesection ,
4876     aftername      = \quad ,
4877     aftertitle     = \@@par ,
4878     beforeskip     = 3.5ex \@plus 1ex \@minus .2ex ,
4879     afterskip      = 2.3ex \@plus .2ex ,
4880     runin          = false ,
4881     break          = \addpenalty \@secpenalty ,
4882 <*plain>
4883     format         = \Large \bfseries ,
4884     afterindent    = false
4885 </plain>
4886 <*chinese>
4887     format         = \Large \bfseries \centering ,
4888     afterindent    = true
4889 </chinese>

```



```

4890 </article|book|report>
4891 <*beamer>
4892 <*plain>
4893     name          = \sectionname \space ,
4894 </plain>
4895     format         = \centering ,
4896     number         = \insertsectionnumber ,
4897     nameformat     = \usebeamerfont { section ~ name }
4898                   \usebeamercolor [fg] { section ~ name } ,
4899     aftername      = \vskip 1em \par ,
4900     titleformat    = \usebeamerfont { section ~ title } ,
4901     aftertitle     = \par
4902 </beamer>
4903 }

4904 \ctex_set:nn { subsection }
4905 {
4906 <*article|book|report>
4907     number         = \thesubsection ,
4908     format         = \large \bfseries ,
4909     aftername      = \quad ,
4910     aftertitle     = \@@par ,
4911     beforekip      = 3.25ex \@plus 1ex \@minus .2ex ,
4912     afterskip      = 1.5ex \@plus .2ex ,
4913     runin          = false ,
4914     break          = \addpenalty \@secpenalty ,
4915 <*plain>
4916     afterindent    = false
4917 </plain>
4918 <*chinese>
4919     afterindent    = true
4920 </chinese>
4921 </article|book|report>
4922 <*beamer>
4923 <*plain>
4924     name           = \subsectionname \space ,
4925     number         = \insertsubsectionnumber ,
4926 </plain>
4927 <*chinese>
4928     number         = \arabic { section } . \arabic { subsection } ,
4929 </chinese>
4930     format         = \centering ,
4931     nameformat     = \usebeamerfont { subsection ~ name }
4932                   \usebeamercolor [fg] { subsection ~ name } ,
4933     aftername      = \vskip 1em \par ,
4934     titleformat    = \usebeamerfont { subsection ~ title } ,
4935     aftertitle     = \par
4936 </beamer>
4937 }

4938 <*article|book|report>

4939 \ctex_set:nn { subsubsection }
4940 {
4941     number         = \thesubsubsection ,
4942     format         = \normalsize \bfseries ,
4943     aftername      = \quad ,
4944     aftertitle     = \@@par ,
4945     beforekip      = 3.25ex \@plus 1ex \@minus .2ex ,
4946     afterskip      = 1.5ex \@plus .2ex ,
4947     runin          = false ,
4948     break          = \addpenalty \@secpenalty ,
4949 <*plain>
4950     afterindent    = false
4951 </plain>
4952 <*chinese>
4953     afterindent    = true
4954 </chinese>

```

```

4955 }

4956 \ctex_set:nn { paragraph }
4957 {
4958     number      = \theparagraph ,
4959     format      = \normalsize \bfseries ,
4960     aftername    = \quad ,
4961     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
4962     break       = \addpenalty \@secpenalty ,
4963     <*plain>
4964     afterindent = false
4965     </plain>
4966     <*chinese>
4967     afterindent = true
4968     </chinese>
4969 }

4970 \ctex_set:nn { subparagraph }
4971 {
4972     number      = \thesubparagraph ,
4973     format      = \normalsize \bfseries ,
4974     aftername    = \quad ,
4975     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
4976     break       = \addpenalty \@secpenalty ,
4977     <*plain>
4978     afterindent = false
4979     </plain>
4980     <*chinese>
4981     afterindent = true
4982     </chinese>
4983 }

```

处理 sub3section 与 sub4section 的格式。

```

4984 \int_compare:nNnTF \g__ctex_section_depth_int > 2
4985 {
4986     \ctex_set:nn { paragraph }
4987     {
4988         aftertitle = \@@par ,
4989         afterskip   = 1ex \@plus .2ex ,
4990         runin       = false
4991     }
4992 }
4993 {
4994     \ctex_set:nn { paragraph }
4995     {
4996         afterskip   = 1em ,
4997         runin       = true
4998     }
4999 }
5000 \int_compare:nNnTF \g__ctex_section_depth_int > 3
5001 {
5002     \ctex_set:nn { subparagraph }
5003     {
5004         aftertitle = \@@par ,
5005         afterskip   = 1ex \@plus .2ex ,
5006         runin       = false
5007     }
5008 }
5009 {
5010     \ctex_set:nn { subparagraph }
5011     {
5012         afterskip   = 1em ,
5013         runin       = true
5014     }
5015 }
5016 \int_compare:nNnTF \g__ctex_section_depth_int > 2

```

```

5017 { \ctex_set:nn { subparagraph } { indent = \c_zero_dim } }
5018 { \ctex_set:nn { subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

5019 \ctex_set:nn { appendix }
5020 <*article>
5021 { number          = \@Alph \c@section }
5022 </article>
5023 <*book|report>
5024 {
5025     name           = \appendixname \space ,
5026     number         = \@Alph \c@chapter
5027 }
5028 </book|report>

5029 </article|book|report>

5030 </scheme&(article|book|report|beamer)>

```

15.18 chinese 方案的其他设置

```

5031 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

<article|book|report> 5032 \pagestyle { headings }

```

日期格式。

```

5033 \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。

```

5034 \ctex_if_autoindent_touched:F
5035 { \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。

```

5036 <!*generic>
5037 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
5038 <*beamer>
5039 {
5040     \uselanguage { ChineseGBK }
5041     \languagealias { chinese } { ChineseGBK }
5042     \ctex_file_input:n { ctex-name-gbk.cfg }
5043 }
5044 {
5045     \uselanguage { ChineseUTF8 }
5046     \languagealias { chinese } { ChineseUTF8 }
5047     \ctex_file_input:n { ctex-name-utf8.cfg }
5048 }

```

让 translator 包优先查找中文翻译。

```

5049 \clist_put_left:Nn \trans@languagepath { chinese }
5050 </beamer>
5051 <!*beamer>
5052 { \ctex_file_input:n { ctex-name-gbk.cfg } }
5053 { \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。

```

5054 \int_compare:nNnF \g__ctex_font_size_int > { -1 }
5055 { \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。

```
5056 \fp_if_nan:nT { \l__ctex_line_spread_fp }
5057 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
5058 </!beamer>
5059 </!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。

```
5060 <*generic>
5061 \tl_if_eq:NnTF \c__ctex_std_class_tl { beamer }
5062 {
5063   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
5064   {
5065     \uselanguage { ChineseGBK }
5066     \languagealias { chinese } { ChineseGBK }
5067     \ctex_file_input:n { ctex-name-gbk.cfg }
5068   }
5069   {
5070     \uselanguage { ChineseUTF8 }
5071     \languagealias { chinese } { ChineseUTF8 }
5072     \ctex_file_input:n { ctex-name-utf8.cfg }
5073   }
5074   \clist_put_left:Nn \trans@languagepath { chinese }
5075 }
5076 {
5077   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
5078   { \ctex_file_input:n { ctex-name-gbk.cfg } }
5079   { \ctex_file_input:n { ctex-name-utf8.cfg } }
5080   \int_compare:nNnF \g__ctex_font_size_int > { -1 }
5081   { \int_gset:Nn \g__ctex_font_size_int { 0 } }
5082   \fp_if_nan:nT { \l__ctex_line_spread_fp }
5083   { \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。

```
5084   \tl_if_exist:NT \c__ctex_std_class_tl
5085   { \RequirePackage { indentfirst } }
5086 }
5087 </generic>
5088 </scheme&chinese>
```

15.18.1 列表环境的缩进

\verse 只在使用文档类的时候修改诗歌和引用环境的缩进。
\quotation

```
5089 <*scheme&chinese&(article|book|report)>
5090 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
5091 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
5092 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
5093 </scheme&chinese&(article|book|report)>

5094 <@@=>
```

ctex-fontset.dtx

```
5095 <@@=ctex>
```

15.18.2 预定义字库

```
5096 <*fontset>
```

15.18.2.1 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。

```

5097 <*adobe>
5098 \ctex_fontset_case:nnnn
5099 { \ctex_fontset_error:n { adobe } }
5100 {
5101   \ctex_zhmap_case:nnn
5102   {
5103     \setCJKmainfont { AdobeSongStd-Light.otf }
5104     [
5105       cmap          = UniGB-UTF16-H,
5106       BoldFont      = AdobeHeitiStd-Regular.otf,
5107       ItalicFont    = AdobeKaitiStd-Regular.otf
5108     ]
5109     \setCJKsansfont { AdobeHeitiStd-Regular.otf }
5110     [ cmap = UniGB-UTF16-H ]
5111     \setCJKmonofont { AdobeFangsongStd-Regular.otf }
5112     [ cmap = UniGB-UTF16-H ]
5113     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light.otf }
5114     [ cmap = UniGB-UTF16-H ]
5115     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular.otf }
5116     [ cmap = UniGB-UTF16-H ]
5117     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular.otf }
5118     [ cmap = UniGB-UTF16-H ]
5119     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular.otf }
5120     [ cmap = UniGB-UTF16-H ]
5121     \ctex_punct_set:n { adobe }
5122     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5123     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
5124     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5125     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5126     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5127   }
5128   {
5129     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { adobe }
5130     \ctex_punct_set:n { adobe }
5131     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5132     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5133     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5134   }
5135   { \ctex_fontset_error:n { adobe } }
5136 }
5137 {
5138   \ctex_set_upfonts:nnnnnn
5139   { AdobeSongStd-Light.otf }
5140   { AdobeHeitiStd-Regular.otf }
5141   { AdobeKaitiStd-Regular.otf }
5142   { AdobeHeitiStd-Regular.otf }
5143   { AdobeHeitiStd-Regular.otf }
5144   { AdobeFangsongStd-Regular.otf }
5145   \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
5146   \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
5147   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
5148   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
5149 }
5150 {
5151   \setCJKmainfont { AdobeSongStd-Light }
5152   [ BoldFont = AdobeHeitiStd-Regular, ItalicFont = AdobeKaitiStd-Regular ]
5153   \setCJKsansfont { AdobeHeitiStd-Regular }
5154   \setCJKmonofont { AdobeFangsongStd-Regular }
5155   \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
5156   \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
5157   \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
5158   \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }

```

```

5159 }
5160 </adobe>

```

15.18.2.2 fandol

```

5161 <*fandol>
5162 \ctex_fontset_case:nnnn
5163 { \ctex_fontset_error:n { fandol } }
5164 {
5165   \ctex_zhmap_case:nnn
5166   {
5167     \setCJKmainfont { FandolSong-Regular.otf }
5168     [
5169       cmap          = UniGB-UTF16-H,
5170       BoldFont      = FandolSong-Bold.otf,
5171       ItalicFont    = FandolKai-Regular.otf
5172     ]
5173     \setCJKsansfont { FandolHei-Regular.otf }
5174     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
5175     \setCJKmonofont { FandolFang-Regular.otf }
5176     [ cmap = UniGB-UTF16-H ]
5177     \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf }
5178     [ cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf ]
5179     \setCJKfamilyfont { zhhei } { FandolHei-Regular.otf }
5180     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
5181     \setCJKfamilyfont { zhfs } { FandolFang-Regular.otf }
5182     [ cmap = UniGB-UTF16-H ]
5183     \setCJKfamilyfont { zhkai } { FandolKai-Regular.otf }
5184     [ cmap = UniGB-UTF16-H ]
5185     \ctex_punct_set:n { fandol }
5186     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5187     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
5188     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5189     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
5190     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
5191     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5192   }
5193   {
5194     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { fandol }
5195     \ctex_punct_set:n { fandol }
5196     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5197     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5198     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5199   }
5200   { \ctex_fontset_error:n { fandol } }
5201 }
5202 {
5203   \ctex_set_upfonts:nnnnnn
5204   { FandolSong-Regular.otf }
5205   { FandolSong-Bold.otf }
5206   { FandolKai-Regular.otf }
5207   { FandolHei-Regular.otf }
5208   { FandolHei-Bold.otf }
5209   { FandolFang-Regular.otf }
5210   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5211   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5212   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
5213   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
5214 }
5215 {
5216   \setCJKmainfont { FandolSong-Regular }
5217   [
5218     Extension = .otf,
5219     BoldFont  = FandolSong-Bold,
5220     ItalicFont = FandolKai-Regular
5221   ]

```

```

5222 \setCJKsansfont { FandolHei-Regular }
5223 [ Extension = .otf, BoldFont = FandolHei-Bold ]
5224 \setCJKmonofont { FandolFang-Regular }
5225 [ Extension = .otf ]
5226 \setCJKfamilyfont { zhsong } { FandolSong-Regular }
5227 [ Extension = .otf, BoldFont = FandolSong-Bold ]
5228 \setCJKfamilyfont { zhhei } { FandolHei-Regular }
5229 [ Extension = .otf, BoldFont = FandolHei-Bold ]
5230 \setCJKfamilyfont { zhfs } { FandolFang-Regular }
5231 [ Extension = .otf ]
5232 \setCJKfamilyfont { zhkai } { FandolKai-Regular }
5233 [ Extension = .otf ]
5234 }
5235 </fandol>

```

15.18.2.3 founder

```

5236 <*founder>
5237 \ctex_fontset_case:nnn
5238 {
5239   \ctex_zhmap_case:nnn
5240   {
5241     \setCJKmainfont { FZSSK.TTF }
5242     [ BoldFont = FZXBSK.TTF, ItalicFont = FZKTK.TTF ]
5243     \setCJKsansfont { FZXH1K.TTF } [ BoldFont = FZHTK.TTF ]
5244     \setCJKmonofont { FZFSK.TTF }
5245     \setCJKfamilyfont { zhsong } { FZSSK.TTF } [ BoldFont = FZXBSK.TTF ]
5246     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
5247     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
5248     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
5249     \setCJKfamilyfont { zhli } { FZLSK.TTF }
5250     \setCJKfamilyfont { zhyou } { FZY1K.TTF } [ BoldFont = FZY3K.TTF ]
5251     \ctex_punct_set:n { founder }
5252     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5253     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
5254     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5255     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5256     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
5257     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
5258     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
5259   }
5260   {
5261     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { founder }
5262     \ctex_punct_set:n { founder }
5263     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5264     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5265     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5266   }
5267   { \ctex_fontset_error:n { founder } }
5268 }
5269 {
5270   \ctex_set_upfonts:nnnnnn
5271   { FZSSK.TTF }
5272   { FZXBSK.TTF }
5273   { FZKTK.TTF }
5274   { FZXH1K.TTF }
5275   { FZHTK.TTF }
5276   { FZFSK.TTF }
5277   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5278   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5279   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
5280   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
5281   \ctex_set_upfamily:nnn { zhli } { upschrn } {}
5282   \ctex_set_upfamily:nnn { zhyou } { upschtgt } {}
5283   \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
5284   \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
5285 }

```

```

5286 {
5287   \setCJKmainfont { FZShuSong-Z01 }
5288   [ BoldFont = FZXiaoBiaoSong-B05, ItalicFont = FZKai-Z03 ]
5289   \setCJKsansfont { FZXiHeiI-Z08 } [ BoldFont = FZHei-B01 ]
5290   \setCJKmonofont { FZFangSong-Z02 }
5291   \setCJKfamilyfont { zhsong } { FZShuSong-Z01 }
5292   [ BoldFont = FZXiaoBiaoSong-B05 ]
5293   \setCJKfamilyfont { zhhei } { FZHei-B01 }
5294   \setCJKfamilyfont { zhkai } { FZKai-Z03 }
5295   \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
5296   \setCJKfamilyfont { zhli } { FZLiShu-S01 }
5297   \setCJKfamilyfont { zhyou } { FZXiYuan-M01 }
5298   [ BoldFont = FZZhunYuan-M02 ]
5299 }
5300 </founder>

```

15.18.2.4 hanyi

```

5301 <*hanyi>
5302 \ctex_fontset_case:nnn
5303 {
5304   \ctex_zhmap_case:nnn
5305   {
5306     \setCJKmainfont { HYShuSongErS.ttf }
5307     [ BoldFont = HYZhongSongS.ttf, ItalicFont = HYKaiTiS.ttf ]
5308     \setCJKsansfont { HYZhongHeiS.ttf } [ BoldFont = HYDaHeiS.ttf ]
5309     \setCJKmonofont { HYFangSongS.ttf }
5310     \setCJKfamilyfont { zhsong } { HYShuSongErS.ttf }
5311     [ BoldFont = HYZhongSongS.ttf ]
5312     \setCJKfamilyfont { zhhei } { HYZhongHeiS.ttf }
5313     [ BoldFont = HYDaHeiS.ttf ]
5314     \setCJKfamilyfont { zhkai } { HYKaiTiS.ttf }
5315     \setCJKfamilyfont { zhfs } { HYFangSongS.ttf }
5316     \ctex_punct_set:n { hanyi }
5317     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5318     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
5319     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5320     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
5321     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
5322     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5323   }
5324   {
5325     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { hanyi }
5326     \ctex_punct_set:n { hanyi }
5327     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5328     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5329     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5330   }
5331   { \ctex_fontset_error:n { hanyi } }
5332 }
5333 {
5334   \ctex_set_upfonts:nnnnnn
5335   { HYShuSongErS.ttf }
5336   { HYZhongSongS.ttf }
5337   { HYKaiTiS.ttf }
5338   { HYZhongHeiS.ttf }
5339   { HYDaHeiS.ttf }
5340   { HYFangSongS.ttf }
5341   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5342   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5343   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
5344   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
5345 }
5346 {
5347   \setCJKmainfont { HYShuSongEr~S }
5348   [ BoldFont = HYZhongSong~S, ItalicFont = HYKaiTi~S ]
5349   \setCJKsansfont { HYZhongHei~S } [ BoldFont = HYDaHei~S ]

```



```

5350 \setCJKmonofont { HYFangSong~S }
5351 \setCJKfamilyfont { zhsong } { HYShuSongEr~S }
5352 [ BoldFont = HYZhongSong~S ]
5353 \setCJKfamilyfont { zhhei } { HYZhongHei~S }
5354 [ BoldFont = HYDaHei~S ]
5355 \setCJKfamilyfont { zhkai } { HYKaiTi~S }
5356 \setCJKfamilyfont { zhfs } { HYFangSong~S }
5357 }
5358 </hanyi>

```

15.18.2.5 mac 相关

按 [Issue 351](#) 的讨论, 以 El Capitan 为分界, 分别设置 macold (El Capitan 之前) 和 macnew (El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。从 macOS 15 (Sequoia) 开始, 苹方字体转为 *downloadable*, 不再位于传统路径, 故增加版本号检测作为后备。

```

5359 <*mac>
5360 \tl_new:N \g__ctex_macos_ver_tl
5361 \msg_new:nnn { ctex } { macos-version-detect-failed }
5362 { Cannot~detect~macOS~version.~Falling~back~to~macold~fontset. }

```

为方便人类阅读, 把 LuaTeX 分支挪到后面, 在不忽略空格的 catcode scheme 下编写。这样 Lua 代码就不必以 ~ 代替空格。非 LuaTeX 分支, 在末尾使用 `\use_none_delimit_by_q_stop:w` 来跳过 LuaTeX 分支。

```

5363 \sys_if_engine_luatex:F
5364 {
5365   \ior_new:N \g__ctex_macos_ver_ior
5366   \bool_new:N \g__ctex_macos_ver_found_bool
5367   \file_if_exist:nT { /System/Library/CoreServices/SystemVersion.plist }
5368   {
5369     \ior_open:Nn \g__ctex_macos_ver_ior
5370     { /System/Library/CoreServices/SystemVersion.plist }
5371     \ior_map_inline:Nn \g__ctex_macos_ver_ior
5372     {
5373       \bool_if:NTF \g__ctex_macos_ver_found_bool
5374       {
5375         \tl_set:Nn \l_tmpa_tl {#1}
5376         \regex_replace_once:nnN
5377         { .*? <string> (\d+) .* } { \1 } \l_tmpa_tl
5378         \tl_gset:NV \g__ctex_macos_ver_tl \l_tmpa_tl
5379         \ior_map_break:
5380       }
5381     }
5382     \tl_if_in:nnT {#1} { ProductVersion }
5383     { \bool_gset_true:N \g__ctex_macos_ver_found_bool }
5384   }
5385 }
5386 \ior_close:N \g__ctex_macos_ver_ior
5387 }
5388 \use_none_delimit_by_q_stop:w
5389 }
5390 \char_set_catcode_space:n { 32 }
5391 \lua_now:n
5392 {
5393   local f = io.open("/System/Library/CoreServices/SystemVersion.plist", "r");
5394   if f then
5395     local content = f:read("*a");
5396     f:close();
5397     local ver = content:match("ProductVersion.-<string>(.)</string>");
5398     if ver then
5399       local dot = ver:find(".", 1, true);
5400       local major = dot and ver:sub(1, dot - 1) or ver;
5401       if major then
5402         token.set_macro("g__ctex_macos_ver_tl", major, "global");

```

```

5403         end;
5404     end;
5405 end
5406 }
5407 \char_set_catcode_ignore:n { 32 }
5408 \use_none_delimit_by_q_stop:w \q_stop
5409 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
5410 { \ctex_file_input:n { ctex-fontset-macnew.def } }
5411 {
5412     \tl_if_empty:NTF \g__ctex_macos_ver_tl
5413     {
5414         \msg_warning:nn { ctex } { macos-version-detect-failed }
5415         \ctex_file_input:n { ctex-fontset-macold.def }
5416     }
5417     {
5418         \int_compare:nNnTF { \g__ctex_macos_ver_tl } < { 15 }
5419         { \ctex_file_input:n { ctex-fontset-macold.def } }
5420         { \ctex_file_input:n { ctex-fontset-macnew.def } }
5421     }
5422 }
5423 </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。

```

5424 <*macold|macnew>
5425 \ctex_fontset_case:nnnn
5426 {
5427     \ctex_fontset_error:n { mac }
5428     \use_none_delimit_by_q_stop:w
5429 }
5430 <*macold>
5431 { \ctex_fontset_error:n { macold } }
5432 { \ctex_fontset_error:n { macold } }
5433 </macold>
5434 <*macnew>
5435 {
5436     \ctex_zhmap_case:nnn
5437     {
5438         \setCJKmainfont { :3:Songti.ttc }
5439         [
5440             BoldFont      = :1:Songti.ttc,
5441             ItalicFont    = :0:Kaiti.ttc,
5442             BoldItalicFont = :3:Kaiti.ttc,
5443         ]
5444         \setCJKsansfont { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
5445         \setCJKmonofont { STFANGSO.ttf }
5446         \setCJKfamilyfont { zhsong } { :3:Songti.ttc } [ BoldFont = :1:Songti.ttc ]
5447         \setCJKfamilyfont { zhhei } { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
5448         \setCJKfamilyfont { zhkai } { :0:Kaiti.ttc } [ BoldFont = :3:Kaiti.ttc ]
5449         \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
5450         \setCJKfamilyfont { zhli } { :0:Baoli.ttc }
5451         \setCJKfamilyfont { zhyou } { :4:Yuanti.ttc } [ BoldFont = :0:Yuanti.ttc ]
5452         \ctex_punct_set:n { mac }
5453         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5454         \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
5455         \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5456         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5457         \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
5458         \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhpfb }
5459         \ctex_punct_map_bfseries:nn { \CJKttdefault, zhyou } { zhyoub }
5460     }

```

```

5461 {
5462   \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { mac }
5463   \ctex_punct_set:n { mac }
5464   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5465   \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfs }
5466   \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhpfs }
5467   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5468 }
5469 { \ctex_fontset_error:n { macnew } }
5470 \use_none_delimit_by_q_stop:w
5471 }
5472 {
5473   \ctex_set_upmap:nnn { upserif } { :3:Songti.ttc } { :1:Songti.ttc }
5474   \ctex_set_upmap:nnn { upserifit } { :0:Kaiti.ttc } { }
5475   \ctex_set_upmap:nnn { upstsl } { :0:Baoli.ttc } { }
5476   \ctex_set_upmap:nnn { upstht } { :4:Yuanti.ttc } { }
5477   \ctex_set_upmap_unicode:nnn { upsans } { :2:PingFang.ttc } { :8:PingFang.ttc }
5478   \ctex_set_upmap_unicode:nnn { upmono } { STFANGSO.ttf } { }
5479   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5480   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5481   \ctex_set_upfamily:nnn { zhfs } { upzhmono } { }
5482   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
5483   \ctex_set_upfamily:nnn { zhli } { upschrn } { }
5484   \ctex_set_upfamily:nnn { zhyou } { upschtg } { }
5485   \use_none_delimit_by_q_stop:w
5486 }
5487 </macnew>
5488 {
5489 <*macold>
5490   \setCJKmainfont { STSong }
5491   [ BoldFont = STHeiti, ItalicFont = STKaiti ]
5492   \setCJKsansfont { STXihei } [ BoldFont = STHeiti ]
5493   \setCJKmonofont { STFangsong }
5494   \setCJKfamilyfont { zhsong } { STSong }
5495   \setCJKfamilyfont { zhhei } { STHeiti }
5496   \setCJKfamilyfont { zhfs } { STFangsong }
5497   \setCJKfamilyfont { zhkai } { STKaiti }
5498 </macold>
5499 <*macnew>
5500   \bool_new:N \g__ctex_mac_pingfang_bool

```

为方便人类阅读，把 LuaTeX 分支挪到 `\ctex_fontset_case:nnnn` 之后，在不忽略空格的 catcode scheme 下编写。这样 Lua 代码就不必以 `~` 代替空格。`macnew` 的其他 `\ctex_fontset_case:nnnn` 和其他 `\ctex_zhmap_case:nnn` 分支，末尾均使用 `\use_none_delimit_by_q_stop:w` 来跳过 LuaTeX 分支。

```

5501 \sys_if_engine luatex:F
5502 {
5503   \fontspec_font_if_exist:nTF { Kaiti~SC }
5504   {
5505     \setCJKmainfont { Songti~SC~Light }
5506     [
5507       BoldFont = Songti~SC~Bold ,
5508       ItalicFont = Kaiti~SC ,
5509       BoldItalicFont = Kaiti~SC~Bold ,
5510     ]
5511   }
5512   {
5513     \setCJKmainfont { Songti~SC~Light }
5514     [ BoldFont = Songti~SC~Bold ]
5515   }
5516   \setCJKfamilyfont { zhsong } { Songti~SC~Light }
5517   [ BoldFont = Songti~SC~Bold ]
5518   \setCJKfamilyfont { zhhei } { Heiti~SC~Light }
5519   [ BoldFont = Heiti~SC~Medium ]
5520   \fontspec_font_if_exist:nTF { PingFang~SC }

```

```

5521     {
5522         \setCJKsansfont { PingFang~SC }
5523         \setCJKfamilyfont { zhpfs } { PingFang~SC }
5524         \bool_gset_true:N \g__ctex_mac_pingfang_bool
5525     }
5526     { \setCJKsansfont { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ] }
5527 \fontspec_font_if_exist:nTF { STFangsong }
5528     {
5529         \setCJKmonofont { STFangsong }
5530         \setCJKfamilyfont { zhfs } { STFangsong }
5531     }
5532     { \setCJKmonofont { Heiti~SC~Light } }
5533 \fontspec_font_if_exist:nT { Kaiti~SC }
5534     {
5535         \setCJKfamilyfont { zhkai } { Kaiti~SC }
5536         [ BoldFont = Kaiti~SC~Bold ]
5537     }
5538 \fontspec_font_if_exist:nT { Baoli~SC }
5539     { \setCJKfamilyfont { zhli } { Baoli~SC } }
5540 \fontspec_font_if_exist:nT { Yuanti~SC~Light }
5541     {
5542         \setCJKfamilyfont { zhyou } { Yuanti~SC~Light }
5543         [ BoldFont = Yuanti~SC~Regular ]
5544     }
5545 \use_none_delimit_by_q_stop:w
5546 }
5547 </macnew>
5548 }
5549 <*macnew>
5550 \char_set_catcode_space:n { 32 }
5551 \lua_now:n
5552 {
5553     local lfs = require("lfs");
5554     local search_bases = {};
5555     local asset_roots = {
5556         "/System/Library/AssetsV2",
5557         "/System/Library/AssetsV2/PreinstalledAssetsV2/InstallWithOs",
5558     };
5559     for _, root in ipairs(asset_roots) do
5560         local ok, iter, obj = pcall(lfs.dir, root);
5561         if ok then
5562             for d in iter, obj do
5563                 if d:find("^com_apple_MobileAsset_Font") then
5564                     table.insert(search_bases, root .. "/" .. d);
5565                 end;
5566             end;
5567         end;
5568     end;
5569     local function find_font_dir(filename)
5570         for _, base in ipairs(search_bases) do
5571             local ok, iter, obj = pcall(lfs.dir, base);
5572             if ok then
5573                 for entry in iter, obj do
5574                     if not (entry == "." or entry == "..") then
5575                         local p = base .. "/" .. entry
5576                         .. "/AssetData/" .. filename;
5577                         if lfs.attributes(p, "mode") then
5578                             return base .. "/" .. entry .. "/AssetData/";
5579                         end;
5580                     end;
5581                 end;
5582             end;
5583         end;
5584         return nil;
5585     end;
5586     local fonts = {
5587         {"PingFang.ttc", "g__ctex_mac_pingfang_dir_t1"},

```

```

5588     {"Kaiti.ttc",    "g__ctex_mac_kaiti_dir_tl"},
5589     {"STFANGSO.ttf", "g__ctex_mac_stfangso_dir_tl"},
5590     {"Baoli.ttc",    "g__ctex_mac_baoli_dir_tl"},
5591     {"Yuanti.ttc",   "g__ctex_mac_yuanti_dir_tl"},
5592 };
5593 for _, f in ipairs(fonts) do
5594     local dir = find_font_dir(f[1]);
5595     token.set_macro(f[2], dir or "", "global");
5596 end
5597 }
5598 \char_set_catcode_ignore:n { 32 }
5599 \tl_if_empty:NTF \g__ctex_mac_kaiti_dir_tl
5600 {
5601     \setCJKmainfont { Songti~SC~Light }
5602     [ BoldFont = Songti~SC~Bold ]
5603 }
5604 {
5605     \setCJKmainfont { Songti~SC~Light }
5606     [
5607         BoldFont      = Songti~SC~Bold ,
5608         ItalicFont    = Kaiti.ttc ,
5609         ItalicFeatures = { Path = \g__ctex_mac_kaiti_dir_tl , FontIndex = 0 } ,
5610         BoldItalicFont = Kaiti.ttc ,
5611         BoldItalicFeatures = { Path = \g__ctex_mac_kaiti_dir_tl , FontIndex = 3 } ,
5612     ]
5613 }
5614 \setCJKfamilyfont { zhsong } { Songti~SC~Light }
5615 [ BoldFont = Songti~SC~Bold ]
5616 \setCJKfamilyfont { zhhei } { Heiti~SC~Light }
5617 [ BoldFont = Heiti~SC~Medium ]
5618 \tl_if_empty:NTF \g__ctex_mac_pingfang_dir_tl
5619 { \setCJKsansfont { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ] }
5620 {
5621     \setCJKsansfont { PingFang.ttc }
5622     [
5623         Path          = \g__ctex_mac_pingfang_dir_tl ,
5624         FontIndex     = 2 ,
5625         BoldFont      = PingFang.ttc ,
5626         BoldFontIndex = 8 ,
5627     ]
5628     \setCJKfamilyfont { zhpj } { PingFang.ttc }
5629     [ Path = \g__ctex_mac_pingfang_dir_tl , FontIndex = 2 ]
5630     \bool_gset_true:N \g__ctex_mac_pingfang_bool
5631 }
5632 \tl_if_empty:NTF \g__ctex_mac_stfangso_dir_tl
5633 { \setCJKmonofont { Heiti~SC~Light } }
5634 {
5635     \setCJKmonofont { STFANGSO.ttf }
5636     [ Path = \g__ctex_mac_stfangso_dir_tl ]
5637     \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
5638     [ Path = \g__ctex_mac_stfangso_dir_tl ]
5639 }
5640 \tl_if_empty:NF \g__ctex_mac_kaiti_dir_tl
5641 {
5642     \setCJKfamilyfont { zhkai } { Kaiti.ttc }
5643     [
5644         Path          = \g__ctex_mac_kaiti_dir_tl ,
5645         FontIndex     = 0 ,
5646         BoldFont      = Kaiti.ttc ,
5647         BoldFontIndex = 3 ,
5648     ]
5649 }
5650 \tl_if_empty:NF \g__ctex_mac_baoli_dir_tl
5651 {
5652     \setCJKfamilyfont { zhli } { Baoli.ttc }
5653     [ Path = \g__ctex_mac_baoli_dir_tl , FontIndex = 0 ]
5654 }

```

```

5655 \tl_if_empty:NF \g__ctex_mac_yuanti_dir_tl
5656 {
5657   \setCJKfamilyfont { zhyou } { Yuanti.ttc }
5658   [
5659     Path          = \g__ctex_mac_yuanti_dir_tl ,
5660     FontIndex     = 4 ,
5661     BoldFont      = Yuanti.ttc ,
5662     BoldFontIndex = 0 ,
5663   ]
5664 }
5665 \use_none_delimit_by_q_stop:w \q_stop
5666 </macnew>
5667 </macold|macnew>

```

15.18.2.6 ubuntu

仿宋在 $\text{X}\text{T}_{\text{E}}\text{X}$ 和 $\text{LuaT}_{\text{E}}\text{X}$ 下按朱雀仿宋 (lxgw-fonts 宏包)、FandolFang、思源宋体的顺序检测选用, 等宽字体 (\ttfamily) 与之保持一致。其他编译方式无法在运行时检测字体, 仿宋直接使用静态映射的思源宋体。

```

5668 <*ubuntu>
5669 \ctex_fontset_case:nnnn
5670 { \ctex_fontset_error:n { ubuntu } }
5671 {
5672   \ctex_zhmap_case:nnn
5673   {
5674     \setCJKmainfont { :2:NotoSerifCJK-Regular.ttc }
5675     [ BoldFont = :2:NotoSerifCJK-Bold.ttc, ItalicFont = gkai00mp.ttf ]
5676     \setCJKsansfont { :2:NotoSansCJK-Regular.ttc }
5677     [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
5678     \setCJKmonofont { :2:NotoSerifCJK-Regular.ttc }
5679     [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
5680     \setCJKfamilyfont { zhsong } { :2:NotoSerifCJK-Regular.ttc }
5681     [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
5682     \setCJKfamilyfont { zhhei } { :2:NotoSansCJK-Regular.ttc }
5683     [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
5684     \setCJKfamilyfont { zhfs } { :2:NotoSerifCJK-Regular.ttc }
5685     \setCJKfamilyfont { zhkai } { gkai00mp.ttf }
5686     \ctex_punct_set:n { ubuntu }
5687     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5688     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
5689     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
5690     \ctex_punct_map_family:nn { zhfs } { zhsong }
5691     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5692     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
5693     \ctex_punct_map_bfseries:nn
5694     { \CJKrmdefault, \CJKttdefault, zhsong }
5695     { zhsongb }
5696   }
5697   {
5698     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { ubuntu }
5699     \ctex_punct_set:n { ubuntu }
5700     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5701     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
5702     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5703     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5704   }
5705   { \ctex_fontset_error:n { ubuntu } }
5706 }
5707 {
5708   \ctex_set_upmap_unicode:nnn { upserif }
5709   { :2:NotoSerifCJK-Regular.ttc } { :2:NotoSerifCJK-Bold.ttc }
5710   \ctex_set_upmap_unicode:nnn { upsans }
5711   { :2:NotoSansCJK-Regular.ttc } { :2:NotoSansCJK-Bold.ttc }
5712   \ctex_set_upmap_unicode:nnn { upmono }

```

```

5713     { :2:NotoSerifCJK-Regular.ttc } { }
5714     \ctex_set_upmap:nnn { upserifit } { gkai00mp.ttf } { }
5715     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5716     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5717     \ctex_set_upfamily:nnn { zhfs } { upzhmono } { }
5718     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
5719 }
5720 {
5721     \setCJKmainfont { Noto~Serif~CJK~SC }
5722     [ BoldFont = Noto~Serif~CJK~SC~Bold, ItalicFont = AR~PL~KaitiM~GB ]
5723     \setCJKsansfont { Noto~Sans~CJK~SC }
5724     [ BoldFont = Noto~Sans~CJK~SC~Bold ]
5725     \fontspec_font_if_exist:nTF { LXGWZhuqueFangsong-Regular.ttf }
5726     {
5727         \setCJKmonofont { LXGWZhuqueFangsong-Regular.ttf }
5728         \setCJKfamilyfont { zhfs } { LXGWZhuqueFangsong-Regular.ttf }
5729     }
5730     {
5731         \fontspec_font_if_exist:nTF { FandolFang-Regular.otf }
5732         {
5733             \setCJKmonofont { FandolFang-Regular.otf }
5734             \setCJKfamilyfont { zhfs } { FandolFang-Regular.otf }
5735         }
5736         {
5737             \setCJKmonofont { Noto~Serif~CJK~SC }
5738             \setCJKfamilyfont { zhfs } { Noto~Serif~CJK~SC }
5739         }
5740     }
5741     \setCJKfamilyfont { zhsong } { Noto~Serif~CJK~SC }
5742     [ BoldFont = Noto~Serif~CJK~SC~Bold ]
5743     \setCJKfamilyfont { zhhei } { Noto~Sans~CJK~SC }
5744     [ BoldFont = Noto~Sans~CJK~SC~Bold ]
5745     \setCJKfamilyfont { zhkai } { AR~PL~KaitiM~GB }
5746 }
5747 </ubuntu>

```

15.18.2.7 windows

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

```

5748 <*windows>
5749 \file_if_exist:nTF { \c_dollar_str WINDIR/Fonts/msyh.ttc }
5750 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
5751 {
5752     \file_if_exist:nTF { msyh.ttc }
5753     { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
5754     { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttf } }
5755 }

5756 \ctex_fontset_case:nnn
5757 {
5758     \ctex_zhmap_case:nnn
5759     {
5760         \ctex_punct_set:n { windows }
5761         \setCJKmainfont { simsun.ttc }
5762         [ BoldFont = simhei.ttf, ItalicFont = simkai.ttf ]
5763         \setCJKsansfont { msyh.\c__ctex_msyh_suffix_tl }
5764         [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
5765         \setCJKmonofont { simfang.ttf }
5766         \setCJKfamilyfont { zhsong } { simsun.ttc }
5767         \setCJKfamilyfont { zhhei } { simhei.ttf }
5768         \setCJKfamilyfont { zhfs } { simfang.ttf }
5769         \setCJKfamilyfont { zhkai } { simkai.ttf }
5770         \setCJKfamilyfont { zhyahei } { msyh.\c__ctex_msyh_suffix_tl }
5771         [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
5772         \setCJKfamilyfont { zhli } { simli.ttf }

```

```

5773 \setCJKfamilyfont { zhyou } { simyou.ttf }
5774 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5775 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5776 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5777 \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
5778 \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhyahei } { zhyaheib }
5779 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5780 }
5781 {
5782 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { windows }
5783 \ctex_punct_set:n { windows }
5784 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5785 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5786 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5787 }
5788 { \ctex_fontset_error:n { windows } }
5789 }
5790 {
5791 \ctex_set_upfonts:nnnnnn
5792 { simsun.ttc }
5793 { simhei.ttf }
5794 { simkai.ttf }
5795 { msyh.\c__ctex_msyh_suffix_tl }
5796 { msyhbd.\c__ctex_msyh_suffix_tl }
5797 { simfang.ttf }
5798 \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
5799 \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
5800 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
5801 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
5802 \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
5803 \ctex_set_upfamily:nnn { zhli } { upschrn } {}
5804 \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
5805 \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
5806 \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
5807 }
5808 {
5809 \setCJKmainfont { SimSun } [ BoldFont = SimHei , ItalicFont = KaiTi ]
5810 \setCJKsansfont { MicrosoftYaHei } [ BoldFont = *~Bold ]
5811 \setCJKmonofont { FangSong }
5812 \setCJKfamilyfont { zhsong } { SimSun }
5813 \setCJKfamilyfont { zhhei } { SimHei }
5814 \setCJKfamilyfont { zhfs } { FangSong }
5815 \setCJKfamilyfont { zhkai } { KaiTi }
5816 \setCJKfamilyfont { zhyahei } { MicrosoftYaHei } [ BoldFont = *~Bold ]
5817 \setCJKfamilyfont { zhli } { LiSu }
5818 \setCJKfamilyfont { zhyou } { YouYuan }
5819 }
5820 </windows>

```

15.18.3 中文字体命令

使用 $\text{\LaTeX}$ 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 $\text{\textbackslash yahei}$ 、 $\text{\textbackslash pingfang}$ 命令被设置为与 $\text{\textbackslash heiti}$ 相同。

```

\songti
\heiti
\fangsong
\kaishu
\lishu
\youyuan
\yahei
\pingfang
5821 <*!mac>
5822 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
5823 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
5824 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
5825 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
5826 <*windows|founder|macnew>
5827 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
5828 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
5829 </windows|founder|macnew>
<windows> 5830 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
5831 <*macnew>

```



```

5832 \bool_lazy_or:nnTF
5833 { \sys_if_engine_pdftex_p: }
5834 { \sys_if_engine_uptex_p: }
5835 {
5836   \cs_new_eq:NN \yahei \heiti
5837   \cs_new_eq:NN \pingfang \heiti
5838 }
5839 {
5840   \bool_if:NTF \g__ctex_mac_pingfang_bool
5841   {
5842     \NewDocumentCommand \yahei { } { \CJKfamily { zhp } }
5843     \NewDocumentCommand \pingfang { } { \CJKfamily { zhp } }
5844   }
5845   {
5846     \cs_new_eq:NN \yahei \heiti
5847     \cs_new_eq:NN \pingfang \heiti
5848   }
5849 }
5850 </macnew>
5851 </!mac>
5852 </fontset>

```

15.18.4 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

5853 <zhmap>
5854 \begingroup\catcode61\catcode48\catcode32=10\relax%
5855 \catcode 35=6 % #
5856 \catcode 45=12 % -
5857 \catcode123=1 % {
5858 \catcode125=2 % }
5859 \toks0{\endlinechar=\the\endlinechar\relax}%
5860 \toks2{\endlinechar=-1}%
5861 \def\x#1 #2 {%
5862   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
5863   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
5864 \x 13 5 % carriage return
5865 \x 32 10 % space
5866 \x 35 6 % #
5867 \x 40 12 % (
5868 \x 41 12 % )
5869 \x 45 12 % -
5870 \x 46 12 % .
5871 \x 47 12 % /
5872 \x 58 12 % :
5873 \x 60 12 % <
5874 \x 61 12 % =
5875 \x 64 11 % @
5876 \x 91 12 % [
5877 \x 93 12 % ]
5878 \x 123 1 % {
5879 \x 125 2 % }
5880 \edef\x#1{\endgroup%
5881   \edef\noexpand#1{%
5882     \the\toks0 %
5883     \let\noexpand\noexpand\noexpand#1%
5884     \noexpand\noexpand\noexpand\noexpand\undefined%
5885     \noexpand\noexpand\noexpand\noexpand\endinput}%
5886   \the\toks2}%
5887 \expandafter\x\csname ctex@zhmap@endinput\endcsname
5888 \ifzhmappdf \begingroup\expandafter\endgroup
5889 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
5890 \expandafter\ifx\csname ifctexpdf\endcsname\relax

```

```

5891 \expandafter\ifx\csname pdfoutput\endcsname\relax
5892   iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
5893 \else ifctexpdf\fi
5894 \endcsname

```

\ProvidesFile 提供非 L^AT_EX 格式下的 \ProvidesFile。

```

5895 \begingroup
5896 \expandafter\ifx\csname ProvidesFile\endcsname\relax
5897 \long\def\x#1\ProvidesFile#2[#3]{%
5898   #1%
5899   \immediate\write-1{File: #2 #3}%
5900   \expandafter\xdef\csname ver@#2\endcsname{#3}}
5901 \expandafter\x%
5902 \fi
5903 \endgroup

```

文件标识信息。

```

5904 <*zhmap>
<adobe> 5905 \ProvidesFile{ctex-zhmap-adobe.tex}%
<fandol> 5906 \ProvidesFile{ctex-zhmap-fandol.tex}%
<founder> 5907 \ProvidesFile{ctex-zhmap-founder.tex}%
<hanyi> 5908 \ProvidesFile{ctex-zhmap-hanyi.tex}%
<mac> 5909 \ProvidesFile{ctex-zhmap-mac.tex}%
<ubuntu> 5910 \ProvidesFile{ctex-zhmap-ubuntu.tex}%
<windows> 5911 \ProvidesFile{ctex-zhmap-windows.tex}%
<zhmap> 5912 [\ExplFileDate\space v\ExplFileVersion\space \ExplFileDescription]
5913 </zhmap>

```

15.18.4.1 ctex-zhmap-adobe.tex

```

5914 <*adobe>
5915 \ifzhmappdf
5916 %% pdfTeX does not support OTF fonts
5917 \else
5918 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5919 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5920 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
5921 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
5922 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
5923 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5924 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5925 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
5926 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
5927 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
5928 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
5929 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
5930 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
5931 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
5932 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
5933 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
5934 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
5935 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
5936 \fi
5937 </adobe>

```

15.18.4.2 ctex-zhmap-fandol.tex

```

5938 <*fandol>
5939 \ifzhmappdf
5940 %% pdfTeX does not support OTF fonts
5941 \else
5942 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
5943 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
5944 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}

```

```

5945 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
5946 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
5947 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
5948 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
5949 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
5950 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
5951 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
5952 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
5953 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
5954 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
5955 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
5956 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
5957 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
5958 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
5959 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
5960 \fi
5961 </fandol>

```

15.18.4.3 ctex-zhmap-founder.tex

```

5962 <*founder>
5963 \ifzhmappdf
5964 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF>
5965 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF>
5966 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF>
5967 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF>
5968 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF>
5969 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF>
5970 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF>
5971 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF>
5972 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF>
5973 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF>
5974 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF>
5975 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF>
5976 \pdfmapline{=unili@Unicode@ <FZLSK.TTF>
5977 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF>
5978 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF>
5979 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF>
5980 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF>
5981 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF>
5982 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF>
5983 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF>
5984 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF>
5985 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF>
5986 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF>
5987 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF>
5988 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF>
5989 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF>
5990 \else
5991 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
5992 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
5993 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
5994 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
5995 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
5996 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
5997 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
5998 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
5999 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
6000 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
6001 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
6002 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
6003 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
6004 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF}
6005 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
6006 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
6007 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
6008 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}

```

```

6009 \special{pdf:mapline gbklisl@UGBK@      unicode FZLSK.TTF -s .167}
6010 \special{pdf:mapline gbkyousl@UGBK@      unicode FZY1K.TTF -s .167}
6011 \special{pdf:mapline unisongsl@Unicode@    unicode FZSSK.TTF -s .167}
6012 \special{pdf:mapline unikaisl@Unicode@    unicode FZKTK.TTF -s .167}
6013 \special{pdf:mapline uniheisl@Unicode@    unicode FZHTK.TTF -s .167}
6014 \special{pdf:mapline unifssl@Unicode@     unicode FZFSK.TTF -s .167}
6015 \special{pdf:mapline unilisl@Unicode@     unicode FZLSK.TTF -s .167}
6016 \special{pdf:mapline uniyousl@Unicode@    unicode FZY1K.TTF -s .167}
6017 \fi
6018 </founder>

```

15.18.4.4 ctex-zhmap-hanyi.tex

```

6019 <*hanyi>
6020 \ifzhmappdf
6021 \pdfmapline{=gbk@UGBK@      <HYShuSongErS.ttf}
6022 \pdfmapline{=gbksong@UGBK@  <HYShuSongErS.ttf}
6023 \pdfmapline{=gbkkai@UGBK@   <HYKaiTiS.ttf}
6024 \pdfmapline{=gbkhei@UGBK@   <HYZhongHeiS.ttf}
6025 \pdfmapline{=gbkfs@UGBK@    <HYFangSongS.ttf}
6026 \pdfmapline{=cyberb@Unicode@ <HYShuSongErS.ttf}
6027 \pdfmapline{=unisong@Unicode@ <HYShuSongErS.ttf}
6028 \pdfmapline{=unikai@Unicode@ <HYKaiTiS.ttf}
6029 \pdfmapline{=unihei@Unicode@ <HYZhongHeiS.ttf}
6030 \pdfmapline{=unifs@Unicode@  <HYFangSongS.ttf}
6031 \pdfmapline{=gbksongsl@UGBK@ <HYShuSongErS.ttf}
6032 \pdfmapline{=gbkkaisl@UGBK@  <HYKaiTiS.ttf}
6033 \pdfmapline{=gbkheisl@UGBK@  <HYZhongHeiS.ttf}
6034 \pdfmapline{=gbkfssl@UGBK@   <HYFangSongS.ttf}
6035 \pdfmapline{=unisongsl@Unicode@ <HYShuSongErS.ttf}
6036 \pdfmapline{=unikaisl@Unicode@ <HYKaiTiS.ttf}
6037 \pdfmapline{=uniheisl@Unicode@ <HYZhongHeiS.ttf}
6038 \pdfmapline{=unifssl@Unicode@  <HYFangSongS.ttf}
6039 \else
6040 \special{pdf:mapline gbk@UGBK@      unicode HYShuSongErS.ttf}
6041 \special{pdf:mapline gbksong@UGBK@   unicode HYShuSongErS.ttf}
6042 \special{pdf:mapline gbkkai@UGBK@    unicode HYKaiTiS.ttf}
6043 \special{pdf:mapline gbkhei@UGBK@    unicode HYZhongHeiS.ttf}
6044 \special{pdf:mapline gbkfs@UGBK@     unicode HYFangSongS.ttf}
6045 \special{pdf:mapline cyberb@Unicode@  unicode HYShuSongErS.ttf}
6046 \special{pdf:mapline unisong@Unicode@ unicode HYShuSongErS.ttf}
6047 \special{pdf:mapline unikai@Unicode@  unicode HYKaiTiS.ttf}
6048 \special{pdf:mapline unihei@Unicode@  unicode HYZhongHeiS.ttf}
6049 \special{pdf:mapline unifs@Unicode@   unicode HYFangSongS.ttf}
6050 \special{pdf:mapline gbksongsl@UGBK@   unicode HYShuSongErS.ttf -s .167}
6051 \special{pdf:mapline gbkkaisl@UGBK@    unicode HYKaiTiS.ttf -s .167}
6052 \special{pdf:mapline gbkheisl@UGBK@    unicode HYZhongHeiS.ttf -s .167}
6053 \special{pdf:mapline gbkfssl@UGBK@     unicode HYFangSongS.ttf -s .167}
6054 \special{pdf:mapline unisongsl@Unicode@ unicode HYShuSongErS.ttf -s .167}
6055 \special{pdf:mapline unikaisl@Unicode@ unicode HYKaiTiS.ttf -s .167}
6056 \special{pdf:mapline uniheisl@Unicode@ unicode HYZhongHeiS.ttf -s .167}
6057 \special{pdf:mapline unifssl@Unicode@  unicode HYFangSongS.ttf -s .167}
6058 \fi
6059 </hanyi>

```

15.18.4.5 ctex-zhmap-mac.tex

```

6060 <*mac>
6061 \ifzhmappdf
6062 %% pdfTeX does not support OTF fonts
6063 \else
6064 \special{pdf:mapline gbk@UGBK@      UniGB-UTF16-H :3:Songti.ttc}
6065 \special{pdf:mapline gbksong@UGBK@   UniGB-UTF16-H :3:Songti.ttc}
6066 \special{pdf:mapline gbkkai@UGBK@    UniGB-UTF16-H :0:Kaiti.ttc}
6067 \special{pdf:mapline gbkhei@UGBK@    unicode :2:PingFang.ttc}
6068 \special{pdf:mapline gbkfs@UGBK@     unicode STFANGSO.ttf}
6069 \special{pdf:mapline gbkli@UGBK@     UniGB-UTF16-H :0:Baoli.ttc}

```

```

6070 \special{pdf:mapline gbkyou@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc}
6071 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
6072 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
6073 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc}
6074 \special{pdf:mapline unihei@Unicode@ unicode :2:PingFang.ttc}
6075 \special{pdf:mapline unifs@Unicode@ unicode STFANGSO.ttf}
6076 \special{pdf:mapline unili@Unicode@ UniGB-UTF16-H :0:Baoli.ttc}
6077 \special{pdf:mapline uniyoud@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc}
6078 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H :3:Songti.ttc -s .167}
6079 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
6080 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:PingFang.ttc -s .167}
6081 \special{pdf:mapline gbkfssl@UGBK@ unicode STFANGSO.ttf -s .167}
6082 \special{pdf:mapline gbklisl@UGBK@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
6083 \special{pdf:mapline gbkyousl@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
6084 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H :3:Songti.ttc -s .167}
6085 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
6086 \special{pdf:mapline uniheisl@Unicode@ unicode :2:PingFang.ttc -s .167}
6087 \special{pdf:mapline unifssl@Unicode@ unicode STFANGSO.ttf -s .167}
6088 \special{pdf:mapline unilisl@Unicode@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
6089 \special{pdf:mapline uniyousl@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
6090 \fi
6091 </mac>

```

15.18.4.6 ctex-zhmap-ubuntu.tex

```

6092 <*ubuntu>
6093 \ifzhmappdf
6094 %% pdfTeX does not support OTF fonts
6095 \else
6096 \special{pdf:mapline gbk@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
6097 \special{pdf:mapline gbksong@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
6098 \special{pdf:mapline gbkkai@UGBK@ unicode gkai00mp.ttf}
6099 \special{pdf:mapline gbkhei@UGBK@ unicode :2:NotoSansCJK-Regular.ttc}
6100 \special{pdf:mapline gbkfs@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
6101 \special{pdf:mapline cyberb@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
6102 \special{pdf:mapline unisong@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
6103 \special{pdf:mapline unikai@Unicode@ unicode gkai00mp.ttf}
6104 \special{pdf:mapline unihei@Unicode@ unicode :2:NotoSansCJK-Regular.ttc}
6105 \special{pdf:mapline unifs@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
6106 \special{pdf:mapline gbksongsl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
6107 \special{pdf:mapline gbkkaisl@UGBK@ unicode gkai00mp.ttf -s .167}
6108 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
6109 \special{pdf:mapline gbkfssl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
6110 \special{pdf:mapline unisongsl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
6111 \special{pdf:mapline unikaisl@Unicode@ unicode gkai00mp.ttf -s .167}
6112 \special{pdf:mapline uniheisl@Unicode@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
6113 \special{pdf:mapline unifssl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
6114 \fi
6115 </ubuntu>

```

15.18.4.7 ctex-zhmap-windows.tex

```

6116 <*windows>
6117 \ifzhmappdf
6118 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
6119 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
6120 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
6121 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
6122 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
6123 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
6124 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
6125 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
6126 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
6127 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
6128 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
6129 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
6130 \pdfmapline{=unili@Unicode@ <simli.ttf}
6131 \pdfmapline{=uniyoud@Unicode@ <simyou.ttf}

```



```

6132 \pdfmapline{=gbksongsl@UGBK@ <simsum.ttc}
6133 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
6134 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
6135 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
6136 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
6137 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
6138 \pdfmapline{=unisongsl@Unicode@ <simsum.ttc}
6139 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
6140 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
6141 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
6142 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
6143 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
6144 \else
6145 \special{pdf:mapline gbk@UGBK@ unicode :0:simsum.ttc -v 50}
6146 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsum.ttc -v 50}
6147 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
6148 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
6149 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
6150 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
6151 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
6152 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsum.ttc -v 50}
6153 \special{pdf:mapline unisong@Unicode@ unicode :0:simsum.ttc -v 50}
6154 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
6155 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
6156 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
6157 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
6158 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
6159 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsum.ttc -v 50 -s .167}
6160 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -v 70 -s .167}
6161 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -v 150 -s .167}
6162 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -v 50 -s .167}
6163 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -v 150 -s .167}
6164 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -v 60 -s .167}
6165 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsum.ttc -v 50 -s .167}
6166 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -v 70 -s .167}
6167 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -v 150 -s .167}
6168 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -v 50 -s .167}
6169 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -v 150 -s .167}
6170 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -v 60 -s .167}
6171 \fi
6172 </windows>

6173 \ctex@zhmap@endinput
6174 </zhmap>

```

15.18.5 制作 spa 文件

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

```

6175 <*spa>
6176 <*macro>
6177 \input expl3-generic %
6178 \ExplSyntaxOn
6179 \sys_if_engine_xetex:F
6180 {
6181   \msg_new:nnn { ctex } { xetex }
6182   { XeTeX is required to compile this document! }
6183   \msg_fatal:nn { ctex } { xetex }
6184 }

```

CJKpunct 定义的标点符号是:

“ 「 『 [([{ < 《 【
 —…、 。 , . : ; ! ? %)] } > 》 】 ” ” 』

注意顺序不能改变。

```

6185 \seq_const_from_clist:Nn \c__ctex_punct_seq
6186 {
6187   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
6188   "3008 , "300A , "3016 , "3010 ,
6189   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
6190   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
6191   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
6192 }

```

`\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。

```

6193 \cs_new_protected:Npn \ctex_make_spa:nn #1#2
6194 {
6195   \iow_open:Nn \g__ctex_spa_iow {#1}
6196   \clist_map_inline:nn {#2}
6197   { \__ctex_write_family:nn ##1 }
6198   \iow_close:N \g__ctex_spa_iow
6199 }
6200 \iow_new:N \g__ctex_spa_iow
6201 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

6202 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
6203 {
6204   \group_begin:
6205   \__ctex_if_font_exists:nTF {#2}
6206   { \tex_font:D \l__ctex_punct_font = "[#2]" ~ at ~ 100 pt \scan_stop: }
6207   { \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop: }
6208   \l__ctex_punct_font
6209   \clist_clear:N \l__ctex_punct_bounds_clist
6210   \seq_map_inline:Nn \c__ctex_punct_seq
6211   {
6212     \exp_args:No \__ctex_save_bounds:n
6213     { \int_use:N \tex_XeTeXcharglyph:D ##1 }
6214   }
6215   \iow_now:Ne \g__ctex_spa_iow
6216   {
6217     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

6218       { \l__ctex_punct_bounds_clist , , , }
6219     }
6220   \group_end:
6221 }

```

`\__ctex_if_font_exists:nTF` 优先搜索字体是否为 TeX Tree 字体或是否存在于当前路径下²¹, 如果不存在, `\__ctex_write_family:nn` 会根据结果在 shell 中调用 `kpathsea` 查文件或 `fontconfig` 查名称²²。

```

6222 \prg_new_protected_conditional:Nnn \__ctex_if_font_exists:n { TF }
6223 {
6224   \tl_clear:N \l__ctex_if_otf_exists_tl
6225   \tl_clear:N \l__ctex_if_ttf_exists_tl
6226   \bool_case:nF
6227   {
6228     { \file_if_exist_p:n { #1.otf } }
6229     { \tl_set:Nn \l__ctex_if_otf_exists_tl {#1} }
6230     { \file_if_exist_p:n { #1.ttf } }
6231     { \tl_set:Nn \l__ctex_if_ttf_exists_tl {#1} }
6232   }
6233   {

```

²¹事实上 TEXINPUTS 也会被一并优先搜索。

²²如果用户把字体目录加入到了 TEXINPUTS 中, `\file_if_exist_p:n` 可能找到一个 X_YTeX [basename] 加载不了的文件, 因为 X_YTeX 搜索字体专属路径 OPENTYPEFONTS/ TTFONTS, 不搜索 TEXINPUTS。

```

6234 \sys_get_shell:nnN { kpsewhich ~ #1.otf } { \endlinechar=-1 }
6235 \l__ctex_if_otf_exists_tl
6236 \sys_get_shell:nnN { kpsewhich ~ #1.ttf } { \endlinechar=-1 }
6237 \l__ctex_if_ttf_exists_tl
6238 }
6239 \bool_lazy_and:nnTF
6240 { \tl_if_blank_p:V \l__ctex_if_otf_exists_tl }
6241 { \tl_if_blank_p:V \l__ctex_if_ttf_exists_tl }
6242 { \prg_return_false: } { \prg_return_true: }
6243 }

```

\l__ctex_if_otf_exists_tl 用于存储 shell 输出的 kpsewhich 结果。

\l__ctex_if_ttf_exists_tl

```

6244 \tl_clear_new:N \l__ctex_if_otf_exists_tl
6245 \tl_clear_new:N \l__ctex_if_ttf_exists_tl

6246 \cs_new_protected:Npn \__ctex_save_bounds:n #1
6247 {
6248   \clist_put_right:Ne \l__ctex_punct_bounds_clist
6249   {
6250     \__ctex_calc_bounds:nn { 1 } {#1} ,
6251     \__ctex_calc_bounds:nn { 3 } {#1}
6252   }
6253 }
6254 \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。

```

6255 \cs_new:Npn \__ctex_calc_bounds:nn #1#2
6256 {
6257   \fp_eval:n
6258   {
6259     round
6260     (
6261       \dim_to_decimal_in_unit:nn
6262       { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
6263       { 1 em }
6264     )
6265   }
6266 }
6267 \ExplSyntaxOff
6268 </macro>

```

下面是 CT_EX 定义的一些字体。

```

6269 <*make>
6270 \input ctex-spa-macro %
6271 \MAKESPA {ctexpunct.spa}
6272 {
6273   {adobezhsong}      {AdobeSongStd-Light} ,
6274   {adobezhhei}       {AdobeHeitiStd-Regular} ,
6275   {adobezhkai}       {AdobeKaitiStd-Regular} ,
6276   {adobezhfs}        {AdobeFangsongStd-Regular} ,
6277   %
6278   {fandolzhsong}     {FandolSong-Regular} ,
6279   {fandolzhsongb}    {FandolSong-Bold} ,
6280   {fandolzhhei}      {FandolHei-Regular} ,
6281   {fandolzhheib}     {FandolHei-Bold} ,
6282   {fandolzhkai}      {FandolKai-Regular} ,
6283   {fandolzhfs}       {FandolFang-Regular} ,
6284   %
6285   {founderzhsong}    {FZShuSong-Z01} ,
6286   {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
6287   {founderzhhei}     {FZHei-B01} ,
6288   {founderzhheil}    {FZXiHeiI-Z08} ,
6289   {founderzhkai}     {FZKai-Z03} ,
6290   {founderzhfs}      {FZFangSong-Z02} ,

```



```

6291 {founderzhli} {FZLiShu-S01} ,
6292 {founderzhyou} {FZXiYuan-M01} ,
6293 {founderzhyoub} {FZZhunYuan-M02} ,
6294 %
6295 {hanyizhsong} {HYShuSongEr S} ,
6296 {hanyizhsongb} {HYZhongSong S} ,
6297 {hanyizhhei} {HYZhongHei S} ,
6298 {hanyizhheib} {HYDaHei S} ,
6299 {hanyizhkai} {HYKaiTi S} ,
6300 {hanyizhfs} {HYFangSong S} ,
6301 %
6302 {maczhsong} {Songti SC Light} ,
6303 {maczhsongb} {Songti SC Bold} ,
6304 {maczhhei} {Heiti SC Medium} ,
6305 {maczhheil} {Heiti SC Light} ,
6306 {maczhkai} {Kaiti SC} ,
6307 {maczhkaib} {Kaiti SC Bold} ,
6308 {maczhfs} {STFangsong} ,
6309 {maczhli} {Baoli SC} ,
6310 {maczhyou} {Yuanti SC Light} ,
6311 {maczhyoub} {Yuanti SC Regular} ,
6312 {maczhpf} {PingFang SC} ,
6313 {maczhpfb} {PingFang SC Semibold} ,
6314 %
6315 {ubuntuzhsong} {Noto Serif CJK SC} ,
6316 {ubuntuzhsongb} {Noto Serif CJK SC Bold} ,
6317 {ubuntuzhhei} {Noto Sans CJK SC} ,
6318 {ubuntuzhheib} {Noto Sans CJK SC Bold} ,
6319 {ubuntuzhkai} {AR PL KaitiM GB} ,
6320 %
6321 {windowszhsong} {SimSun} ,
6322 {windowszhhei} {SimHei} ,
6323 {windowszhkai} {KaiTi} ,
6324 {windowszhfs} {FangSong} ,
6325 {windowszhli} {LiSu} ,
6326 {windowszhyou} {YouYuan} ,
6327 {windowszhyahei} {Microsoft YaHei} ,
6328 {windowszhyaheib} {Microsoft YaHei Bold}
6329 }
6330 \primitive\end
6331 </make>
6332 </spa>

```

15.19 字体定义文件

15.19.1 传统定义方式

```

6333 <*c19|c70>
6334 %%
6335 %% Chinese characters
6336 %%
<c19> 6337 %% character set: GBK (extension of GB 2312)
<c70> 6338 %% character set: Unicode
6339 %% font encoding: Unicode
6340 %%
6341 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm&c19> 6342 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm&c70> 6343 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf&c19> 6344 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 6345 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 6346 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt&c70> 6347 \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> 6348 \DeclareKanjiFamily{JY2}{zhrm}{}
<rm&jt2> 6349 \DeclareKanjiFamily{JT2}{zhrm}{}
<sf&jy2> 6350 \DeclareKanjiFamily{JY2}{zhstf}{}
<sf&jt2> 6351 \DeclareKanjiFamily{JT2}{zhstf}{}
<tt&jy2> 6352 \DeclareKanjiFamily{JY2}{zhtt}{}
<tt&jt2> 6353 \DeclareKanjiFamily{JT2}{zhtt}{}

```

在 upL^AT_EX 字体族 zh_{rm} 下采用如下 Fallback 机制:形状 sl 优先保留加粗字重,形状 it 优先保留楷书字形。

```

6354 <*rm>
6355 <*c19>
6356 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
6357 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
6358 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
6359 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
6360 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
6361 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
6362 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
6363 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
6364 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
6365 </c19>
6366 <*c70>
6367 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
6368 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
6369 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
6370 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
6371 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
6372 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
6373 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
6374 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
6375 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
6376 </c70>
6377 <*jy2>
6378 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{}
6379 \DeclareFontShape{JY2}{zhrm}{b}{n}{<-> upzhserifb-h}{}
6380 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{}
6381 \DeclareFontShape{JY2}{zhrm}{m}{sl}{<-> upzhserifit-h}{}
6382 \DeclareFontShape{JY2}{zhrm}{b}{sl}{<-> upzhserifb-h}{}
6383 \DeclareFontShape{JY2}{zhrm}{bx}{sl}{<-> upzhserifb-h}{}
6384 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{}
6385 \DeclareFontShape{JY2}{zhrm}{b}{it}{<-> upzhserifit-h}{}
6386 \DeclareFontShape{JY2}{zhrm}{bx}{it}{<-> upzhserifit-h}{}
6387 </jy2>
6388 <*jt2>
6389 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{}
6390 \DeclareFontShape{JT2}{zhrm}{b}{n}{<-> upzhserifb-v}{}
6391 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{}
6392 \DeclareFontShape{JT2}{zhrm}{m}{sl}{<-> upzhserifit-v}{}
6393 \DeclareFontShape{JT2}{zhrm}{b}{sl}{<-> upzhserifb-v}{}
6394 \DeclareFontShape{JT2}{zhrm}{bx}{sl}{<-> upzhserifb-v}{}
6395 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{}
6396 \DeclareFontShape{JT2}{zhrm}{b}{it}{<-> upzhserifit-v}{}
6397 \DeclareFontShape{JT2}{zhrm}{bx}{it}{<-> upzhserifit-v}{}
6398 </jt2>
6399 </rm>

6400 <*sf>
6401 <*c19>
6402 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
6403 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
6404 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
6405 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
6406 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
6407 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}

```

```

6408 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
6409 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
6410 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
6411 </c19>
6412 <*c70>
6413 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
6414 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
6415 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
6416 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
6417 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
6418 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
6419 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
6420 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
6421 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
6422 </c70>
6423 <*jy2>
6424 \DeclareFontShape{JY2}{zhfsf}{m}{n}{<-> upzhsans-h}{\}
6425 \DeclareFontShape{JY2}{zhfsf}{b}{n}{<-> upzhsansb-h}{\}
6426 \DeclareFontShape{JY2}{zhfsf}{bx}{n}{<-> upzhsansb-h}{\}
6427 \DeclareFontShape{JY2}{zhfsf}{m}{sl}{<-> upzhsans-h}{\}
6428 \DeclareFontShape{JY2}{zhfsf}{b}{sl}{<-> upzhsansb-h}{\}
6429 \DeclareFontShape{JY2}{zhfsf}{bx}{sl}{<-> upzhsansb-h}{\}
6430 \DeclareFontShape{JY2}{zhfsf}{m}{it}{<-> upzhsans-h}{\}
6431 \DeclareFontShape{JY2}{zhfsf}{b}{it}{<-> upzhsansb-h}{\}
6432 \DeclareFontShape{JY2}{zhfsf}{bx}{it}{<-> upzhsansb-h}{\}
6433 </jy2>
6434 <*jt2>
6435 \DeclareFontShape{JT2}{zhfsf}{m}{n}{<-> upzhsans-v}{\}
6436 \DeclareFontShape{JT2}{zhfsf}{b}{n}{<-> upzhsansb-v}{\}
6437 \DeclareFontShape{JT2}{zhfsf}{bx}{n}{<-> upzhsansb-v}{\}
6438 \DeclareFontShape{JT2}{zhfsf}{m}{sl}{<-> upzhsans-v}{\}
6439 \DeclareFontShape{JT2}{zhfsf}{b}{sl}{<-> upzhsansb-v}{\}
6440 \DeclareFontShape{JT2}{zhfsf}{bx}{sl}{<-> upzhsansb-v}{\}
6441 \DeclareFontShape{JT2}{zhfsf}{m}{it}{<-> upzhsans-v}{\}
6442 \DeclareFontShape{JT2}{zhfsf}{b}{it}{<-> upzhsansb-v}{\}
6443 \DeclareFontShape{JT2}{zhfsf}{bx}{it}{<-> upzhsansb-v}{\}
6444 </jt2>
6445 </sf>
6446 <*tt>
6447 <*c19>
6448 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
6449 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
6450 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
6451 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
6452 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
6453 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
6454 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
6455 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
6456 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
6457 </c19>
6458 <*c70>
6459 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
6460 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
6461 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
6462 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
6463 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
6464 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
6465 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
6466 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
6467 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
6468 </c70>
6469 <*jy2>
6470 \DeclareFontShape{JY2}{zhhtt}{m}{n}{<-> upzhmono-h}{\}
6471 \DeclareFontShape{JY2}{zhhtt}{b}{n}{<-> upzhmono-h}{\}
6472 \DeclareFontShape{JY2}{zhhtt}{bx}{n}{<-> upzhmono-h}{\}
6473 \DeclareFontShape{JY2}{zhhtt}{m}{sl}{<-> upzhmono-h}{\}
6474 \DeclareFontShape{JY2}{zhhtt}{b}{sl}{<-> upzhmono-h}{\}

```

```

6475 \DeclareFontShape{JY2}{zhzt}{bx}{sl}{<-> upzhmono-h}{f}
6476 \DeclareFontShape{JY2}{zhzt}{m}{it}{<-> upzhmono-h}{f}
6477 \DeclareFontShape{JY2}{zhzt}{b}{it}{<-> upzhmono-h}{f}
6478 \DeclareFontShape{JY2}{zhzt}{bx}{it}{<-> upzhmono-h}{f}
6479 </jy2>
6480 <*j2>
6481 \DeclareFontShape{JT2}{zhzt}{m}{n}{<-> upzhmono-v}{f}
6482 \DeclareFontShape{JT2}{zhzt}{b}{n}{<-> upzhmono-v}{f}
6483 \DeclareFontShape{JT2}{zhzt}{bx}{n}{<-> upzhmono-v}{f}
6484 \DeclareFontShape{JT2}{zhzt}{m}{sl}{<-> upzhmono-v}{f}
6485 \DeclareFontShape{JT2}{zhzt}{b}{sl}{<-> upzhmono-v}{f}
6486 \DeclareFontShape{JT2}{zhzt}{bx}{sl}{<-> upzhmono-v}{f}
6487 \DeclareFontShape{JT2}{zhzt}{m}{it}{<-> upzhmono-v}{f}
6488 \DeclareFontShape{JT2}{zhzt}{b}{it}{<-> upzhmono-v}{f}
6489 \DeclareFontShape{JT2}{zhzt}{bx}{it}{<-> upzhmono-v}{f}
6490 </jt2>
6491 </tt>

```

第 16 节 代码索引

意大利体的数字表示描述对应索引项的页码;带下划线的数字表示定义对应索引项的代码行号;罗马字体的数字表示使用对应索引项的代码行号。

Symbols		10pt	10, <u>329</u>
\#	1556, 1615	11pt	10, <u>329</u>
.../afterindent	26	12pt	10, <u>329</u>
.../aftername	22		
.../aftername+	22		
.../afterskip	25		
.../aftertitle	22		
.../aftertitle+	22		
.../beforeskip	25		
.../break	26		
.../break+	26		
.../fixskip	25		
.../format	20		
.../format+	20		
.../hang	24		
.../indent	25		
.../name	18		
.../nameformat	20		
.../nameformat+	20		
.../number	19		
.../numberformat	21		
.../numberformat+	21		
.../titleformat	22		
.../titleformat+	22		
.../tocline	27		
\/	3817		
\\	17, 18, 91, 147, 149, 201, 210, 650, 1899, 1900, 1934, 1935, 2041, 2248, 2250, 2279, 2721, 2735, 3302, 3384, 4364, 4365, 4366, 4459, 4554		
{	3313, 3314, 4366, 4445, 4460		
}	3313, 3314, 4366, 4445, 4460		
_	1662, 1670, 1683, 1686, 1712, 1720		

`\arabic` 4928
`\AtBeginDocument` 2461, 3794
`\AtBeginDvi` 2325
`\AtBeginShipoutFirst` 2330
`\AtBeginUTFCommand` 2866
`\AtEndOfClass` 115
`\AtEndOfPackage` 116
`\AtEndUTFCommand` 2867
`autoindent` 12, 180, 435

B

`\baselinestretch` 494
`\begin` 1743, 1762, 1781
`\begingroup` 929,
 971, 991, 1012, 1111, 1136, 1201, 1239,
 1272, 1731, 1750, 1769, 2291, 5854, 5888, 5895
`\belowdisplayshortskip`
 2098, 2106, 2117, 2137, 2145, 2156
`\belowdisplayskip`
 2099, 2111, 2122, 2138, 2150, 2161
`\bfdefault` 3169, 3175, 3510, 3517
`\bfseries` 934,
 938, 976, 980, 996, 1017, 1117, 1122, 1141,
 4809, 4811, 4815, 4828, 4829, 4832, 4858,
 4860, 4866, 4883, 4887, 4908, 4942, 4959, 4973
`\bgroup` 3809
`\bibname` 639, 642, 650, 658, 664, 665
`bibname` 14

bool commands:

`\bool_case:nTF` 6226
`\bool_const:Nn` 49, 52
`\bool_gset_false:N` 243, 248
`\bool_gset_true:N` ... 238, 5383, 5524, 5630
`\bool_if:NTF`
 803, 1294, 1486, 1496, 1507, 1850, 1903,
 2395, 2470, 3342, 3895, 4103, 4387, 5373, 5840
`\bool_if_exist:NTF` 4382
`\bool_lazy_and:nnTF` 6239
`\bool_lazy_or:nnTF` 2225, 4582, 5832
`\bool_new:N` 111, 3379, 5366, 5500
`\bool_set_false:N` 3354, 4388, 4397
`\bool_set_true:N` 3372, 4396
`\c_false_bool` 49, 1328, 1330
`\c_true_bool` 52, 1327, 1329

box commands:

`\box_ht:N` 2177
`\box_new:N` 57
`\box_use_drop:N` 1304
`\box_wd:N` 1303, 1611, 4040

C

`c5size` 38
`cap` 38, 288
`captiondelimiter` 39, 2184
`\catcode` . 5854, 5855, 5856, 5857, 5858, 5862, 5863
`CCT` 38

cctab commands:

`\cctab_begin:N` 4389, 4404
`\cctab_const:Nn` 4408
`\cctab_end:` 4393, 4406
`\cctab_select:N` 4410
`\c_document_cctab` 4410
`CCTfont` 38
`\ccwd` 31, 109, 184, 441, 479, 505, 506,
 513, 539, 540, 541, 544, 546, 547, 558, 569,
 571, 4036, 4094, 4095, 4096, 5090, 5091, 5092
`\cdp` 4191
`\centering` 972, 1013, 1732, 1751, 1770,
 4815, 4827, 4832, 4836, 4866, 4887, 4895, 4930
`\chapter` 1042, 1909
`chapter/beforeskip` 39
`chapter/fixbeforeskip` 40
`chapter/lofskip` 27, 775
`chapter/lotskip` 27, 775
`chapter/numbering` 17
`chapter/pagestyle` 23, 775
`\chaptermark` 1086
`\chaptername` 4855

char commands:

`\char_generate:nn` 2618, 2621,
 2622, 2626, 2627, 2628, 2632, 2633, 2634, 2635
`\char_set_catcode:nn` 2570
`\char_set_catcode_active:n` 2563
`\char_set_catcode_comment:n` 2754
`\char_set_catcode_ignore:n` ... 5407, 5598
`\char_set_catcode_letter:n`
 1631, 1870, 3797, 4411, 4546
`\char_set_catcode_other:N` 1556, 1615, 3069
`\char_set_catcode_other:n` 4570
`\char_set_catcode_space:n`
 2872, 4196, 5390, 5550
`\char_value_catcode:n` 2572
`CharRange` 32, 3520
`\Chinese` 39, 585
`\chinese` 31, 583, 593, 4802, 4865
`\cht` 4191
`\CJKaddEncHook` 2418
`\CJKbold` .. 6363, 6364, 6374, 6375, 6403, 6404,
 6406, 6407, 6409, 6410, 6414, 6415, 6417,
 6418, 6420, 6421, 6449, 6450, 6452, 6453,
 6455, 6456, 6460, 6461, 6463, 6464, 6466, 6467
`\CJKfamily` 3241, 3261,
 3615, 3825, 3920, 3921, 3922, 3923, 5822,
 5823, 5824, 5825, 5827, 5828, 5830, 5842, 5843
`\CJKfamilydefault`
 2478, 3474, 3476, 3480, 3487, 3500, 3914,
 3915, 3923, 3932, 3939, 3943, 3951, 3962, 3964
`\CJKfontspec` 3401
`\CJKglue` 4039, 4053, 4056, 4069, 4081
`\CJKhook` 2652
`\CJKnormal` 6356, 6357, 6358, 6359,
 6360, 6361, 6362, 6367, 6368, 6369, 6370,

- 6371, 6372, 6373, 6402, 6405, 6408, 6413,
6416, 6419, 6448, 6451, 6454, 6459, 6462, 6465
- \CJKpunctmapfamily . 2687, 2688, 2689, 2690,
2691, 2692, 2696, 2697, 2698, 2699, 2700, 2701
- \CJKrmdefault . . 2409, 3309, 3320, 3439, 3476,
3479, 3480, 3915, 3917, 3920, 3926, 3929,
3947, 5122, 5125, 5126, 5131, 5132, 5133,
5186, 5189, 5191, 5196, 5197, 5198, 5252,
5255, 5256, 5263, 5264, 5265, 5317, 5320,
5322, 5327, 5328, 5329, 5453, 5456, 5457,
5464, 5466, 5467, 5687, 5691, 5694, 5700,
5702, 5703, 5774, 5775, 5776, 5784, 5785, 5786
- \CJKsfdefault 2410,
3310, 3321, 3446, 3918, 3921, 3927, 3930,
3948, 5123, 5187, 5190, 5253, 5257, 5318,
5321, 5454, 5458, 5465, 5688, 5692, 5777, 5778
- \CJKtilde 2479
- \CJKttdefault 2411, 3311, 3322,
3452, 3919, 3922, 3928, 3931, 3949, 5124,
5188, 5254, 5319, 5455, 5689, 5694, 5701, 5779
- clearalternatefont 33, 3699
- \cleardoublepage 890, 1043, 4825, 4853
- \clearpage 892, 1043, 4825, 4853
- clist commands:
- \clist_clear:N 3190, 6209
- \clist_concat:NNN 3210
- \clist_const:Nn 2666
- \clist_gput_right:Nn 345, 357, 383, 384
- \clist_gset:Nn 3462
- \clist_if_empty:NTF 3545
- \clist_map_break:n 1465, 1894
- \clist_map_function:nN
 3528, 3702, 3704, 3737, 3751
- \clist_map_inline:Nn 2656, 3645
- \clist_map_inline:nn 330, 1460, 1891,
 1957, 1980, 2011, 2059, 2679, 3744, 4031, 6196
- \clist_new:N
 126, 3207, 3463, 3541, 3588, 4359, 6254
- \clist_put_left:Nn 3211, 5049, 5074
- \clist_put_right:Nn 3371, 3634, 6248
- \clist_set:Nn
 3194, 3531, 3571, 3572, 3791, 4360
- \clist_use:Nn 4369
- \clubpenalty 1271, 1278
- codepoint commands:
- \codepoint_str_generate:n 2417
- \contentsname 629
- contentsname 13
- continuation 15
- cs commands:
- \cs_generate_variant:Nn 80, 1306, 1807,
 2349, 3142, 3273, 3360, 3380, 3381, 3589,
 3605, 3677, 3698, 3737, 3745, 4508, 4637, 4744
- \cs_gset:Npn 2541, 2548, 2555
- \cs_gset_eq:NN 239, 244,
 249, 1417, 1418, 1424, 1449, 2332, 2391,
 2392, 2534, 2535, 2536, 2537, 2566, 3347, 3350
- \cs_gset_protected:Npn 1381, 1430, 1445, 2384
- \cs_gset_protected:Npx 1357, 3238, 3609, 3695
- \cs_gset_protected_nopar:Npx 3012
- \cs_if_eq:NNTF
 1415, 1448, 3103, 4519, 4525, 4670, 4750
- \cs_if_exist:NTF
 3, 24, 321, 363, 1531, 1827, 1877,
 1909, 1911, 2317, 2360, 2962, 2978, 3010,
 3044, 3122, 3413, 3608, 3693, 4055, 4155, 4422
- \cs_if_exist_p:N 4583, 4584
- \cs_if_exist_use:N 3257, 3258
- \cs_if_exist_use:NTF
 2543, 2550, 2557, 2822, 2944, 3279, 3713, 4515
- \cs_if_free:NTF 454, 468, 589, 1479,
 1524, 1581, 1586, 1590, 2343, 2353, 2431,
 2452, 2658, 2674, 3344, 3509, 4105, 4114, 4117
- \cs_new:Npn 101, 108, 583,
 757, 1331, 1339, 1347, 1366, 1504, 1579,
 1584, 2483, 2485, 2491, 2499, 2505, 2507,
 2516, 2522, 2524, 2564, 2601, 2603, 2605,
 2617, 2619, 2624, 2630, 2637, 2638, 2639,
 3029, 3059, 3064, 3066, 3070, 3186, 3252,
 3260, 3305, 3316, 3766, 4374, 4627, 4745, 6255
- \cs_new:Npx 2196, 2202
- \cs_new_eq:NN 95, 96, 97, 113,
 114, 322, 584, 585, 586, 821, 840, 847, 1315,
 1325, 1326, 1327, 1328, 1329, 1330, 1380,
 1404, 1405, 1880, 1907, 1912, 1949, 2350,
 2378, 2482, 2705, 2987, 3014, 3080, 3110,
 3435, 3442, 3958, 4081, 4102, 4111, 4573,
 4574, 4674, 4746, 5836, 5837, 5846, 5847, 6201
- \cs_new_protected:Npn 60, 63, 65,
 67, 69, 99, 115, 116, 118, 134, 136, 141, 402,
 406, 453, 455, 480, 488, 500, 502, 521, 531,
 536, 691, 721, 768, 804, 811, 816, 824, 831,
 838, 842, 848, 850, 853, 869, 1283, 1289,
 1291, 1299, 1307, 1316, 1397, 1407, 1441,
 1452, 1472, 1477, 1482, 1492, 1502, 1552,
 1557, 1605, 1623, 1801, 1823, 1832, 1838,
 1848, 1865, 1925, 1940, 2044, 2049, 2079,
 2085, 2192, 2208, 2218, 2220, 2313, 2315,
 2319, 2324, 2329, 2341, 2351, 2356, 2358,
 2363, 2407, 2416, 2423, 2433, 2574, 2580,
 2586, 2592, 2598, 2599, 2600, 2644, 2650,
 2654, 2672, 2677, 2685, 2694, 2703, 2794,
 2796, 2802, 2804, 2812, 2814, 2819, 2832,
 2837, 2852, 2941, 2951, 2965, 2970, 2976,
 3031, 3042, 3054, 3073, 3082, 3098, 3112,
 3137, 3187, 3208, 3217, 3236, 3253, 3263,
 3288, 3326, 3338, 3340, 3362, 3364, 3387,
 3470, 3495, 3504, 3527, 3529, 3536, 3543,
 3549, 3556, 3558, 3563, 3569, 3590, 3599,
 3606, 3612, 3623, 3631, 3637, 3643, 3661,

3670, 3678, 3690, 3708, 3725, 3743, 3746,
 3748, 3761, 3780, 3788, 3802, 3818, 3820,
 3844, 3858, 3871, 3884, 3900, 3905, 3937,
 4014, 4022, 4026, 4036, 4050, 4093, 4097,
 4112, 4146, 4150, 4170, 4376, 4384, 4391,
 4403, 4405, 4413, 4420, 4433, 4441, 4449,
 4455, 4462, 4472, 4474, 4485, 4487, 4489,
 4491, 4501, 4503, 4509, 4514, 4517, 4523,
 4530, 4535, 4540, 4551, 4555, 4560, 4565,
 4575, 4580, 4599, 4650, 4665, 4675, 4687,
 4696, 4715, 4720, 4732, 4747, 6193, 6202, 6246
 \cs_new_protected:Npx
 1533, 2371, 2825, 4470, 4476
 \cs_new_protected_nopar:Npn
 2988, 3002, 3004
 \cs_replacement_spec:N 4591, 4613
 \cs_set:Npn 1596, 4108
 \cs_set_eq:NN 98, 185,
 190, 195, 600, 605, 608, 613, 849, 851, 1318,
 1319, 1320, 1321, 1322, 1323, 1412, 1456,
 1466, 2417, 2430, 2450, 2647, 2651, 2660,
 2675, 2806, 2839, 2847, 2854, 2856, 2857,
 2956, 2981, 2982, 3081, 3111, 3129, 3184,
 3241, 3608, 3615, 3625, 3626, 3627, 3693,
 3694, 3817, 3942, 4056, 4126, 4129, 4130, 4133
 \cs_set_nopar:Npn 3843
 \cs_set_nopar:Npx 1494, 4493
 \cs_set_protected:Npn 1526, 1594,
 2086, 2646, 3115, 3130, 4053, 4085, 4578, 4642
 \cs_set_protected:Npx
 2817, 3417, 4099, 4567, 4568
 \cs_to_str:N 105, 3412, 3903, 3908, 4576
 \cs_undefine:N
 . 90, 3221, 3222, 3226, 3227, 3628, 4134, 4515
 cs4size 38
 \csname 1169, 1178,
 1179, 1180, 1187, 1207, 1208, 1210, 1213,
 1222, 1223, 1224, 1227, 1872, 1873, 2292,
 2295, 2296, 5887, 5889, 5890, 5891, 5896, 5900
 \CTeX 31, 2180
 ctex commands:
 \ctex_add_cmap:n ... 2346, 2351, 2351, 2391
 \ctex_add_dvi_zhmap: 2324, 2326, 2332
 \ctex_add_to_selectfont:n
 467, 4097, 4097, 4161, 4189, 4213
 \ctex_after_end_preamble:n
 45, 98, 2326, 4474, 4485, 4487
 \ctex_appto_cmd:NnnTF 2844, 3796, 4560, 4560
 \ctex_assign_heading_name:nn
 727, 765, 765, 1373
 \ctex_at_begin_package:nn
 133, 2727, 4449, 4449, 4460
 \ctex_at_end:n 115, 115, 116, 261,
 297, 303, 314, 319, 2286, 2568, 2754, 4081, 4083
 \ctex_at_end_package:nn
 134, 587, 1428, 1443, 1512, 1519,
 1544, 1589, 1591, 1592, 1628, 1674, 1836,
 1886, 2182, 2327, 2389, 2729, 2863, 2869,
 3120, 3974, 3977, 4003, 4011, 4018, 4462, 4462
 \ctex_at_end_preamble:n
 324, 2469, 2475, 3936, 4472, 4485, 4485
 \ctex_at_shipout_first:n
 2319, 2321, 2329, 2331, 3981, 4005
 \ctex_auto_ignorespaces:
 2482, 2482, 2651, 4223
 \ctex_CJK_input:n 2423, 2423, 2430
 \ctex_cleveref_hook: 1836, 1848, 1848
 \ctex_declare_math_sizes:nnnn
 2044, 2044, 2078
 \ctex_define:n 59, 63, 435, 469, 601,
 627, 655, 665, 668, 671, 797, 1369, 1792,
 1843, 2184, 2253, 3699, 3738, 4215, 4236, 4248
 \ctex_define_option:n . 59, 60, 156, 351, 354
 \ctex_deprecated_command:Nn
 134, 141, 418, 420, 428, 430, 568, 575
 \ctex_deprecated_option:n 134,
 138, 199, 208, 296, 302, 306, 313, 318, 2188
 \ctex_deprecated_option:nn 134, 263
 \ctex_detect_platform: 2208, 2208, 2223, 2238
 \ctex_disable_package:n
 40, 41, 43, 44, 2731, 4413, 4413
 \ctex_family_cmap:nn 2341, 2341, 2349, 2350
 \ctex_file_input:n 97, 360,
 409, 1918, 2005, 2245, 2286, 2412, 2706,
 2855, 4376, 4376, 5042, 5047, 5052, 5053,
 5067, 5072, 5078, 5079, 5410, 5415, 5419, 5420
 \ctex_fix_varioref_label:n 1835, 1838, 1838
 \ctex_fontset_case:nnn
 2196, 2196, 5237, 5302, 5756
 \ctex_fontset_case:nnnn
 151, 2202, 2202, 5098, 5162, 5425, 5669
 \ctex_fontset_error:n
 2192, 2192, 5099, 5135, 5163, 5200, 5267,
 5331, 5427, 5431, 5432, 5469, 5670, 5705, 5788
 \ctex_gadd_hook:Nn .. 4501, 4502, 4503, 4508
 \ctex_gadd_ltxhook:nn 2317,
 2320, 3908, 4151, 4470, 4473, 4475, 4479, 4480
 \ctex_gadd_package_hook:nnn
 4453, 4466, 4476, 4501, 4501
 \ctex_gadd_selectfont_hook:n
 4146, 4150, 4154, 4158
 \ctex_get_macro_meaning:N 4645
 \ctex_get_macro_meaning:NTF
 4641, 4667, 4723
 \ctex_hang_from:n 1291, 1295, 1299
 \ctex_heading_depth:nn
 1795, 1798, 1801, 1801, 1807
 \ctex_heading_glue:n 1307, 1307, 1315
 \ctex_heading_hang:Nnn
 1285, 1290, 1291, 1291, 1306
 \ctex_hypersetup:n 51, 322, 399,
 402, 406, 3972, 3976, 3988, 3991, 3995, 3997

<code>\ctex_if_autoindent_touched:TF</code>	<code>\ctex_ltj_if_jfont:nTF</code>
..... 113 , 113 , 185 , 190 , 195 , 5034	 107 , 3059 , 3059 , 3071 , 3075
<code>\ctex_if_ccglue_touched:</code>	<code>\ctex_ltj_if_jfont_math:NTF</code>
<code>\ctex_if_ccglue_touched:TF</code> 490 , 4066 , 4087	 3066 , 3066 , 3105
<code>\ctex_if_ccglue_touched_p:</code>	<code>\ctex_ltj_is_kenc:n</code>
4066	2897 , 3061
<code>\ctex_if_format_at_least:nTF</code>	<code>\ctex_ltj_math_group_hook:</code>
..... 48 , 4374 , 4374 , 4431 , 4468 , 4478	 106 , 3104 , 3110 , 3130
<code>\ctex_if_platform_macos:TF</code> 2213 , 2218 , 2218	<code>\ctex_ltj_patch_external_font:n</code> 2898 , 3030
<code>\ctex_if_preamble:TF</code>	<code>\ctex_ltj_pickup_alt_font:nn</code> ..
97 , 97 , 98 , 2257	2924 , 3035
<code>\ctex_if_rescanable:NnTF</code> ..	<code>\ctex_ltj_pickup_font:</code>
4665 , 4665 , 4699	2956 ,
<code>\ctex_ignorespaces_case:N</code>	2958 , 2976 , 2976 , 2987 , 3049 , 3085 , 3087 , 3089
..... 2644 , 2644 , 4220 , 4223 , 4226	<code>\ctex_ltj_reset_alternate_font:n</code> ..
<code>\ctex_indent_box:n</code>	 3704 , 3725
811 , 811 , 821	<code>\ctex_ltj_save_alternate_family:Nnnn</code>
<code>\ctex_load_fontset:</code>	 3582 , 3603 , 3606 , 3606
..... 2220 , 2220 , 2252 , 2265 , 2285	<code>\ctex_ltj_save_alternate_seq:Nn</code>
<code>\ctex_load_std_class:n</code>	 3553 , 3558 , 3558 , 3589
..... 117 , 118 , 387 , 390 , 393 , 396	<code>\ctex_ltj_save_char_range:n</code>
<code>\ctex_load_zhmap:nnnn</code>	 3751 , 3758 , 3758
2407 , 2407 ,	<code>\ctex_ltj_save_char_range:nn</code> ..
2414 , 5129 , 5194 , 5261 , 5325 , 5462 , 5698 , 5782	3760 , 3761
<code>\ctex_ltj_add_font_features:n</code>	<code>\ctex_ltj_select_alternate_font:</code> ...
..... 3362 , 3362 , 3380 , 3432	 3031 , 3031 , 4164
<code>\ctex_ltj_add_font_features:nn</code>	<code>\ctex_ltj_select_font:</code> ...
..... 3362 , 3363 , 3364 , 3381	2941 , 2941 , 4163
<code>\ctex_ltj_add_kyenc:n</code>	<code>\ctex_ltj_set_alt_font:nnnn</code>
2896 , 3165	 2913 , 3672 , 3675
<code>\ctex_ltj_char_range_key:nn</code>	<code>\ctex_ltj_set_alternate_family:nn</code> ..
..... 3785 , 3788 , 3788	 3557 , 3590 , 3590
<code>\ctex_ltj_clear_alt_font:n</code>	<code>\ctex_ltj_set_alternate_family:nnn</code> ..
2920 , 3656	 3602 , 3610 , 3612 , 3612
<code>\ctex_ltj_clear_alternate_font:n</code> ...	<code>\ctex_ltj_set_alternate_family:Nnnnn</code>
..... 3702 , 3708	 3594 , 3599 , 3605
<code>\ctex_ltj_declare_alternate_shape:nnnnnn</code>	<code>\ctex_ltj_set_alternate_family:nnTF</code> ..
..... 3617 , 3637 , 3637 , 3639	 3195 , 3543 , 3543
<code>\ctex_ltj_declare_char_range:n</code>	<code>\ctex_ltj_set_alternate_prop:n</code>
..... 3740 , 3743 , 3745	 3522 , 3527 , 3527
<code>\ctex_ltj_declare_char_range:nn</code>	<code>\ctex_ltj_set_alternate_shape:n</code>
..... 3747 , 3748 , 3748	 3663 , 3670 , 3677 , 3682
<code>\ctex_ltj_def_char_range_key:n</code>	<code>\ctex_ltj_set_alternate_shape:nnN</code> ..
..... 3753 , 3780 , 3780	 3649 , 3678 , 3678
<code>\ctex_ltj_ensure_default_family:</code> ...	<code>\ctex_ltj_set_alternate_shape:nnn</code> ..
..... 3470 , 3470 , 3955	 3652 , 3658 , 3658
<code>\ctex_ltj_extract_font:</code> ..	<code>\ctex_ltj_set_alternate_shape:nnnn</code> ..
2981 , 2988 , 2988	 3660 , 3661
<code>\ctex_ltj_family_if_exist:nN</code> ..	<code>\ctex_ltj_set_alternate_shape:Nnnnnnnn</code>
3274 , 3287	 3640 , 3643 , 3643
<code>\ctex_ltj_family_if_exist:nTF</code>	<code>\ctex_ltj_set_family:nnn</code>
... 3265 , 3274 , 3474 , 3479 , 3497 , 3500 , 3711	3187 ,
<code>\l_ctex_ltj_family_tl</code>	3187 , 3357 , 3404 , 3421 , 3439 , 3446 , 3452 , 3458
3267 , 3272 ,	<code>\ctex_ltj_set_kanjiskip:N</code> 3818 , 3818 , 4060
3329 , 3330 , 3345 , 3349 , 3352 , 3363 , 3705 , 3706	<code>\ctex_ltj_set_math_letter:NN</code>
<code>\ctex_ltj_fontspec:nn</code>	 3112 , 3124 , 3132
..... 3326 , 3326 , 3338 , 3339 , 3373 , 3427	<code>\ctex_ltj_set_xkanjiskip:N</code>
<code>\ctex_ltj_get_and_define_fonts:nN</code> ..	 3818 , 3820 , 4177 , 4261
..... 3073 , 3073 , 3081	<code>\ctex_ltj_subst_font:</code>
<code>\ctex_ltj_get_and_define_fonts_al:nN</code>	2982 , 3004 , 3004
..... 3077 , 3080	<code>\ctex_ltj_swap_cs:NN</code>
<code>\ctex_ltj_get_and_define_fonts_ja:nN</code>	3616 , 3623 , 3623
..... 3076 , 3082	<code>\ctex_ltj_switch_family:n</code>
<code>\ctex_ltj_if_alternate_shape_exist:n</code> 3024	 3262 , 3263 , 3273 , 3330 , 3358 , 3418
<code>\ctex_ltj_if_alternate_shape_-</code>	
<code>exist:nTF</code>	
2991 , 3006 , 3024 , 3033	

`\ctex_ljt_update_mathfont`: 3492, 3495, 3495
`\ctex_ljt_update_mathfont:n` 3498, 3501, 3504
`\ctex_ljt_use_jfont`: 2998, 3002, 3002
`\ctex_ljt_use_math_group:Nn` 3098, 3098, 3111, 3129
`\ctex_ljt_zero_globaldefs`: 2799, 2936
`\ctex_make_spa:nn` 6193, 6193, 6201
`\ctex_parse_name:NN` . 4137, 4571, 4575, 4575
`\ctex_parse_name:NNn` 4576, 4580, 4637
`\ctex_patch_boot:NNnnTF`
... 4532, 4537, 4542, 4557, 4562, 4565, 4565
`\ctex_patch_cmd:Nnn` .. 1546, 1559, 1565,
1571, 1644, 1647, 1652, 1658, 1669, 1676,
1679, 1682, 1685, 1694, 1697, 1702, 1708,
1719, 2454, 3839, 4540, 4540, 5090, 5091, 5092
`\ctex_patch_cmd_all:NnnnTF` 1867, 4535, 4535
`\ctex_patch_cmd_once:NnnnTF`
..... 3833, 4139, 4530, 4530
`\ctex_patch_failure:N` 1622,
1634, 1875, 2848, 2850, 3800, 4144, 4549, 4551
`\ctex_plane_to_utfxvibe:Nn` 2431, 2433, 2450
`\ctex_pop_file`:
..... 78, 2428, 4376, 4380, 4391, 4405
`\ctex_preto_cmd:NnnTF`
..... 1619, 1630, 2840, 4555, 4555
`\ctex_provide_font_hook:NNN`
..... 3900, 3900, 3910, 3911, 3912, 3913
`\ctex_punct_map_bfseries:nn` 2677, 2677,
5125, 5132, 5189, 5190, 5197, 5256, 5257,
5258, 5264, 5320, 5321, 5328, 5457, 5458,
5459, 5466, 5692, 5693, 5702, 5775, 5778, 5785
`\ctex_punct_map_family:nn`
..... 2672, 2672, 5122, 5123,
5124, 5131, 5186, 5187, 5188, 5196, 5252,
5253, 5254, 5263, 5317, 5318, 5319, 5327,
5453, 5454, 5455, 5464, 5465, 5687, 5688,
5689, 5690, 5700, 5701, 5774, 5777, 5779, 5784
`\ctex_punct_map_itshape:nn` 2694,
2694, 5126, 5133, 5191, 5198, 5255, 5265,
5322, 5329, 5456, 5467, 5691, 5703, 5776, 5786
`\ctex_punct_map_series:nnn` 2681, 2682, 2685
`\ctex_punct_set:n` 2654,
2654, 5121, 5130, 5185, 5195, 5251, 5262,
5316, 5326, 5452, 5463, 5686, 5699, 5760, 5783
`\ctex_punct_space:nn` 2703, 2703, 2705
`\ctex_push_file`:
..... 71, 2425, 4376, 4378, 4384, 4403
`\ctex_replace_package:nn`
..... 4431, 4433, 4441, 4445
`\ctex_save_font_size:nn` 9, 1938, 1949
`\ctex_scheme_input:n` . 69, 69, 80, 1888, 1920
`\ctex_select_size`: 442, 449, 452, 453, 474, 534
`\ctex_set:n` .. 59, 65, 261, 412, 432, 5033, 5035
`\ctex_set:nn` 33, 59, 67, 139, 422,
4783, 4786, 4844, 4872, 4904, 4939, 4956,
4970, 4986, 4994, 5002, 5010, 5017, 5018, 5019
`\ctex_set_default_ccwd:Nn`
..... 99, 99, 194, 448, 473
`\ctex_set_deprecated_option:n` ... 134,
136, 173, 174, 228, 229, 230, 252, 257, 288, 289
`\ctex_set_font_size:Nnn`
2079, 2079, 2094, 2102, 2113, 2124, 2125,
2126, 2127, 2128, 2129, 2130, 2133, 2141,
2152, 2163, 2164, 2165, 2166, 2167, 2168, 2169
`\ctex_set_ignorespaces`:
..... 2644, 2646, 2648, 2650, 2653
`\ctex_set_jfm:n`
..... 3137, 3137, 3142, 3153, 3157, 4243
`\ctex_set_upfamily:nnn` 3844, 3844,
3891, 5145, 5146, 5147, 5148, 5210, 5211,
5212, 5213, 5277, 5278, 5279, 5280, 5281,
5282, 5341, 5342, 5343, 5344, 5479, 5480,
5481, 5482, 5483, 5484, 5715, 5716, 5717,
5718, 5798, 5799, 5800, 5801, 5802, 5803, 5804
`\ctex_set_upfonts:nnnnn`
3884, 3884, 3894, 5138, 5203, 5270, 5334, 5791
`\ctex_set_upmap:nnn` 3858,
3858, 3886, 3887, 3888, 3889, 3892, 5283,
5284, 5473, 5474, 5475, 5476, 5714, 5805, 5806
`\ctex_set_upmap_unicode:nnn`
3871, 3871, 3893, 5477, 5478, 5708, 5710, 5712
`\ctex_set_zhmap:n`
..... 2312, 2313, 2336, 2412, 3860, 3873
`\ctex_titleps_hook`: . 1555, 1557, 1590, 1591
`\ctex_titlesec_hook`: 1452, 1452, 1516
`\ctex_undeclare_unicode_character:n` .
..... 2416, 2416, 2420, 2421
`\ctex_update_ccglue`: 497, 519, 562, 4050, 4050
`\ctex_update_ccwd`: 491, 570, 4036, 4036, 4088
`\ctex_update_default_family`:
..... 92, 2469, 3936, 3937, 3937
`\ctex_update_em_unit`: . 482, 538, 4093, 4093
`\ctex_update_kanjisize`: 4191, 4213
`\ctex_update_parindent`: . 460, 521, 521, 563
`\ctex_update_size`: 454, 455, 455, 468
`\ctex_update_stretch`: 459, 480, 480
`\ctex_update_xkanjiskip`: . 4169, 4170, 4189
`\ctex_update_ziju`: 462, 536, 536
`\ctex_use_package_hook:nn`
..... 4509, 4509, 4520, 4526
`\ctex_use_zhmap`: ... 2315, 2321, 2325, 2331
`\ctex_varioref_hook`: 1832, 1832, 1878, 1886
`\ctex_zhmap_case:nnn`
..... 151, 114, 114, 239, 244,
249, 5101, 5165, 5239, 5304, 5436, 5672, 5758
`\ctex_zihao:n` 1924, 1925, 1925
`\ctex_ziju:n` 530, 531
ctex internal commands:
`\_ctex_add_cmap_auxi:n` 2354, 2356

<code>_ctex_add_cmap_auxii:Nn</code>	2357, 2358	<code>\c_ctex_everysel_loaded_bool</code>	
<code>\g_ctex_after_end_preamble_hook_tl</code>	4488, 4492, 4500		46, 3895, 4103
<code>_ctex_assign_heading_name:nnn</code>		<code>\g_ctex_expl_status_seq</code> ..	4386, 4394, 4400
	765, 767, 768	<code>\l_ctex_expl_status_tl</code> ..	4394, 4395, 4399
<code>\l_ctex_autoindent_tl</code>	112,	<code>\l_ctex_family_default_init_tl</code>	3939, 3957
	184, 189, 194, 441, 445, 448, 523, 526, 807	<code>_ctex_family_default_wrap:n</code>	
<code>_ctex_calc_bounds:nn</code> ...	6250, 6251, 6255		3942, 3958, 3961
<code>_ctex_ccglue:</code>	4055, 4056, 4069, 4081	<code>\g_ctex_font_size_int</code> ..	151, 167, 168,
<code>\l_ctex_ccglue_skip</code>	495, 509, 518,		169, 340, 344, 381, 2087, 5054, 5055, 5080, 5081
	553, 561, 4054, 4060, 4063, 4065, 4073, 4077	<code>\c_ctex_font_size_prop</code>	
<code>_ctex_char:N</code>	2483, 2534		1927, 1938, 2054, 2081
<code>_ctex_char:NN</code>	2485, 2535	<code>\c_ctex_font_size_seq</code>	1936, 2035
<code>_ctex_char:NNN</code>	2499, 2536	<code>\g_ctex_font_size_system_tl</code>	
<code>_ctex_char:NNNN</code>	2516, 2537		160, 1953, 2003, 2006, 2010
<code>_ctex_char_aux:NN</code>		<code>\l_ctex_font_size_temp_prop</code>	
	2488, 2489, 2491, 2541, 2545		1938, 1942, 1951, 2034
<code>_ctex_char_auxi:NNNN</code>	2502, 2505	<code>\l_ctex_font_size_temp_seq</code>	
<code>_ctex_char_auxi:NNNNN</code>	2519, 2522		1939, 1947, 1952, 2036
<code>_ctex_char_auxii:NNN</code>		<code>\l_ctex_font_size_tl</code>	
	2503, 2506, 2507, 2548, 2552		1927, 1928, 2046, 2047, 2081, 2082
<code>_ctex_char_auxii:NNNN</code>		<code>\g_ctex_fontset_tl</code>	227,
	2520, 2523, 2524, 2555, 2559		2211, 2214, 2215, 2222, 2226, 2227, 2230,
<code>_ctex_char_preproc:w</code>	2564, 2566		2231, 2234, 2240, 2245, 2262, 2264, 2269, 2284
<code>_ctex_char_raw:n</code>	2617, 2640	<code>_ctex_get_font_sizes:Nn</code>	2046, 2049, 2049
<code>_ctex_char_raw:nn</code>	2619, 2637	<code>_ctex_get_macro_meaning:w</code>	
<code>_ctex_char_raw:nnn</code>	2624, 2638		4641, 4647, 4650
<code>_ctex_char_raw:nnnn</code>	2630, 2639	<code>\l_ctex_heading_bool</code>	265, 1903
<code>_ctex_char_raw_nn:n</code>	2637, 2641	<code>\c_ctex_heading_level_prop</code> ...	1803, 1808
<code>_ctex_char_raw_nnn:n</code>	2638, 2642	<code>\l_ctex_heading_skip</code>	
<code>_ctex_char_raw_nnnn:n</code>	2639, 2643		828, 835, 837, 839, 840,
<code>_ctex_char_wrap:nn</code>	2601, 2614		855, 856, 859, 861, 862, 865, 1310, 1311, 1312
<code>_ctex_char_wrap:nnn</code>	2603, 2615	<code>\c_ctex_headings_cs_seq</code>	
<code>_ctex_char_wrap:nnnn</code>	2605, 2616		1399, 1413, 1423, 1446
<code>\c_ctex_class_tl</code>		<code>\c_ctex_headings_seq</code>	
	72, 74, 120, 1907, 1913, 1915, 1916, 1918		682, 792, 1529, 1603, 1834
<code>_ctex_clear_fntef_color:n</code>		<code>_ctex_hookto_cmd:Nnnnw</code>	
	4014, 4022, 4026, 4033		4557, 4562, 4720, 4720
<code>_ctex_cleveref_hook_aux:N</code>		<code>_ctex_hookto_cmd_parameter:Nnnnw</code> ..	
	1848, 1856, 1859, 1860, 1862, 1865		4727, 4747, 4747
<code>\c_ctex_cmap_encoding_seq</code>	2339, 2345	<code>_ctex_hookto_cmd_parameterless:Nnnnw</code>	
<code>\l_ctex_cmap_file_tl</code>	2366, 2370, 2380		4726, 4732, 4732
<code>\l_ctex_ctexcap_options_clist</code>		<code>_ctex_if_font_exists:n</code>	6222
	4359, 4360, 4369	<code>_ctex_if_font_exists:nTF</code>	6205, 6222
<code>_ctex_def_heading_keys:n</code> ..	721, 721, 795	<code>\l_ctex_if_otf_exists_tl</code>	
<code>_ctex_default_ccwd_aux:n</code>	100, 101		6224, 6229, 6235, 6240, 6244
<code>_ctex_default_ccwd_aux:w</code>	104, 108	<code>\l_ctex_if_ttf_exists_tl</code>	
<code>_ctex_disable_package_aux:nnnn</code> ...			6225, 6231, 6237, 6241, 6244
	2745, 4417, 4420	<code>_ctex_initial_heading:n</code> ..	691, 691, 794
<code>\g_ctex_encoding_tl</code>		<code>_ctex_insert_indent:</code>	814, 816, 1297
	110, 218, 221, 224, 373, 377, 580, 2398,	<code>_ctex_left_hook_aux:nn</code>	4745
	2403, 2472, 2478, 2538, 3970, 5037, 5063, 5077	<code>\l_ctex_line_spread_fp</code>	
<code>\g_ctex_end_preamble_hook_tl</code>			84, 177, 2173, 2175, 5056, 5057, 5082, 5083
	4486, 4490, 4499	<code>\l_ctex_line_stretch_tl</code>	473, 478, 479, 483
<code>\c_ctex_engine_file_str</code>	23, 409	<code>\l_ctex_ltj_add_alternate_bool</code>	
<code>\c_ctex_engine_str</code>	23		3342, 3354, 3372, 3379

`\__ctex_ltj_altername_cs:n` 3222, 3225, 3226,
 3227, 3257, 3258, 3260, 3345, 3348, 3349,
 3351, 3352, 3553, 3595, 3655, 3674, 3714, 3731
`\l__ctex_ltj_altername_family_tl` ...
 3619, 3622
`\l__ctex_ltj_altername_options_clist`
 3572, 3573, 3584, 3588
`\l__ctex_ltj_altername_prop`
 3191, 3538, 3539, 3542, 3552, 3560
`\l__ctex_ltj_base_CJKfamily_tl`
 3192, 3206, 3255, 3655, 3674
`\l__ctex_ltj_base_family_tl`
 116, 3223, 3250,
 3256, 3601, 3641, 3694, 3711, 3717, 3718, 3730
`\l__ctex_ltj_char_range_clist` .. 3190,
 3524, 3545, 3571, 3583, 3596, 3618, 3640, 3791
`\g__ctex_ltj_char_range_prop` .. 3647, 3748
`\l__ctex_ltj_char_range_tl`
 ... 3647, 3650, 3750, 3752, 3754, 3756, 3763
`\__ctex_ltj_chardef_text_cmd:` . 2808, 2812
`\__ctex_ltj_chardef_text_cmd_aux:N` ..
 2813, 2814
`\__ctex_ltj_check_family:n` 3217, 3217, 3551
`\l__ctex_ltj_cmd_tl` 2807, 2811, 2813
`\l__ctex_ltj_current_font_tl`
 102, 2944, 2948, 2949, 2962, 3010, 3015
`\l__ctex_ltj_current_shape_tl`
 3033, 3036, 3039, 3040
`\__ctex_ltj_declare_char_range:nn` ..
 3744, 3746
`\g__ctex_ltj_default_features_clist` ..
 3210, 3462, 3463
`\__ctex_ltj_family_csname:n`
 3221, 3238, 3252, 3279
`\g__ctex_ltj_family_font_name_prop` ..
 ... 3197, 3214, 3219, 3290, 3366, 3472, 3484
`\g__ctex_ltj_family_font_options-`
`prop` 3198, 3214, 3369
`\g__ctex_ltj_family_int`
 3332, 3334, 3633, 3634, 3636
`\g__ctex_ltj_family_name_prop`
 3214, 3223, 3244, 3276, 3601
`\__ctex_ltj_family_unknown_warning:n`
 3270, 3288, 3288, 3722
`\l__ctex_ltj_font_options_clist`
 3194, 3199, 3200, 3201,
 3207, 3243, 3370, 3371, 3374, 3592, 3593, 3597
`\g__ctex_ltj_fontname_seq` 2967, 2972, 2975
`\__ctex_ltj_fontspec:nnn` . 3333, 3340, 3360
`\g__ctex_ltj_fontspec_family_tl`
 3242, 3245, 3251, 3256, 3281
`\g__ctex_ltj_fontspec_prop` 3328, 3356, 3361
`\__ctex_ltj_gset_family_cs:nn`
 3202, 3236, 3236
`\__ctex_ltj_if_alt_set:nTF` 2935, 3026
`\__ctex_ltj_if_jfont_math:w` ... 3067, 3070
`\l__ctex_ltj_jfm_tl` . 3137, 3168, 3170, 3212
`\__ctex_ltj_ltj_if_jfont:w` 3061, 3064
`\c__ctex_ltj_math_fam_int` 3176, 3177
`\c__ctex_ltj_math_family_tl`
 3506, 3508, 3510, 3513, 3517
`\c__ctex_ltj_math_tl` 3171,
 3172, 3174, 3176, 3458, 3497, 3507, 3512, 3516
`\__ctex_ltj_msg_def_family_map:n` ...
 3303, 3305
`\__ctex_ltj_msg_family_map:n`
 3235, 3302, 3316
`\__ctex_ltj_noembed_wrap:n` 3184, 3186
`\__ctex_ltj_pass_args:nnnn` 3387,
 3387, 3403, 3420, 3426, 3438, 3445, 3451, 3457
`\__ctex_ltj_patch_external_font:w` ..
 2994, 3029, 3029
`\__ctex_ltj_patch_text_cmd:NN` . 2834, 2837
`\__ctex_ltj_pickup_alt_font:Nn` 2930, 3057
`\__ctex_ltj_pop_fontname:` 2961, 2970, 3051
`\__ctex_ltj_post_arg:w` 3390, 3396
`\__ctex_ltj_push_altername_prop:n` ..
 3528, 3529
`\__ctex_ltj_push_altername_prop:nn` ..
 3533, 3536, 3792
`\__ctex_ltj_push_fontname:n`
 2957, 2965, 3048
`\c__ctex_ltj_range_max_int` 3775, 3779
`\c__ctex_ltj_range_min_int` 3774, 3778
`\__ctex_ltj_range_normalization:nn` ..
 3665, 3764, 3766
`\l__ctex_ltj_redirect_jfm_prop` 3137
`\g__ctex_ltj_reset_altername_prop` ..
 3228, 3716, 3728, 3736
`\__ctex_ltj_save_altername_auxi:w` ..
 3561, 3563
`\__ctex_ltj_save_altername_auxii:w` ..
 3566, 3567, 3569
`\__ctex_ltj_save_altername_shape:Nn` .
 3654, 3673, 3690, 3690, 3698
`\__ctex_ltj_select_font_aux:`
 2945, 2951, 2951, 2963
`\__ctex_ltj_set_altername_family:n` ..
 3246, 3253
`\__ctex_ltj_set_altername_family-`
`aux:nnn` 3547, 3556
`\__ctex_ltj_set_family_aux:nnn` 3546, 3549
`\__ctex_ltj_tmp:w` 3625, 3627, 3628
`\l__ctex_ltj_tmp_clist` 3531, 3532, 3534, 3541
`\l__ctex_ltj_tmp_tl`
 2972, 2973, 3193, 3194, 3219, 3230, 3233,
 3367, 3375, 3411, 3418, 3421, 3474, 3479,
 3497, 3498, 3500, 3501, 3575, 3577, 3578, 3585
`\g__ctex_ltj_unknown_family_seq`
 3292, 3294, 3299

_ctex_ltj_update_family_uid:N	_ctex_rebuild_cmd:w
..... 3200, 3592, 3630, 3631	 4669, 4670, 4674, 4749, 4750
_ctex_ltj_use_global_options:N ...	\l_ctex_replacement_str
..... 3201, 3208, 3208, 3573, 3593	 4638, 4658, 4679, 4717, 4718, 4752
\g_ctex_mac_baoli_dir_tl 5650, 5653	\l_ctex_replacement_tl .. 4679, 4684, 4695
\g_ctex_mac_kaiti_dir_tl	_ctex_restore_selectfont:
..... 5599, 5609, 5611, 5640, 5644	 4112, 4134, 4143
\g_ctex_mac_pingfang_bool	_ctex_right_hook_aux:nn
..... 5500, 5524, 5630, 5840	_ctex_save_bounds:n
\g_ctex_mac_pingfang_dir_tl	_ctex_save_cmap:Nn
..... 5618, 5623, 5629	_ctex_save_font_size:nn
\g_ctex_mac_stfangso_dir_tl	 1938, 1940, 1949, 1976, 1999, 2030
..... 5632, 5636, 5638	\l_ctex_scheme_tl
\g_ctex_mac_yuantian_dir_tl 5655, 5659	276, 1888, 1920
\g_ctex_macos_ver_found_bool	\g_ctex_section_depth_int
..... 5366, 5373, 5383	 81, 270, 272, 4984, 5000, 5016
\g_ctex_macos_ver_iior 5365, 5369, 5371, 5386	\c_ctex_section_headings_seq
\g_ctex_macos_ver_tl 5360, 5378, 5412, 5418	 678, 1354, 1470
\c_ctex_msyh_suffix_tl	_ctex_set_font_size:nnNn 2082, 2085
... 5748, 5763, 5764, 5770, 5771, 5795, 5796	\g_ctex_spa_iow ... 6195, 6198, 6200, 6215
\c_ctex_package_cctab ... 4389, 4404, 4408	\c_ctex_std_class_tl
_ctex_package_loaded_warning:nn ..	 144, 1891, 1906, 1907, 5061, 5084
..... 4452, 4455	\g_ctex_std_options_clist
\l_ctex_parameter_str 4638, 4657, 4678, 4725	 117, 345, 357, 383, 384
\l_ctex_parameter_tl 4678, 4683, 4694	_ctex_titlesec_format:n 1466, 1474, 1477
_ctex_parse_name:nNNNnN 4590, 4599	_ctex_titlesec_format:Nn 1456, 1480, 1482
_ctex_parse_name:w 4612, 4618, 4627	_ctex_titlesec_format_explicit:Nn .
\l_ctex_patch_cleveref_bool .. 1845, 1850	 1457, 1492
_ctex_patch_cmd:Nnnnw	_ctex_titlesec_hook:n .. 1466, 1470, 1472
..... 4532, 4537, 4542, 4696, 4696	_ctex_titlesec_spacing:Nn ... 1475, 1502
_ctex_patch_false:w	_ctex_titlesec_spacing:nnnnnn 1503, 1504
..... 4568, 4574, 4711, 4713, 4730, 4756	_ctex_tmp:N
_ctex_patch_rebuild:Nn	4108, 4137
..... 4669, 4675, 4675, 4708, 4749, 4753	_ctex_tmp:w
_ctex_patch_replace:nnnTF	4578, 4631, 4642, 4663
..... 4703, 4715, 4715	\l_ctex_tmp_box
_ctex_patch_rescan:NNn	. 55, 1302, 1303, 1304, 1607, 1611, 4039, 4040
..... 4677, 4678, 4679, 4687	\l_ctex_tmp_dim
_ctex_patch_toc_width:n	55,
..... 1614, 1623, 1639, 1640	483, 484, 493, 496, 505, 543, 545, 547, 556, 558
_ctex_patch_true:w	\l_ctex_tmp_int . 55, 504, 506, 507, 513,
..... 4567, 4573, 4709, 4742, 4754	514, 545, 546, 547, 550, 551, 556, 558, 1353,
_ctex_plus_key_aux:nn	1356, 1360, 2435, 2437, 2438, 2440, 2444, 2446
..... 747, 749, 750, 751, 752, 753, 754, 757	\l_ctex_tmp_tl
\l_ctex_prefix_str	 55, 329, 336, 351, 352, 723, 777,
..... 4638, 4656, 4677, 4734, 4735	790, 797, 1803, 1804, 2054, 2055, 4110, 4139
\l_ctex_prefix_tl	_ctex_update_stretch_auxi: ... 485, 488
4677, 4682, 4693	_ctex_update_stretch_auxii:
_ctex_provide_font_hook_aux:NNNN ..	 486, 500, 4085
..... 3902, 3905	_ctex_update_stretch_auxiii:
\l_ctex_punct_bounds_clist	 501, 502, 4089
..... 6209, 6218, 6248, 6254	_ctex_use_package_hook_aux:N 4512, 4514
\c_ctex_punct_family_clist ... 2656, 2666	_ctex_write_family:nn .. 163, 6197, 6202
\l_ctex_punct_font	\l_ctex_xkanjiskip_skip
6206, 6207, 6208	 4169, 4260, 4261, 4262
\c_ctex_punct_seq	\l_ctex_xkanjiskip_tl 4176, 4181, 4182, 4259
6185, 6210	\c_ctex_zero_tl
\l_ctex_punct_tl	457, 464, 466
254,	\g_ctex_zhmap_tl
2465, 2466, 2713, 3157, 4240, 4241, 4242, 4243	2314, 2316, 2335
	\g_ctex_zhmCJK_bool
	 111, 238, 243, 248, 2395, 2470, 3899

<code>\l__ctex_ziju_dim</code>	6383, 6384, 6385, 6386, 6389, 6390, 6391,
..... 539, 540, 544, 555, 557, 561, 565	6392, 6393, 6394, 6395, 6396, 6397, 6402,
<code>\l__ctex_ziju_tl</code> ... 457, 465, 466, 533, 539	6403, 6404, 6405, 6406, 6407, 6408, 6409,
<code>ctex.newluacmd</code>	6410, 6413, 6414, 6415, 6416, 6417, 6418,
..... 2875	6419, 6420, 6421, 6424, 6425, 6426, 6427,
<code>\ctexbibname</code>	6428, 6429, 6430, 6431, 6432, 6435, 6436,
..... 651, 671	6437, 6438, 6439, 6440, 6441, 6442, 6443,
<code>\CTEXcounter</code>	6448, 6449, 6450, 6451, 6452, 6453, 6454,
..... 586	6455, 6456, 6459, 6460, 6461, 6462, 6463,
<code>\CTEXdigits</code>	6464, 6465, 6466, 6467, 6470, 6471, 6472,
..... 31, 596	6473, 6474, 6475, 6476, 6477, 6478, 6481,
<code>CTEXfilltwosides</code>	6482, 6483, 6484, 6485, 6486, 6487, 6488, 6489
..... 39, 40	
<code>\CTEXheadinglabel</code>	<code>\DeclareFontSubstitution</code>
..... 29, 798	3162
<code>\CTEXheadingnumber</code>	<code>\DeclareKanjiFamily</code>
..... 29, 798	3846, 3847, 6348, 6349, 6350, 6351, 6352, 6353
<code>\CTEXifheadingnumbering</code>	<code>\DeclareKanjiSubstitution</code>
..... 29, 798	3828, 3829
<code>\CTEXifname</code>	<code>\DeclareMathSizes</code>
..... 29, 847, 937, 979, 1121, 1333, 1341, 1349,	2047
1598, 1646, 1649, 1654, 1666, 1671, 1678,	<code>\DeclareSymbolFont</code>
1681, 1684, 1687, 1696, 1699, 1704, 1716, 1721	3172, 3507, 3830
<code>\CTEXindent</code>	<code>\def</code>
..... 39, 566	910, 951, 987,
<code>\CTEXnoindent</code>	1008, 1024, 1055, 1096, 1106, 1131, 1151,
..... 39, 566	1177, 1181, 1218, 1233, 1249, 1250, 1254,
<code>\CTEXnumber</code>	2107, 2118, 2146, 2157, 2298, 3806, 5861, 5897
..... 31, 596	<code>\defaultCJKfontfeatures</code>
<code>\CTEXnumberline</code>	3436
..... 1347, 1367, 4862	<code>\defbeamertemplate</code>
<code>\CTEXoptions</code>	1730, 1749, 1768
..... 39, 414	<code>\defbeamertemplatealias</code>
<code>ctexpar internal commands:</code>	1787, 1788, 1789
<code>__ctexpar</code>	<code>\detokenize</code>
..... 1243	2299
<code>\ctexset</code>	dim commands:
..... 5, 411, 421, 431	<code>\dim_abs:n</code>
<code>\CTEXsetfont</code>	557
..... 39, 452	<code>\dim_add:Nn</code>
<code>\CTEXsetup</code>	540
..... 39, 414	<code>\dim_compare:nNnTF</code>
<code>\CTEXsout</code>	 484, 525, 541, 548, 818, 827, 1311
..... 39, 40	<code>\dim_eval:n</code>
<code>\ctexspadef</code>	511, 556
..... 2703, 6217	<code>\dim_max:nn</code>
<code>\CTEXthechapter</code>	1610
..... 28, 1064, 1342, 1522, 1666, 1684, 1716	<code>\dim_min:nn</code>
<code>\CTEXtheparagraph</code>	557
..... 28	<code>\dim_new:N</code> 58, 565, 4049, 4192, 4193, 4194
<code>\CTEXthepart</code>	<code>\dim_set:Nn</code> 483, 493, 526, 539, 543, 571, 813,
..... 28, 1334, 1527	834, 1293, 1608, 2177, 4040, 4094, 4095, 4096
<code>\CTEXthesection</code>	<code>\dim_sub:Nn</code>
..... 28,	547
1646, 1654, 1671, 1678, 1687, 1696, 1704, 1721	<code>\dim_to_decimal:n</code>
<code>\CTEXthesubparagraph</code>	1944, 1945, 2056
..... 28	<code>\dim_to_decimal_in_unit:nn</code>
<code>\CTEXthesubsection</code>	6261
..... 28, 1649, 1681, 1699	<code>\dim_use:N</code>
<code>\CTEXthesubsubsection</code>	105, 2177
..... 28	<code>\dim_zero:N</code>
<code>\CTEXunderdblline</code>	577
..... 39, 40	<code>\c_max_dim</code>
<code>\CTEXunderdot</code>	484, 827
..... 39, 40	<code>\c_zero_dim</code>
<code>\CTEXunderline</code>	 496, 511, 525, 541, 548, 736, 818, 1311, 5017
..... 39, 40	
<code>\CTEXunderwave</code>	<code>\directlua</code>
..... 39, 40	2777, 2781
<code>\CTEXxout</code>	<code>\do</code>
..... 39, 40	2303, 3810
<code>\CurrentOption</code>	<code>\document</code>
..... 357	4493, 4496
<code>\cwd</code>	<code>\dospecials</code>
..... 4191	3810
D	
<code>\d</code>	
..... 5377	
<code>declarecharrange</code>	
..... 32, 3738	
<code>\DeclareErrorKanjiFont</code>	
..... 3827	
<code>\DeclareFontEncoding</code>	
..... 3159	
<code>\DeclareFontFamily</code>	
..... 2381, 3166, 6342, 6343, 6344, 6345, 6346, 6347	
<code>\DeclareFontShape</code>	
..... 3167, 3169, 3848,	
3849, 3852, 3853, 3854, 3855, 6356, 6357,	
6358, 6359, 6360, 6361, 6362, 6363, 6364,	
6367, 6368, 6369, 6370, 6371, 6372, 6373,	
6374, 6375, 6378, 6379, 6380, 6381, 6382,	
	E
	<code>\edef</code>
	5880, 5881
	<code>\else</code> 882, 891, 900, 916, 921, 957, 962, 1043, 1051,
	1067, 1073, 1079, 1092, 1102, 1162, 1167,
	1186, 1191, 1212, 1217, 1226, 1248, 1264,
	1277, 2293, 2296, 2297, 2301, 3805, 3808,

- 3835, 3836, 3840, 3841, 4825, 4853, 5892,
5893, 5917, 5941, 5990, 6039, 6063, 6095, 6144
- else commands:
- \else: 4070
 - \em 3833
 - \eminnershapes 3835, 3836, 3841, 3843
 - \end 1746, 1765, 1784, 6330
 - \endcsname 1169, 1178, 1179,
1180, 1187, 1207, 1208, 1210, 1213, 1222,
1223, 1224, 1227, 1872, 1873, 2292, 2295,
2296, 5887, 5889, 5890, 5891, 5894, 5896, 5900
 - \endgroup 942, 983,
999, 1020, 1125, 1144, 1209, 1247, 1272,
1748, 1767, 1786, 2293, 2310, 5880, 5888, 5903
 - \endinput 2293, 5885
 - \endlinechar 5859, 5860, 6234, 6236
 - \everypar 1166, 1267, 1279
 - \EverySelectfont 4147
- exp commands:
- \exp_after:wN 104, 1503, 1928,
2047, 2082, 2813, 2978, 2994, 2996, 3067, 4647
 - \exp_args:cc 1828
 - \exp_args:Nc ... 1475, 1480, 2357, 3902, 4601
 - \exp_args:Nco 1596
 - \exp_args:Ne 530, 579, 843, 1924, 2175, 2354,
2365, 2397, 2436, 2475, 2484, 2568, 2710, 3908
 - \exp_args:NNc 2834, 2943, 4511
 - \exp_args:Nnc 78
 - \exp_args:NNe 723, 2033, 2035
 - \exp_args:Nne .. 1841, 2540, 2547, 2554, 4424
 - \exp_args:NNNo 2176
 - \exp_args:NNo 797
 - \exp_args:Nno 2383
 - \exp_args:No
. 121, 351, 1499, 2565, 3075, 3533, 4663, 6212
 - \exp_last_unbraced:Ne 2493, 2509, 2526, 4612
 - \exp_last_unbraced:NNNo 107
 - \exp_last_unbraced:NNo 4138
 - \exp_not:N 697, 698, 699,
704, 706, 707, 708, 709, 712, 713, 716, 728,
729, 730, 731, 732, 741, 742, 743, 745, 746,
759, 761, 763, 1359, 1361, 1362, 1363, 1364,
1487, 1488, 1497, 1498, 1535, 1537, 1538,
1841, 2240, 2466, 2477, 2478, 2479, 2543,
2544, 2550, 2551, 2557, 2558, 2828, 2829,
2968, 3015, 3047, 3162, 3244, 3245, 3582,
3947, 3948, 3949, 3961, 4592, 4593, 4594, 4682
 - \exp_not:n 103, 144, 1506,
1829, 2545, 2552, 2559, 3241, 3242, 3243,
3374, 3375, 3583, 3584, 3585, 3610, 3666,
3667, 3695, 3942, 3962, 4100, 4361, 4496,
4567, 4568, 4610, 4682, 4683, 4684, 4739, 4740
 - \expandafter 1873, 2292,
2293, 2295, 2296, 2304, 2308, 5862, 5863,
5887, 5888, 5889, 5890, 5891, 5896, 5900, 5901
 - experiment 4248
 - experiment/CJKecglue 13, 4248
 - experiment/font-size-system 9, 159
 - \ExplFileDate 5912
 - \ExplFileDescription 5912
 - \ExplFileVersion 5912
 - \ExplSyntaxOff 1619, 1729,
1869, 2755, 2841, 2845, 3834, 4140, 4545, 6267
 - \ExplSyntaxOn 1790, 2792, 6178
- ## F
- \familydefault 3945
 - fancyhdr 38, 311
 - \fangsong 5821
 - \fi 873, 878, 884, 893, 902,
905, 920, 925, 935, 946, 961, 966, 977, 1003,
1027, 1035, 1036, 1039, 1043, 1053, 1071,
1077, 1083, 1095, 1105, 1109, 1118, 1119,
1129, 1134, 1148, 1152, 1158, 1164, 1171,
1173, 1195, 1196, 1214, 1228, 1231, 1252,
1261, 1280, 1281, 1645, 1648, 1653, 1663,
1664, 1670, 1677, 1680, 1683, 1686, 1695,
1698, 1703, 1713, 1714, 1720, 1742, 1761,
1780, 2293, 2296, 2297, 2305, 2306, 2307,
2309, 3808, 3815, 4825, 4853, 5892, 5893,
5902, 5936, 5960, 6017, 6058, 6090, 6114, 6171
 - fi commands:
 - \fi: 3934, 4070 - \figurename 632
 - figurename 14
 - file commands:
 - \g_file_curr_name_str 2725
 - \file_get_full_name:nTF 2365
 - \file_if_exist:nTF
. 28, 2003, 2219, 2234, 5367, 5409, 5749, 5752
 - \file_if_exist_input:nTF 74
 - \file_if_exist_p:n 163, 6228, 6230
 - \file_input:n
..... 131, 75, 77, 2089, 2090, 2427, 4379
 - \file_input_stop: 4482
 - \fmtversion 4375
 - fnf 38, 40, 292
 - \font 6342, 6343, 6344, 6345, 6346, 6347
 - fontset 7, 227, 2253
 - \fontsize 1928 - fontspec commands:
 - \fontspec_font_if_exist:nTF
5503, 5520, 5527, 5533, 5538, 5540, 5725, 5731
 - \fontspec_gset_family:Nnn 3242
 - \fontspec_set_family:Nnn 3619 - fontspec internal commands:
 - __fontspec_fontname_wrap:n 107, 3184
 - \footnotesep 2176
 - \footnotesize 2113, 2152, 2176 - fp commands:
 - \fp_eval:n 533, 6257
 - \fp_if_nan:nTF 2173, 5056, 5082

- \fp_set:Nn 5057, 5083
 - \fp_use:N 464, 2175
 - \c_nan_fp 178
 - \c_zero_fp 464
- G**
- GBK 6, 40, 215
 - \gdef 1385, 1386, 1387, 1388, 1391, 1392, 1393, 1394, 2300, 2306
 - \global 1047, 1266, 1269
 - \globaljfont 2996
 - \glueexpr 1736, 1747, 1755, 1766, 1774, 1785
 - group commands:
 - \group_begin: 703, 711, 775, 1309, 1409, 1555, 1614, 1950, 2176, 2562, 2798, 2871, 2943, 2953, 2980, 3008, 3046, 3068, 3114, 3189, 3240, 3614, 3692, 3710, 3727, 3941, 4107, 4195, 4490, 4511, 4569, 4577, 4586, 4641, 6204
 - \group_end: 714, 718, 797, 1313, 1426, 1578, 1618, 2037, 2176, 2567, 2803, 2940, 2943, 2959, 2984, 3021, 3050, 3072, 3118, 3204, 3247, 3620, 3696, 3723, 3734, 3953, 4138, 4212, 4490, 4511, 4589, 4636, 4664, 4698, 4722, 6220
 - \group_insert_after:N 3018, 3019
 - \gtfamily 3835, 3840
- H**
- \hbox 3808
 - hbox commands:
 - \hbox_set:Nn 1302, 1607, 4039
 - heading 10, 265
 - \heiti 5821
 - \highLight 2739
 - hook commands:
 - \hook_gput_code:nnn 4471
 - \hskip ... 1204, 1220, 1242, 1249, 1275, 1677, 1680
 - \hspace 915, 956, 1334, 1342, 1527
 - \Huge . 980, 1017, 1122, 1141, 2130, 2169, 4829, 4860
 - \huge 938, 976, 996, 1117, 2129, 2168, 4811, 4828, 4832, 4858, 4866
 - hyperref 38, 316
 - \hypersetup 321, 322, 403
 - \hyphenchar .. 6342, 6343, 6344, 6345, 6346, 6347
- I**
- \if 2299
 - if commands:
 - \if_bool:N 3899
 - \if_false: 96
 - \if_meaning:w 4069
 - \if_true: 95
 - \ifctexpdf 94
 - \ifdim 1156, 1199, 1237, 1256
 - \ifmmode 3808
 - \ifnum 911, 933, 952, 975, 1056, 1115, 1182, 1212, 1226, 1645, 1648, 1653, 1660, 1661, 1670, 1677, 1680, 1683, 1686, 1695, 1698, 1703, 1710, 1711, 1720, 2297, 5892
 - \ifodd 878, 880, 905, 912, 946, 953, 1003, 1027, 1049, 1060, 1109, 1129, 1134, 1148, 1160, 1171, 1187, 1200, 1238, 1257, 1261, 1740, 1759, 1778, 2294, 2306
 - \ifx 2292, 2295, 2296, 5890, 5891, 5896
 - \ifzhmappdf 5888, 5915, 5939, 5963, 6020, 6061, 6093, 6117
 - \ignorespaces 1282
 - \immediate 5899
 - indent 38, 197
 - \indexname 635
 - indexname 14
 - \input 6177, 6270
 - \insertcontinuationtext 644, 653, 660
 - \insertpart 1744, 1745
 - \insertromanpartnumber 1734, 4798
 - \insertsection 1763, 1764
 - \insertsectionnumber 1753, 4896
 - \insertsubsection 1782, 1783
 - \insertsubsectionnumber 1772, 4925
 - int commands:
 - \int_case:nn 381, 2087
 - \int_compare:nNnTF 507, 550, 2437, 2816, 4984, 5000, 5016, 5054, 5080, 5418
 - \int_const:Nn 3176, 3778, 3779
 - \int_div_truncate:nn 53, 2444
 - \int_eval:n 1805, 3770, 3771, 3774, 3775
 - \int_from_hex:n 2436
 - \int_gincr:N 3332, 3633
 - \int_gset:Nn 83, 153, 167, 168, 169, 270, 272, 340, 344, 5055, 5081
 - \int_if_odd:nTF 4395
 - \int_incr:N 1356
 - \int_mod:nn 2446
 - \int_new:N 56, 82, 152, 3636
 - \int_set:Nn 504, 545, 808, 2426, 2435
 - \int_sub:Nn 551, 2440
 - \int_to_Hex:n 2438, 2443, 2445
 - \int_use:N 1360, 3334, 3634, 6213
 - \int_value:w 2484, 2495, 2496, 2511, 2512, 2513, 2528, 2529, 2530, 2531, 2565, 2571
 - \int_zero:N 1353
 - \c_max_char_int 3779
 - \c_zero_int 507
 - \interlinepenalty 930, 972, 992, 1013, 1137, 1205, 1243
 - ior commands:
 - \ior_close:N 5386
 - \ior_map_break: 5379
 - \ior_map_inline:Nn 5371
 - \ior_new:N 5365
 - \ior_open:Nn 5369

iow commands:

`\iow_close:N` 6198
`\iow_indent:n` 4366
`\iow_new:N` 6200
`\iow_now:Nn` 6215
`\iow_open:Nn` 6195
`\itemsep` 2110, 2121, 2149, 2160
`\itshape` 3835, 3836, 3840, 3841

J

`\jfam` 3106, 3177, 3832

K

`\kaishu` 5821
`\kanjifamily` 3826, 3929, 3930, 3931
`\kanjifamilydefault` 3932
`\kern` 3816

kernel internal commands:

`\l__kernel_expl_bool`
 4382, 4387, 4388, 4396, 4397

keys commands:

`\keys_define:nn`
 ... 61, 64, 3151, 3181, 3520, 3630, 3784, 4250
`\keys_if_exist:nnTF` 3782
`\l_keys_key_str` 129, 131, 147
`\keys_set:nn` 66, 68
`\keys_set_known:nn` 4762
`\keys_set_known:nnN` 3193

L

`\labelformat` 1841, 1877
`\language` 3812
`\languagealias` 5041, 5046, 5066, 5071
`\LARGE` 2128, 2167
`\Large` 934, 2127, 2166, 4809, 4815, 4883, 4887
`\large` 2126, 2165, 4908
`\lastbox` 1270
`\leavevmode` 873, 1152, 3808
`\leftmargin` 2107, 2118, 2146, 2157
`\leftmargini` 2107, 2118, 2146, 2157

legacy commands:

`\legacy_if:nTF`
 ... 1410, 1553, 1642, 1667, 1692, 1717, 3062
`\let` 1185, 1194, 1547,
 1549, 2100, 2139, 2302, 2305, 3810, 5883, 5889
`\linespread` 2175
`linespread` 12, 177
`linestretch` 12, 469
`\linewidth` 505, 513, 544
`\lishu` 5821
`\listfigurename` 630
`listfigurename` 14
`\listtablename` 631
`listtablename` 14
`\LoadClass` 124
`\long` 5897

`\ltjdefcharrange` 2756,
 2757, 2758, 2760, 2765, 2766, 2770, 2773, 2774
`LTJFONTUID` 3630
`\ltjgetparameter` 4044, 4073, 4173, 4187
`\ltjlineendcomment` 2754
`\ltjsetkanjiskip` 3819
`\ltjsetmathletter` 3116
`\ltjsetparameter` 2776, 2779, 2785, 3803
`\ltjsetxkanjiskip` 3821

lua commands:

`\lua_now:n` 2873, 4197, 5391, 5551

M

`\makeatletter` 2841, 2845
`\MAKESPA` 6201, 6271
`\markboth` 846, 940, 969
`\mathgroup` 3106
`\mcfamily` 3840
`\mddefault` 3163, 3167, 3173, 3508, 3513

mode commands:

`\mode_if_math:TF` 3100
`\mode_leave_vertical:`
 93, 2576, 2582, 2588, 2594

msg commands:

`\msg_critical:nnn` 33, 2193, 2725, 2742
`\msg_error:nn` 2273
`\msg_error:nnn` 22,
 617, 1929, 2009, 2083, 2716, 2750, 2753, 3414
`\msg_error:nnnn` 2239, 2268, 4416
`\msg_fatal:nn` 6183
`\g_msg_module_name_prop` .. 6, 7, 8, 9, 10, 11
`\g_msg_module_type_prop` 5
`\msg_new:nnn` 128,
 130, 132, 146, 148, 371, 648, 1897, 2194,
 2247, 2282, 2719, 2733, 3234, 3300, 3382,
 3407, 4362, 4429, 4443, 4457, 4553, 5361, 6181
`\msg_new:nnnn`
 14, 25, 88, 619, 1931, 2038, 2249, 2276
`\msg_redirect_name:nnn` 2728, 2730
`\msg_warning:nn`
 .. 220, 376, 670, 3377, 4233, 4244, 4267, 5414
`\msg_warning:nnn`
 135, 1916, 2260, 3295, 4368, 4456, 4552
`\msg_warning:nnnn` 143, 2229, 3230, 4425, 4442

N

`\newCJKfontfamily` 3401
`\NewDocumentCommand` 3, 412, 415, 425, 452, 529,
 566, 573, 596, 598, 765, 1923, 2180, 3261,
 3396, 3401, 3409, 3424, 3430, 3436, 3443,
 3449, 3455, 3461, 3658, 3758, 3825, 5822,
 5823, 5824, 5825, 5827, 5828, 5830, 5842, 5843
`\NewExpandableDocumentCommand` .. 798, 800, 802
`\newfontfeature` 3178, 3179, 3180
`\newpage` 1029, 1034
`\ngostype` 24

- \nobreak 934, 943, 1000, 1117, 1122, 1126, 1141, 1145, 1258, 4810, 4859
 - \nobreakspace 934, 976
 - nocap 38, 288
 - \noexpand 5881, 5883, 5884, 5885
 - nofonts 38
 - noindent 38
 - nopunct 38
 - \normalem 4013
 - \normalfont .. 806, 930, 972, 992, 1013, 1112, 1137, 1364, 3440, 3447, 3453, 3491, 3913, 3935
 - \normalsize 2094, 2133, 2171, 2179, 4942, 4959, 4973
 - nospace 38
 - nozhmap 38
 - \null 903, 1032
 - \numberline 1066, 1213, 1227, 1342, 1350, 1538, 1614
- O**
- \onecolumn 898
- P**
- \pagestyle 5032
 - \par 833, 874, 934, 940, 941, 976, 980, 982, 996, 998, 1017, 1019, 1117, 1122, 1124, 1141, 1143, 1153, 1208, 1246, 1258, 1735, 1736, 1744, 1747, 1754, 1755, 1763, 1766, 1773, 1774, 1782, 1785, 4788, 4810, 4821, 4839, 4847, 4859, 4899, 4901, 4933, 4935
 - paragraph/afterskip 40
 - paragraph/beforeskip 39
 - paragraph/numbering 17
 - paragraph/runin 23
 - \parindent 525, 526, 548, 569, 571, 576, 577, 930, 992, 1112, 1137, 5018
 - \parsep 2109, 2110, 2120, 2121, 2148, 2149, 2159, 2160
 - \part 871
 - part/beforeskip 39
 - part/fixbeforeskip 40
 - part/numbering 17
 - part/pagestyle 23, 775
 - \partmark 845, 928, 970
 - \partname 934, 976, 1734, 4793
 - \PassOptionsToClass 121
 - \PassOptionsToPackage 407, 579, 2397, 3969, 4370
 - patch/cleveref 35, 1843
 - \patchcmd 3816
- pdf commands:
- \pdf_object_ref_last: 2375
 - \pdf_object_unnamed_write:nn 2368
 - \pdffontattr 2455, 2458
 - \pdfmapline 5964, 5965, 5966, 5967, 5968, 5969, 5970, 5971, 5972, 5973, 5974, 5975, 5976, 5977, 5978, 5979, 5980, 5981, 5982, 5983, 5984, 5985, 5986, 5987, 5988, 5989, 6021, 6022, 6023, 6024, 6025, 6026, 6027, 6028, 6029, 6030, 6031, 6032, 6033, 6034, 6035, 6036, 6037, 6038, 6118, 6119, 6120, 6121, 6122, 6123, 6124, 6125, 6126, 6127, 6128, 6129, 6130, 6131, 6132, 6133, 6134, 6135, 6136, 6137, 6138, 6139, 6140, 6141, 6142, 6143
 - \pdfnobuiltintounicode 2373, 2457
 - \pdfoutput 2297, 5892
 - \pdfstringdefDisableCommands 2183
 - \pingfang 5821
- prg commands:
- \prg_do_nothing: 1326, 2332, 2378, 3110, 3608, 3693, 4102, 4220, 4674
 - \prg_generate_conditional_variant:Nnn 3287
 - \prg_new_conditional:Npnn 3024, 4066
 - \prg_new_protected_conditional:Nnn . 6222
 - \prg_new_protected_conditional:Npnn 3274, 4644
 - \prg_return_false: 3027, 3284, 4070, 4074, 4078, 4654, 6242
 - \prg_return_true: 3026, 3277, 3282, 4070, 4074, 4078, 4659, 6242
 - \primitive 6330
 - \ProcessKeyOptions 363, 364
 - \ProcessKeysOptions 367
 - \proofname 622, 637
 - proofname 14
- prop commands:
- \prop_clear:N 1951, 3191
 - \prop_const_from_keyval:Nn 1808, 2033
 - \prop_get:NnN 3369
 - \prop_get:NnNTF 1803, 1927, 2054, 2081, 3139, 3276, 3328, 3366, 3601, 3647
 - \prop_gpop:NnNTF 3219, 3223, 3728
 - \prop_gput:Nnn 5, 6, 7, 8, 9, 10, 11, 3197, 3198, 3244, 3356, 3716, 3752
 - \prop_gremove:Nn 3228
 - \prop_if_empty:NTF 3290, 3472, 3552
 - \prop_map_break:n 3486
 - \prop_map_inline:Nn 3484, 3560
 - \prop_new:N 1938, 3143, 3214, 3215, 3216, 3361, 3542, 3736, 3757
 - \prop_put:Nnn 1942, 3539
 - \prop_remove:Nn 3538
 - \prop_set_from_keyval:Nn 3144
 - \prop_to_keyval:N 2034
 - \protect 859, 865, 1066, 1087, 1088, 1213, 1227, 1342, 1350, 1537, 1560, 1562, 1566, 1568, 1572, 1574, 1582, 1587, 1599, 1600, 2487, 2501, 2518, 4605, 4606
 - \ProvideDocumentCommand 845
 - \ProvidesFile 5895, 5905, 5906, 5907, 5908, 5909, 5910, 5911
 - \providetranslation 4276, 4277, 4278, 4279, 4280, 4281, 4282, 4283, 4284, 4285, 4286, 4287, 4288, 4289, 4290, 4291, 4292, 4293, 4294, 4295, 4296, 4297, 4298,

4299, 4300, 4301, 4302, 4303, 4304, 4305,
4306, 4307, 4308, 4309, 4310, 4311, 4312,
4313, 4314, 4315, 4316, 4317, 4318, 4319,
4320, 4321, 4322, 4323, 4324, 4325, 4326,
4327, 4328, 4329, 4330, 4331, 4332, 4333,
4334, 4335, 4336, 4337, 4338, 4339, 4340,
4341, 4342, 4343, 4344, 4345, 4346, 4347,
4348, 4349, 4350, 4351, 4352, 4353, 4354, 4355
punct 11, 38, 254, 4236
\punctstyle 2466, 4241

Q

\quad 1178, 1645, 1646, 1648, 1649, 1653,
1654, 1666, 1671, 1678, 1681, 1684, 1687,
1695, 1696, 1698, 1699, 1703, 1704, 1716,
1721, 4816, 4867, 4876, 4909, 4943, 4960, 4974
quark commands:
 \q_mark 3561, 3569, 4648, 4651
 \q_stop 105, 108, 3061, 3064, 3067,
 3070, 4613, 4618, 4627, 4648, 4651, 5408, 5665
 \quotation 5089

R

\raggedright 930, 992, 1112, 1137, 4808, 4857
\refname 638, 643, 650, 659, 667, 668
refname 14
\refstepcounter
 914, 955, 1062, 1189, 1823, 1880, 1881
regex commands:
 \regex_replace_once:nnN 5376
\relax 952,
 975, 1154, 1190, 1198, 1204, 1220, 1236,
 1249, 1255, 1274, 1677, 1680, 1736, 1747,
 1755, 1766, 1774, 1785, 2292, 2295, 2296,
 2299, 3808, 5854, 5859, 5862, 5890, 5891, 5896
\renewcommand 872, 888, 1042
\RenewDocumentCommand 1881
\RequirePackage .. 4, 36, 37, 51, 297, 303, 314,
 324, 366, 582, 2400, 2404, 2405, 2406, 2709,
 2748, 2751, 2865, 2870, 3896, 3974, 3984, 5085
\RequirePackageWithOptions 4371
resetalternatefont 33, 3699
reverse commands:
 \reverse_if:N 3899
\rmdefault 3947
\rmfamily 3910

S

scan commands:
 \scan_stop: 116, 105, 494,
 505, 2829, 2996, 3106, 3694, 4492, 6206, 6207
scheme 11, 276
\scriptsize 2124, 2163
\secdef 885, 907, 1054
secnumdepth 18, 1792
section/afterskip 40
section/beforeskip 39

section/numbering 17
section/runin 23
\sectionname 1753, 4893
\selectfont 454, 1928, 3268, 3719,
 3732, 3826, 3910, 3911, 3912, 4137, 4144, 4166
seq commands:
 \seq_clear:N 1952
 \seq_const_from_clist:Nn
 679, 682, 1399, 2035, 2339, 6185
 \seq_gpop:NN 4394
 \seq_gpop:NNTF 2972
 \seq_gpush:Nn 2967, 4386
 \seq_gput_right:Nn 3294
 \seq_if_in:NnTF 2345, 3292
 \seq_map_function:NN 1470, 1603
 \seq_map_inline:Nn 792,
 1354, 1402, 1413, 1423, 1446, 1529, 1834, 6210
 \seq_new:N 1939, 2975, 3299, 4400
 \seq_put_right:Nn 1947
 \seq_use:Nn 2036
 \seq_use:Nnnn 1936
\setbox 1270
\setCJKfamilyfont 3313,
 3401, 5113, 5115, 5117, 5119, 5155, 5156,
 5157, 5158, 5177, 5179, 5181, 5183, 5226,
 5228, 5230, 5232, 5245, 5246, 5247, 5248,
 5249, 5250, 5291, 5293, 5294, 5295, 5296,
 5297, 5310, 5312, 5314, 5315, 5351, 5353,
 5355, 5356, 5446, 5447, 5448, 5449, 5450,
 5451, 5494, 5495, 5496, 5497, 5516, 5518,
 5523, 5530, 5535, 5539, 5542, 5614, 5616,
 5628, 5637, 5642, 5652, 5657, 5680, 5682,
 5684, 5685, 5728, 5734, 5738, 5741, 5743,
 5745, 5766, 5767, 5768, 5769, 5770, 5772,
 5773, 5812, 5813, 5814, 5815, 5816, 5817, 5818
\setCJKmainfont 3309, 3436, 5103, 5151, 5167,
 5216, 5241, 5287, 5306, 5347, 5438, 5490,
 5505, 5513, 5601, 5605, 5674, 5721, 5761, 5809
\setCJKmathfont 3436
\setCJKmonofont
 3311, 3436, 5111, 5154, 5175, 5224, 5244,
 5290, 5309, 5350, 5445, 5493, 5529, 5532,
 5633, 5635, 5678, 5727, 5733, 5737, 5765, 5811
\setCJKromanfont 3442, 3468
\setCJKsansfont 3310, 3436, 5109, 5153, 5173,
 5222, 5243, 5289, 5308, 5349, 5444, 5492,
 5522, 5526, 5619, 5621, 5676, 5723, 5763, 5810
\setcounter 1804, 1805
\SetSymbolFont 3174, 3512, 3516, 3831
\sfdefault 3948
\sffamily 3911
\shapedefault 3163,
 3167, 3169, 3173, 3175, 3508, 3510, 3513, 3517
skip commands:
 \skip_horizontal:N 1312, 4054
 \skip_horizontal:n 4254

<code>\skip_if_eq:nnTF</code> .	856, 862, 4073, 4077, 4172	<code>sub3section</code>	11, 268
<code>\skip_new:N</code>	837, 4065, 4185	<code>sub4section</code>	11, 268
<code>\skip_set:Nn</code>	495, 509, 553, 561, 839, 855, 861, 1310, 4044, 4047, 4176, 4186, 4260	<code>subparagraph/afterskip</code>	40
<code>\skip_set_eq:NN</code>	4063, 4178, 4262	<code>subparagraph/beforeskip</code>	39
<code>\skip_sub:Nn</code>	828, 835	<code>subparagraph/numbering</code>	17
<code>\skip_use:N</code>	859, 865	<code>subparagraph/runin</code>	23
<code>\skip_zero:N</code>	518	<code>subsection/afterskip</code>	40
<code>\c_zero_skip</code>	734, 735, 783, 784, 856, 862	<code>subsection/beforeskip</code>	39
<code>\small</code>	2102, 2141	<code>subsection/numbering</code>	17
<code>\songti</code>	5821	<code>subsection/runin</code>	23
<code>\space</code>	1063, 1117, 2297, 4793, 4855, 4893, 4924, 5025, 5912	<code>\subsectionname</code>	1772, 4924
<code>space</code>	11, 38, 259, 4215	<code>subsubsection/afterskip</code>	40
<code>\special</code>	3862, 3863, 3866, 3867, 3875, 3876, 3879, 3880, 3982, 4006, 5918, 5919, 5920, 5921, 5922, 5923, 5924, 5925, 5926, 5927, 5928, 5929, 5930, 5931, 5932, 5933, 5934, 5935, 5942, 5943, 5944, 5945, 5946, 5947, 5948, 5949, 5950, 5951, 5952, 5953, 5954, 5955, 5956, 5957, 5958, 5959, 5991, 5992, 5993, 5994, 5995, 5996, 5997, 5998, 5999, 6000, 6001, 6002, 6003, 6004, 6005, 6006, 6007, 6008, 6009, 6010, 6011, 6012, 6013, 6014, 6015, 6016, 6040, 6041, 6042, 6043, 6044, 6045, 6046, 6047, 6048, 6049, 6050, 6051, 6052, 6053, 6054, 6055, 6056, 6057, 6064, 6065, 6066, 6067, 6068, 6069, 6070, 6071, 6072, 6073, 6074, 6075, 6076, 6077, 6078, 6079, 6080, 6081, 6082, 6083, 6084, 6085, 6086, 6087, 6088, 6089, 6096, 6097, 6098, 6099, 6100, 6101, 6102, 6103, 6104, 6105, 6106, 6107, 6108, 6109, 6110, 6111, 6112, 6113, 6145, 6146, 6147, 6148, 6149, 6150, 6151, 6152, 6153, 6154, 6155, 6156, 6157, 6158, 6159, 6160, 6161, 6162, 6163, 6164, 6165, 6166, 6167, 6168, 6169, 6170	<code>subsubsection/beforeskip</code>	39
<code>\SplitArgument</code>	766, 3659, 3759	<code>subsubsection/numbering</code>	17
<code>str</code> commands:		<code>subsubsection/runin</code>	23
<code>\c_backslash_str</code>	4632	<code>\symmincho</code>	3832
<code>\c_colon_str</code>	3093, 3094, 3095	<code>sys</code> commands:	
<code>\c_dollar_str</code>	5749	<code>\c_sys_engine_str</code>	24
<code>\c_left_brace_str</code>	4633	<code>\sys_get_shell:nnN</code>	6234, 6236
<code>\str_case:nnTF</code>	1953, 3945, 4603, 4618	<code>\sys_if_engine luatex:TF</code>	5363, 5501
<code>\str_case_e:nnTF</code>	3307, 3318	<code>\sys_if_engine pdftex:TF</code>	217, 300, 375, 2198, 2204
<code>\str_const:Nn</code>	23, 30	<code>\sys_if_engine pdftex_p:</code>	5833
<code>\str_if_empty:N</code>	4689, 4725, 4734	<code>\sys_if_engine uptex:TF</code>	2200, 2206
<code>\str_if_eq:nnTF</code>	2259, 2262, 2279, 2284, 2403, 2465, 2472, 2538, 3476, 3970, 3979, 4609, 5037, 5063, 5077	<code>\sys_if_engine uptex_p:</code>	5834
<code>\str_if_eq_p:nn</code>	2226, 2227	<code>\sys_if_engine xetex:TF</code>	294, 6179
<code>\str_lowercase:n</code>	2366	<code>\sys_if_output_pdf:TF</code> .	94, 2205, 2381, 3973
<code>\str_new:N</code>	4638, 4639, 4640	<code>\sys_if_platform_windows:TF</code>	2210
<code>\str_set:Nn</code>	4656, 4657, 4658		
<code>\strikeThrough</code>	2738		
<code>\strutbox</code>	2177		

T

<code>\tablename</code>	633
<code>tablename</code>	14
<code>\TeX</code>	2181
<code>TeX</code> and <code>L^AT_EX</code> commands:	
<code>\@@ifdefinable</code>	2806
<code>\@@italiccorr</code>	3817
<code>\@@par</code>	1205, 4877, 4910, 4944, 4988, 5004
<code>\@afterheading</code> ..	948, 1005, 1094, 1104, 1263
<code>\@afterindentfalse</code>	883, 1048, 1052, 1157, 1163
<code>\@afterindenttrue</code> ...	881, 1050, 1155, 1161
<code>\@Alph</code>	5021, 5026
<code>\@break@tfor</code>	2305
<code>\@chapapp</code>	1063, 1117, 1662, 1683, 1712
<code>\@chapter</code>	1054, 1055
<code>\@chinese</code>	584, 593
<code>\@classoptionslist</code>	87, 2300, 2306, 2308
<code>\@clubpenalty</code>	1278
<code>\@currentHref</code>	1435
<code>\@currentlabel</code>	1825, 1829
<code>\@currext</code>	133, 4361, 4519, 4525
<code>\@currname</code>	133, 4361, 4418, 4520, 4526
<code>\@dblarg</code>	1176
<code>\@defaultfamilyhook</code>	3913
<code>\@empty</code>	78, 1185, 1194
<code>\@endpart</code>	984, 1021, 1023
<code>\@EverySelectfont@Init</code> ...	4105, 4120, 4132

\@firstofone	101	\@startsection	68, 71, 139, <u>1151</u> , 1359
\@firstoftwo	1599	\@sverb	3813
\@gobble	101	\@svsec 1185, 1190, 1194, 1204, 1206, 1220, 1221	
\@changfrom	1204, 1242	\@svsechd	1218, 1249, 1250, 1272
\@ifclassloaded	1893	\@tempdima	1608, 1610
\@ifdefinable	2806	\@tempskipa . 1154, 1156, 1157, 1168, 1198,	
\@ifl@t@r	4375	1199, 1236, 1237, 1255, 1256, 1259, 1274, 1275	
\@ifpackagelater		\@tempswafalse	901
..... 21, 2715, 2749, 2752, 2778, 4020		\@tempwattrue	899
\@ifpackageloaded 400, 1509, 1625, 1852,		\@text@composite@x	99, 2856
2724, 2741, 2860, 3967, 3994, 4415, 4451, 4464		\@tfor	2303
\@ifpackagewith 1454, 1463, 1514, 1689, 1854		\@topnewpage	1091, 1101
\@ifstar	1174, 3813	\@topnum	1047
\@listI	2100, 2139	\@ttfamilyhook	3912
\@listi ... 2100, 2107, 2118, 2139, 2146, 2157		\@verb	3813
\@M . 930, 972, 992, 1013, 1137, 1205, 1243, 1271		\@xsect	1232, 1253, <u>1254</u>
\@makechapterhead	1091, 1093, <u>1106</u>	\@zhdig	592
\@makeother	3810	\@zhnum	584, 591
\@makeschapterhead	1101, 1103, <u>1131</u>	\abstractname	14
\@minus	2096, 2098, 2104,	\add@unicode@accent	99, 2857
2106, 2108, 2109, 2115, 2117, 2119, 2120,		\addCJKfontfeature	116
2135, 2137, 2143, 2145, 2147, 2148, 2154,		\AfterEndPreamble	133
2156, 2158, 2159, 4878, 4911, 4945, 4961, 4975		\algorithmname	14
\@namedef	1560,	\appendixname	14, 28
1562, 1566, 1568, 1572, 1574, 1582, 1587, 3804		\AtBeginDocument	45, 91
\@nameuse	1560, 1562, 1572, 1574, 1582	\AtBeginDvi	88
\@ne	1507, 1648, 1680, 1698, 2302	\AtBeginShipoutFirst	88
\@nil	3047	\AtEndOfClass	45
\@nobreakfalse	1265	\AtEndOfPackage	45
\@noligs	3811	\AtEndOfPackageFile*	132
\@noskipsecfalse	1269	\AtEndPreamble	133
\@noskipsectrue	1266	\baselineskip	12, 13, 53, 127
\@onlypreamble		\bfseries	95
..... 45, 2252, 2336, 2414, 3464, 3465,		\bibname	14, 36
3466, 3467, 3468, 3469, 3891, 3892, 3893, 3894		\blx@	37
\@part	885, 907, <u>909</u>	\c@chapter	5026
\@pkgextension . 4427, 4436, 4437, 4519, 4525		\c@secnumdepth	911, 933, 952,
\@plus	2096, 2097,	975, 1056, 1115, 1182, 1212, 1226, 1645,	
2098, 2104, 2105, 2106, 2108, 2109, 2115,		1648, 1653, 1660, 1661, 1670, 1677, 1680,	
2116, 2117, 2119, 2120, 2135, 2136, 2137,		1683, 1686, 1695, 1698, 1703, 1710, 1711, 1720	
2143, 2144, 2145, 2147, 2148, 2154, 2155,		\c@section	5021
2156, 2158, 2159, 4822, 4823, 4878, 4879,		\caption	39
4911, 4912, 4945, 4946, 4961, 4975, 4989, 5005		\catcode	97, 134, 157
\@popfilename	134, <u>4522</u>	\catcodetable	97
\@pushfilename	133	\ccwd	12, 25, 31, 39, 45, 52–54, 124, 125
\@reset@ptions	133, <u>4516</u>	\chapter	10, 18, 23, 28, 29, 60, 67, 80
\@rmfamilyhook	3910	\chaptermark	15
\@schapter	1054, <u>1096</u>	\chaptername	18
\@seccntformat	<u>1177</u> , 1190	\char	99
\@secondoftwo	1600	\chardef@text@cmd	98, 2854
\@secpenalty 1168, 4881, 4914, 4948, 4962, 4976		\CheckCommand	125
\@sect	1176, <u>1181</u>	\Chinese	39
\@setfontsize	2086	\chinese	31, 39, 55
\@sffamilyhook	3911	\CJK@@ignorespaces	92, 2482
\@spart	885, 907, <u>986</u>	\CJK@addcmap	<u>2452</u>
\@ssect	1175, <u>1233</u>		

- \CJK@encoding 107, 108,
115, 2950, 2954, 3041, 3158, 3212, 3507,
3510, 3512, 3516, 3656, 3666, 3667, 3685, 3686
- \CJK@envStart 91, 2477
- \CJK@family 102, 2941, 2955,
3009, 3041, 3265, 3718, 3728, 3730, 4166, 4167
- \CJK@ignorespaces
..... 95, 2578, 2584, 2590, 2596, 2647, 2651
- \CJK@input 2423
- \CJK@loadBinding 2473, 2474
- \CJK@makeActive 91
- \CJK@plane 90, 126, 2353, 2354, 4155
- \CJK@surr 90, 2431
- \CJK@upperReset 91
- \CJK@X 92, 2483
- \CJK@XX 2483
- \CJK@XXX 2483
- \CJK@XXXX 2483
- \CJKfamily 108, 119
- \CJKfamilydefault 91, 92, 112, 122
- \CJKglue 53, 85
- \CJKhook 95
- \CJKkrmdefault 90
- \CJKsymbol 126
- \CJKunderdotbasesep 39
- \CJKunderline 39
- \clearpage 91
- \contentsname 13
- \CS 103
- \CTEX@add 2302, 2305, 2306
- \CTEX@add@unicode@accent 2825, 2857
- \CTEX@addloflotskip 852, 1089
- \CTEX@addtocline
..... 869, 927, 968, 1085, 1216, 1230
- \CTEX@afterindent 1160, 1320, 1327
- \CTEX@aftertitle ... 1246, 1251, 1319, 1326
- \CTEX@alchar 2794, 2817
- \CTEX@appendix@number 1374, 1386, 1392
- \CTEX@appendix@numbering . 1375, 1388, 1394
- \CTEX@beginallalchar
..... 2794, 2821, 2827, 2842, 2866
- \CTEX@chapter@afterindent 1049
- \CTEX@chapter@aftername 1121
- \CTEX@chapter@afterskip 1128, 1147
- \CTEX@chapter@aftertitle 1124, 1143
- \CTEX@chapter@beforeskip 1108, 1133
- \CTEX@chapter@break 1045
- \CTEX@chapter@fixskip 1109, 1129, 1134, 1148
- \CTEX@chapter@format 1114, 1139
- \CTEX@chapter@lofskip 781
- \CTEX@chapter@lotskip 782
- \CTEX@chapter@numbering 1060, 1394
- \CTEX@chapter@pagestyle 780
- \CTEX@chapter@titleformat 1123, 1142
- \CTEX@chapter@tocline 1331
- \CTEX@chaptername 1121
- \CTEX@char@n 2484, 2574, 2609, 2640
- \CTEX@char@nn 94, 2493, 2565, 2574, 2598, 2614
- \CTEX@char@nn@n 94, 2598
- \CTEX@char@nnn 2509, 2574, 2599, 2615
- \CTEX@char@nnn@n 2598
- \CTEX@char@nnnn 2526, 2574, 2600, 2616
- \CTEX@char@nnnn@n 2598
- \CTEX@chardef@text@cmd 2804, 2854
- \CTEX@defaultfamilyhook 3923
- \CTEX@document@left@hook 4489, 4495
- \CTEX@document@right@hook 4491, 4497
- \CTEX@endallalchar
..... 2794, 2823, 2830, 2846, 2867
- \CTEX@Family@CMap 2350, 2387, 2392
- \CTEX@fixheadingskip 826, 831,
878, 946, 1003, 1027, 1129, 1148, 1171, 1261
- \CTEX@fixskip 1171, 1261, 1321, 1328
- \CTEX@fixtopskip 823, 905, 1109, 1134
- \CTEX@getttitle 926,
967, 990, 1011, 1084, 1099, 1197, 1235, 1441
- \CTEX@hang 1290, 1322, 1329
- \CTEX@heading@format@initial
..... 804, 931, 973, 993,
1014, 1113, 1138, 1202, 1240, 1737, 1756, 1775
- \CTEX@heading@glue 1276, 1307
- \CTEX@headinghang
..... 936, 978, 995, 1016, 1120, 1140, 1283
- \CTEX@headingskip ... 840, 879, 906, 947,
1004, 1028, 1110, 1130, 1135, 1149, 1172, 1262
- \CTEX@hyperheadinghook 1407, 1439
- \CTEX@ifnamefalse
..... 847, 917, 922, 958, 963, 988, 1009,
1068, 1074, 1080, 1097, 1183, 1192, 1234, 1553
- \CTEX@ifnametrue
..... 847, 913, 954, 1061, 1188, 1553
- \CTEX@indentbox
..... 821, 1221, 1250, 1739, 1758, 1777
- \CTEX@makeanchor 918, 923, 959,
964, 1069, 1193, 1397, 1405, 1418, 1424, 1430
- \CTEX@makeanchor@chapter 1075, 1081
- \CTEX@makeanchor@schapter 1098
- \CTEX@makeanchor@ssect 1184
- \CTEX@makeanchor@spart 989, 1010
- \CTEX@makeanchor@ssect 1175
- \CTEX@part@afterindent 880
- \CTEX@part@aftername 937, 979, 1741
- \CTEX@part@afterskip . 945, 1002, 1026, 1747
- \CTEX@part@aftertitle
..... 941, 982, 998, 1019, 1745
- \CTEX@part@beforeskip 877, 904, 1736
- \CTEX@part@break 875, 894
- \CTEX@part@fixskip . 878, 905, 946, 1003, 1027
- \CTEX@part@format . 932, 974, 994, 1015, 1738
- \CTEX@part@indent 1739
- \CTEX@part@numbering 912, 953, 1740
- \CTEX@part@pagestyle 779

\CTEX@part@titleformat	\CTEX@verbatim@font@hook
..... 939, 981, 997, 1018, 1745	 3798, 3802
\CTEX@part@tocline	\ctex@zhmap@endinput
..... 1331	 6173
\CTEX@partname	\CTEXdigits
..... 937, 979, 1741	 31
\CTEX@patch@text@cmd	\CTEXheadinglabel
..... 2832, 2858	 29, 59
\CTEX@patch@tunec	\CTEXheadingnumber
..... 2852	 29
\CTEX@popfilename@hook	\CTEXifheadingnumbering
..... 4522	 29
\CTEX@postappendix	\CTEXifname
..... 1379, 1387, 1393	 27, 29, 72
\CTEX@postchapter	\CTEXnumber
..... 1393	 31
\CTEX@postsection	\CTEXnumberline
..... 1387	 27
\CTEX@preappendix	\CTEXoptions
..... 1378, 1385, 1391	 39
\CTEX@prechapter	\ctexset 5, 6, 8, 9, 13, 16, 17, 29, 32, 33, 35, 39, 86
..... 1391	\CTEXsetup
\CTEX@presection	 39
..... 1385	\CTEXthe
\CTEX@reset@ptions@hook	 29, 59
..... 4516	\CTEXthechapter
\CTEX@rmfamilyhook ...	 15, 16, 28, 72
..... 121, 3900, 3920, 3929	\CTEXthesection
\CTEX@runin	 15, 19, 75
..... 1200, 1238, 1257, 1323, 1330	\CTEXunderdotbasesep
\CTEX@save@appendix	 39
..... 1380, 1383	\CTEXunderline
\CTEX@save@refstepcounter	 39
..... 1880, 1883	\curr@fontshape 2957, 2991, 3000, 3006, 3048
\CTEX@section@aftername	\declare@file@substitution
..... 1760	 132, 4435
\CTEX@section@afterskip	\DeclareAlternateKanjiFont
..... 1766	 115
\CTEX@section@aftertitle	\DeclareFontEncoding
..... 1764	 105
\CTEX@section@beforeskip	\DeclareFontFamily
..... 1755	 88, 103
\CTEX@section@format	\DeclareFontShape
..... 1757	 103
\CTEX@section@indent	\DeclareFontShape@
..... 1758	 115, 3617
\CTEX@section@numbering	\DeclareRobustCommand
..... 1388, 1759	 135
\CTEX@section@titleformat	\DeclareTextCommand
..... 1764	 99
\CTEX@sectionhang	\DeclareTextSymbol
..... 1206, 1244, 1283	 98
\CTEX@sectionname	\DeclareUnicodeAccent
..... 1760	 99
\CTEX@selectfont@hook	\DeclareUnicodeComposite
..... 4097, 4155, 4158	 99
\CTEX@selectfont@save 4111, 4126, 4130, 4133	\defbibheading
\CTEX@setcurrentlabel@n	 36
..... 1823, 1884	\define@newfont
\CTEX@setheadingskip .	 102, 2983
..... 838, 877, 904, 945,	\DefineBibliographyStrings
1002, 1026, 1108, 1128, 1133, 1147, 1170, 1260	 36
\CTEX@setthispagestyle	\disable@package@load
..... 841, 896, 1046	 131, 4422, 4424
\CTEX@sffamilyhook	\do@noligs
..... 3921, 3930	 3816
\CTEX@subsection@aftername	\do@subst@correction
..... 1779	 102, 103, 2982
\CTEX@subsection@afterskip	\em
..... 1785	 119
\CTEX@subsection@aftertitle	\endlinechar
..... 1783	 90
\CTEX@subsection@beforeskip	\escapechar
..... 1774	 105
\CTEX@subsection@format	\EverySelectfont
..... 1776	 125
\CTEX@subsection@indent	\expandafter
..... 1777	 35
\CTEX@subsection@numbering	\external@font
..... 1778	 2993, 2994, 2996
\CTEX@subsection@titleformat	\extract@font
..... 1783	 102, 2981
\CTEX@subsectionname	\extraschinese
..... 1779	 35
\CTEX@text@composite@x	\extrasenglish
..... 2819, 2856	 35
\CTEX@textcmd	\f@baselineskip
..... 2839, 2840, 2844, 2847	 494
\CTEX@thechapter	\f@encoding
..... 1392	 2954, 2999, 3103
\CTEX@thesection	\f@family
..... 1386	 102, 2955, 2999, 3009
\CTEX@titleformat@n 68, 1245, 1251, 1318, 1325	\f@nch@initialise
\CTEX@titleslabel@clear	 1674
..... 1569, 1579	\f@series
\CTEX@titleslabel@set ...	 2950, 3041
..... 1563, 1575, 1579	\f@shape
\CTEX@toc@width@n	 2950, 3041
..... 1605, 1620, 1632	\f@size
\CTEX@todayold	 1611,
..... 600, 605	2950, 2957, 3036, 3044, 3047, 3048, 4040, 4094
\CTEX@ttfamilyhook	\fangsong
..... 3922, 3931	 8
\CTEX@update@sectionformat@n ..	\fbox
..... 1159, 1316	 20
\CTEX@updatettlifname	\figurename
..... 1550, 1552	 14
	\font
	 103

\font@name	102, 103, 2455, 2457, 2458, 2960, 2967, 2968, 2973, 2978, 2996, 2997, 3016, 3084, 3085, 3086, 3087, 3088, 3095
\fontcharwd	125
\footnotesep	84, 85
\footnotesize	85
\foreignlanguage	36
\get@external@font	2990
\getanddefine@fonts	3073
\globaldefs	101
\globaljfont	102
\H@old@chapter	1412
\H@refstepcounter	1856
\hbox	24, 60
\heiti	8, 156
\highLight	38
\hrule	60
\Hy@chapapp	1069, 1075, 1081, 1098
\Hy@driver	3979
\Hy@MakeCurrentHrefAuto	1432
\Hy@org@chapter	1412
\Hy@raisedlink	1433
\hyper@anchorend	1436
\hyper@anchorstart	1435
\hypersetup	123
\if@compatibility	3805
\if@mainmatter	1058, 1116, 1660, 1710, 1911, 1912
\if@nobreak	1165
\if@noskipsec	873, 1152, 1268
\if@openright	889, 1031, 1043, 4825, 4853
\if@tempswa	1037
\if@twocolumn	897, 1090, 1100
\if@twoside	1030
\iffalse	105
\IfFormatAtLeastTF	131
\ifin@	105, 2305
\iftrue	105
\ifttl@label	1547, 1549
\ifttl@toclabel	1547, 1549
\in@	2304
\indexname	14
\input	131
\InputIfFileExists	131
\insertcontinuationtext	15
\insertsection	29
\itdefault	103
\itshape	96
\jfam	119
\kaishu	8
\kanjifamilydefault	122
\kanjiskip	127
\l@nohyphenation	3812
\labelformat	34, 35, 78, 79
\leftmark	15
\linewidth	53
\lishu	8
\listfigurename	14
\listtablename	14
\ltj@@does@alt@set	104
\ltj@@getjfontnumber	104, 3056
\ltj@@set@stackfont	3093, 3094, 3095
\ltj@allalchar	2800
\ltj@curjfont	103, 127, 3003
\ltj@pickup@altfont@auxy	101, 3042
\ltj@pickup@altfont@copy	101, 3054
\ltj@setpar@global	119, 3092, 3819, 3821
\ltj@tempcntc	104, 3057
\ltjalchar	99
\ltjlineendcomment	97
\ltjsetkanjiskip	119
\ltjsetparameter	119
\ltjsetxkanjiskip	119
\m@ne	911, 933, 1056, 1115, 1653, 1660, 1661, 1683, 1710, 1711, 6342, 6343, 6344, 6345, 6346, 6347
\MakeUppercase	92
\markboth	15
\markright	15
\math@bgroup	3102
\math@egroup	3107
\math@fonts	3090
\mathbb	85
\maxdimen	13, 53
\meaning	135-137
\mule@arg	2562
\newcommand	135
\newrobustcmd	135
\newtitlemark	73, 74
\nfss@catcodes	107
\noindent	59
\normalfont	122
\normalsize	10, 143
\nouppercase	90
\NR@getttitle	1445
\null	119
\numberline	74
\p@	976, 1087, 1088, 1107, 1117, 1127, 1132, 1146, 1611, 2096, 2097, 2098, 2104, 2105, 2106, 2108, 2109, 2115, 2116, 2117, 2119, 2120, 2135, 2136, 2137, 2143, 2144, 2145, 2147, 2148, 2154, 2155, 2156, 2158, 2159, 4040, 4094, 4821, 4849, 4850, 4851, 4852, 4859
\pagenumbering	31
\pagestyle	11
\par@update	126, 4117, 4121, 4125
\paragraph	11, 23, 44
\parindent	12, 39, 52, 54, 60, 68, 85
\parskip	60
\part	17, 23, 25, 26, 60, 67
\partmark	60
\partname	18

<code>\partpage</code>	17, 76	<code>\tf@size</code>	3084
<code>\PassOptionsToClass</code>	45	<code>\thechapter</code>	15, 28
<code>\PassOptionsToPackage</code>	51, 55	<code>\thesection</code>	15, 19, 28, 75
<code>\pdfmapline</code>	145	<code>\thispagestyle</code>	60
<code>\pickup@font</code>	102, 2956	<code>\titleformat</code>	71
<code>\pickup@jfont</code>	2987	<code>\titlespacing</code>	25, 71
<code>\pingfang</code>	8, 156	<code>\today</code>	13
<code>\printbibliography</code>	36	<code>\trans@languagepath</code>	5049, 5074
<code>\ProcessKeysOptions</code>	34, 45	<code>\ttfamily</code>	96, 154
<code>\proofname</code>	14, 56	<code>\ttl@a</code>	1527, 1535
<code>\protected</code>	73, 92	<code>\ttl@chapterout</code>	1522
<code>\protected@edef</code>	597, 599, 1190, 1825	<code>\ttl@extract</code>	71
<code>\ProvidesFile</code>	158	<code>\ttl@labelling</code>	1546
<code>\ps@headings</code>		<code>\ttl@setifthe</code>	1592
.....	75, 1641, 1694, 1697, 1702, 1708, 1719	<code>\ttl@setsubmark</code>	73, 1565, 1571
<code>\refname</code>	14, 36	<code>\ttl@settopmark</code>	73, 1559, 1590
<code>\refstepcounter</code>	35, 78	<code>\ttl@tocpart</code>	1524, 1526
<code>\refstepcounter@noarg</code>	1859	<code>\ttlh@hang</code>	1488, 1498
<code>\refstepcounter@optarg</code>	1860	<code>\ttlh@runin</code>	1487, 1497
<code>\relax</code>	53, 90	<code>\tw@</code>	2305
<code>\rightmark</code>	15	<code>\underline</code>	38
<code>\rmfamily</code>	96, 121	<code>\uppercase</code>	90
<code>\roman</code>	31	<code>\use@mathgroup</code>	3098, 3129
<code>\scantokens</code>	135	<code>\usepackage</code>	38, 40
<code>\scr@new@selectfont</code>	4114, 4129	<code>\verb</code>	119
<code>\scr@selectfont</code>	4126, 4130	<code>\verb@eol@error</code>	3810
<code>\scriptfont@name</code>	3087, 3094	<code>\verbatim@font</code>	3796, 3800, 3811
<code>\section</code> ..	11, 17–19, 23–26, 28, 40, 57, 67, 75	<code>\vskip</code>	24
<code>\section*</code>	17	<code>\wrong@fontshape</code>	102
<code>\sectionmark</code>	15	<code>\x@protect</code>	4606
<code>\sectionname</code>	18	<code>\xdef</code>	73
<code>\sectionpage</code>	17, 76	<code>\XeTeXglyphbounds</code>	162
<code>\selectfont</code>	53, 104, 121, 125–127	<code>\xkanjiskip</code>	127
<code>\selectlanguage</code>	36	<code>\yahei</code>	8, 156
<code>\setCJKfamilyfont</code>	32	<code>\youyuan</code>	8
<code>\setCJKmainfont</code>	7, 32, 97	<code>\z@</code>	930, 992, 1047, 1112, 1137, 1156, 1199, 1237, 1256, 1270, 1507, 1645, 1670, 1677, 1686, 1695, 1703, 1720, 2097, 2105, 2116, 2136, 2144, 2155, 2785, 2788, 3816
<code>\setcounter</code>	77	<code>\zhdig</code>	55
<code>\sf@size</code>	3086	<code>\zhdigits</code>	31
<code>\sffamily</code>	96	<code>\zhnum</code>	55
<code>\size@update</code>	53, 454, 468, 4141, 4142	<code>\zhnumber</code>	31
<code>\sldefault</code>	103	<code>\zihao</code>	30
<code>\songti</code>	8	<code>\ziju</code>	31
<code>\special</code>	7, 88, 123	<code>\zw</code>	125
<code>\split@name</code>	3047		
<code>\SplitArgument</code>	59	tex commands:	
<code>\ssf@size</code>	3088	<code>\tex_afterassignment:D</code>	2808
<code>\strikeThrough</code>	38	<code>\tex_chardef:D</code>	2809
<code>\strutbox</code>	85	<code>\tex_def:D</code>	4682
<code>\subparagraph</code>	11, 23, 44	<code>\tex_dimexpr:D</code>	105, 505
<code>\subsectionname</code>	18	<code>\tex_divide:D</code>	53, 506, 546
<code>\subsectionpage</code>	17, 77	<code>\tex_edef:D</code>	4736
<code>\subst@correction</code>	3012, 3019	<code>\tex_endlinechar:D</code>	2426
<code>\tablename</code>	14	<code>\tex_font:D</code>	2373, 2374, 6206, 6207
<code>\textfont@name</code>	3085, 3093	<code>\tex_glueexpr:D</code>	494
<code>\textangle</code>	90		
<code>\textriangle</code>	90		

- \tex_hangindent:D 1303
- \tex_iftrue:D 1912
- \tex_ignorespaces:D
 - 530, 1924, 3262, 3428, 3433, 4226, 4492
- \tex_indent:D 819
- \tex_interlinepenalty:D 808
- \tex_kanjiskip:D 4047, 4063, 4077
- \tex_noindent:D 809, 1301
- \tex_numexpr:D 2829
- \tex_pagegoal:D 827
- \tex_parindent:D 813, 818, 1293
- \tex_parskip:D 835
- \tex_pdffontattr:D 2374
- \tex_prevdepth:D 834
- \tex_setfontid:D 3003
- \tex_topskip:D 828
- \tex_Uchar:D 2817, 2828, 2829
- \tex_undefined:D 2417
- \tex_XeTeXcharglyph:D 6213
- \tex_XeTeXglyphbounds:D 6262
- \tex_xkanjiskip:D .. 4174, 4178, 4188, 4262
- text commands:
 - \l_text_case_exclude_arg_tl 2607
 - \text_declare_expand_equivalent:Nn ..
 - 2614, 2615, 2616
 - \text_declare_purify_equivalent:Nn ..
 - 2640, 2641, 2642, 2643
 - \text_expand:n 92, 94
 - \text_purify:n 94
 - \text_uppercase:n 92
 - \textasteriskcentered 2858
 - \the 5859, 5862, 5863, 5882, 5886
 - \thechapter 1063, 1066, 1117, 1662, 1683, 1712, 4856
 - \theparagraph 4958
 - \thepart 915, 934, 956, 976, 4795
 - \thesection 1645,
 - 1653, 1670, 1677, 1686, 1695, 1703, 1720, 4875
 - \thesubparagraph 4972
 - \thesubsection 1648, 1680, 1698, 4907
 - \thesubsubsection 4941
 - \thispagestyle 843, 895, 1033, 1044
 - \tiny 2125, 2164
- tl commands:
 - \c_novalue_tl 4471
 - \tl_clear:N
 - 189, 352, 445, 772, 790, 807, 2051,
 - 3750, 3754, 4015, 4027, 4167, 4690, 6224, 6225
 - \tl_clear_new:N 55, 329, 6244, 6245
 - \tl_const:Nn
 - 120, 464, 695, 701, 1894, 1913, 1915,
 - 2704, 3158, 3171, 3506, 3915, 3917, 3918,
 - 3919, 3926, 3927, 3928, 4427, 5750, 5753, 5754
 - \tl_gput_right:Nn
 - 2314, 2346, 2653, 3920, 3921,
 - 3922, 3923, 3929, 3930, 3931, 4486, 4488, 4506
 - \tl_gset:Nn 218, 221, 224, 377,
 - 2211, 2214, 2215, 2231, 2264, 2438, 2441,
 - 2968, 3084, 3086, 3088, 3480, 3932, 3943, 5378
 - \tl_gset_eq:NN 2973, 3964
 - \tl_gset_rescan:Nnn 3487
 - \tl_head:N 2055
 - \tl_head:n 3531
 - \tl_if_blank:nTF
 - . 417, 427, 2828, 3565, 3574, 3774, 3775, 3790
 - \tl_if_blank_p:n 6240, 6241
 - \tl_if_empty:NnTF 373, 523, 707, 1840, 2222,
 - 2945, 5412, 5599, 5618, 5632, 5640, 5650, 5655
 - \tl_if_empty:nTF 109, 3850, 3864, 3877, 4653
 - \tl_if_eq:NNTF 457, 3939
 - \tl_if_eq:NnTF 5061
 - \tl_if_exist:NnTF
 - . 72, 622, 653, 664, 667, 1906, 2652, 3914,
 - 3917, 3918, 3919, 3926, 3927, 3928, 4505, 5084
 - \tl_if_in:NnTF 4717
 - \tl_if_in:nTF 5382
 - \tl_if_novalue:nTF ... 771, 3389, 3412, 3768
 - \tl_map_inline:Nn 3680
 - \tl_map_inline:nn 2052
 - \tl_new:N ... 110, 112, 465, 478, 624, 693,
 - 694, 1378, 1379, 2335, 2380, 2652, 2811,
 - 2947, 2948, 3039, 3156, 3206, 3233, 3250,
 - 3251, 3272, 3622, 3756, 3907, 3957, 4181,
 - 4399, 4499, 4500, 4505, 4693, 4694, 4695, 5360
 - \tl_put_left:Nn 4132, 4522
 - \tl_put_right:Nn 336, 723, 761, 763, 2055,
 - 2056, 2607, 3090, 3763, 4120, 4121, 4125, 4516
 - \tl_remove_all:Nn 3532
 - \tl_replace_all:Nnn 3578
 - \tl_rescan:nn 137, 4735, 4744
 - \tl_set:Nn 100,
 - 184, 441, 479, 533, 625, 770, 773, 777, 1484,
 - 1503, 1522, 1527, 1535, 2183, 2409, 2410,
 - 2411, 2807, 2949, 2993, 3040, 3140, 3192,
 - 3255, 3267, 3411, 3575, 3577, 3618, 3959,
 - 4110, 4166, 4182, 4240, 4259, 5375, 6229, 6231
 - \tl_set_eq:NN 466, 2954,
 - 2955, 3009, 3085, 3087, 3256, 3281, 3718, 3730
 - \tl_set_rescan:Nnn 4691
 - \tl_tail:N 1489
 - \tl_tail:n 1499
 - \tl_to_str:n
 - 108, 2544, 2551, 2558, 4634, 4663, 4704, 4705
 - \tl_trim_spaces:n 3577
 - \tl_trim_spaces_apply:nN 3747
 - \tl_use:N 2316
 - \l_tmpa_tl 5375, 5377, 5378
- tocdepth 27, 1792
- \today 600, 605, 608, 613
- today 13, 600
- token commands:
 - \c_parameter_token 74, 1623, 1625, 1628

- \token_if_eq_meaning:NNTF 2487, 2501, 2518
 - \token_to_meaning:N 4648
 - \token_to_str:N
 - 105, 144, 650, 651, 1932, 2737,
 - 2738, 2739, 2835, 3067, 3075, 3084, 3086,
 - 3088, 3309, 3310, 3311, 3313, 3320, 3321,
 - 3322, 3384, 3415, 4366, 4445, 4460, 4552, 6217
 - \toks 5859, 5860, 5862, 5863, 5882, 5886
 - \topsep 2108, 2119, 2147, 2158
 - \TrimSpaces 415
 - \ttdefault 3949
 - \ttfamily 3912
 - \twocolumn 1038
 - \typeout 1063, 1064, 1522
- U**
- ubuntu 40
 - um commands:
 - \um_input_math_symbol_table: .. 3122, 3126
 - \um_sym:nnn 3125
 - um internal commands:
 - _um_input_math_symbol_table: 3134
 - _um_switchto_literal: 3131
 - _um_sym:nnn 3133
 - \undefined 5884
 - \underline 2737
 - \UnicodeEncodingName 2835
 - \unless 1200, 1238, 1257
 - \unskip 1273
 - \upshape 3840, 3843
 - use commands:
 - \use:N
 - . 799, 801, 843, 855, 861, 870, 1287, 1350,
 - 2577, 2583, 2589, 2595, 2999, 3000, 3084,
 - 3086, 3088, 3176, 3225, 3731, 4718, 4738, 4752
 - \use:n .. 113, 1296, 1325, 1616, 1904, 2236,
 - 2383, 2463, 2540, 2547, 2554, 2637, 2638,
 - 2639, 3047, 3160, 3339, 3392, 3398, 3477,
 - 3481, 3580, 3958, 4465, 4587, 4629, 4680, 4701
 - \use:nn 4746
 - \use_i:nn 97, 849, 4533, 4538, 4558, 4563, 4573
 - \use_i:nnn 239
 - \use_ii:nn
 - ... 98, 847, 851, 4533, 4538, 4558, 4563, 4574
 - \use_ii:nnn 114, 244
 - \use_iii:nnn 249
 - \use_none:n 185,
 - 190, 195, 586, 2391, 3026, 3241, 3480, 3615
 - \use_none:nn 2392
 - \use_none_delimit_by_q_stop:w ... 149,
 - 151, 5388, 5408, 5428, 5470, 5485, 5545, 5665
- \usebeamerfont 1733, 1744, 1752, 1763,
- 1771, 1782, 4837, 4840, 4897, 4900, 4931, 4934
- \usefont 3913
- \uselanguage 5040, 5045, 5065, 5070
- \usepackage 4366
- UTF8 6, 40, 215
- V**
- \vadjust 3808, 3816
 - \verb 3806
 - \verse 5089
 - \vfil 903, 1025
 - \vskip 944, 947, 976, 1001, 1004, 1028,
 - 1117, 1127, 1130, 1146, 1149, 1259, 1262,
 - 1735, 1754, 1773, 4821, 4839, 4859, 4899, 4933
 - \vspace 906, 1107, 1110, 1132, 1135
- W**
- windows 40
 - windowsnew 40
 - windowsold 40
 - winfonts 38
 - \write 5899
- X**
- \x 2298, 2304,
 - 2308, 5861, 5864, 5865, 5866, 5867, 5868,
 - 5869, 5870, 5871, 5872, 5873, 5874, 5875,
 - 5876, 5877, 5878, 5879, 5880, 5887, 5897, 5901
 - x commands:
 - \x: 2303
 - \xdef 5900
 - \xeCJKsetup 2710, 4023, 4221, 4224, 4227, 4242, 4254
- Y**
- \yahei 5821
 - \youyuan 5821
- Z**
- \zhdig 592
 - \zhdigits 31, 599
 - zhmap 7, 38, 234
 - \zhnum 591
 - zhnum commands:
 - \zhnum_counter:n 583
 - \zhnumber 31, 597
 - \zhnumsetup 609, 614
 - \zhtoday 608, 613
 - \zihao 30, 1923, 1932
 - zihao 10, 164
 - \ziju 31, 529
 - \zw 2786, 4044, 4095, 4183

版本历史

v2.0	(2014/03/06 – 2015/05/06)	将章节标题设置功能提取到可以独立使用的宏包 <code>ctexheading</code> 中。..... 1
General: <code>c5size</code> , <code>cs4size</code> 是过时选项。..... 46		新的标题格式选项 <code>aftertitle</code> 。..... 58
<code>captiondelimiter</code> 是过时选项。..... 85		修复 <code>ctexbook</code> 和 <code>ctexrep</code> 类的中文 <code>part/number</code> 选项初值为空的错误。..... 139
<code>fancyhdr</code> 成为过时选项, 原选项功能总是打开。..... 49		<code>\ctex_if_platform_macos:TF</code> : 改用 <code>/Library/Fonts/Songti.ttc</code> 为特征文件。..... 86
<code>fntef</code> 成为过时选项, 原选项功能总是打开。..... 49		<code>hyperref</code> : 补充定义 <code>\hypersetup</code> 。..... 49
<code>hyperref</code> 成为过时选项, 原选项功能总是打开。..... 49		v2.2
<code>indent</code> , <code>noindent</code> 是过时选项。..... 47		(2015/06/21 – 2015/06/30)
<code>nofonts</code> , <code>adobefonts</code> , <code>winfonts</code> 是过时选项。..... 48		General: <code>before skip</code> 和 <code>after skip</code> 选项的符号不再有特殊意义。..... 58
<code>nopunct</code> 是过时选项。..... 48		<code>before skip</code> , <code>after skip</code> 和 <code>indent</code> 选项支持表达式。.. 61
<code>nospace</code> 是过时选项。..... 48		不再依赖 <code>etoolbox</code> 宏包。..... 1
<code>nozhmap</code> 是过时选项。..... 48		非 <code>ctexart</code> 类的 <code>part/before skip</code> 和 <code>part/after skip</code> 选项有意义。..... 61
<code>punct</code> 选项可以设置标点格式。..... 48		给 <code>enumitem</code> 宏包注册 <code>\chinese</code> 和 <code>\zhnum</code> 。..... 55
<code>ctex</code> 宏包新增 <code>heading</code> 选项。..... 48		将文档开头和宏包末尾钩子提取到 <code>ctexhook</code> 宏包中。.. 131
<code>\CTEXindent</code> , <code>\CTEXnoindent</code> 是过时命令。..... 55		将中文版式下的 <code>part</code> 和 <code>chapter</code> 标题的 <code>nameformat</code> 和 <code>titleformat</code> 选项的初值合并到 <code>format</code> 中。..... 139
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。..... 16, 52		删去 <code>etoolbox</code> 与 <code>breqn</code> 的兼容补丁。..... 85
<code>\CTEXunderdot</code> , <code>\CTEXunderline</code> , <code>\CTEXunderdblline</code> , <code>\CTEXunderwave</code> , <code>\CTEXsout</code> , <code>\CTEXxout</code> 是过时命令; <code>CTEXfilltwosides</code> 是过时环境。..... 124		新的标题格式选项 <code>afterindent</code> 。..... 58
标题设置新增 <code>pagestyle</code> 选项。..... 59		新的标题格式选项 <code>numbering</code> 。..... 58
调整 <code>\footnotesep</code> 的大小, 以适合行距的变化。..... 84		新的标题格式选项 <code>runin</code> 。..... 58
兼容 <code>extsizes</code> 宏包、 <code>beamer</code> 、 <code>memoir</code> 等提供的更多字号选项。..... 50		新增子宏包 <code>ctexpatch</code> 实现给宏打补丁功能。..... 134
将标题汉化功能加入 <code>ctex.sty</code> 。..... 56		v2.3
将中文字号功能提取到可以独立使用的 <code>ctexsize</code> 。..... 80		(2015/09/17 – 2016/01/05)
解决 <code>etoolbox</code> 与 <code>breqn</code> 关于 <code>\end</code> 的冲突。..... 85		General: <code>.value_required</code> : 和 <code>.value_forbidden</code> : 已过时。..... 46
默认关闭 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 的彩色设置。..... 124		代码实现避免使用 <code>\lowercase</code> 技巧 (Joseph Wright)。.. 45
删除 <code>c19gsn.fd</code> 和 <code>c19gkai.fd</code> 。..... 1		更新 <code>LuaTeX-j</code> 支持 (20150922.0)。..... 96
通过 <code>LuaTeX-j</code> 宏包支持 <code>Lua$\TeX$</code> 。..... 96		更新 <code>unicode-math</code> 宏包补丁。..... 99
新增 <code>autoindent</code> 选项。..... 47		兼容 <code>titles</code> 宏包。..... 73
新增 <code>fontset</code> 选项。..... 48		修复 <code>nameformat</code> 作用域问题。..... 57
新增 <code>linespread</code> 选项。..... 47		与 <code>LaTeX3</code> (2015/12/20) 同步。..... 43
新增 <code>linestretch</code> 选项。..... 53		v2.4
新增 <code>scheme</code> 选项, 并将 <code>cap</code> 和 <code>nocap</code> 列为过时选项。.. 49		(2015/02/19 – 2016/04/25)
新增 <code>zhmCJK</code> 支持选项。..... 48		General: 初步支持 <code>up$\TeX$</code> 。..... 119
新增 <code>zihao</code> 选项。..... 46		加强 <code>beamer</code> 宏包支持。..... 1
新增统一设置接口 <code>\ctexset</code> 。..... 52		提供 <code>up$\TeX$</code> 的 NFSS 字体定义。..... 166
应用 <code>LaTeX3</code> 重新整理代码。..... 1		提供 <code>translator</code> 宏包的中文定理名称翻译。..... 129
中文字号不再采用近似值。..... 81		正确更新 <code>CJK</code> 包的 <code>\CJKfamilydefault</code> 。..... 122
自动检测操作系统, 载入对应的字体配置。..... 123		正确设置 <code>up$\TeX$</code> 下字体命令。..... 120
<code>\CJK@surr</code> : 解决与 <code>\nouppercase</code> 的冲突。..... 90		<code>\ctex_parse_name:NN</code> : 修复宏名解析错误。..... 135
v2.0.1	(2015/05/15)	v2.4.1
General: 修复 10pt、11pt 等选项无效的问题。..... 50		(2016/04/26 – 2016/05/14)
v2.0.2	(2015/05/16)	General: <code>beamer</code> 不调整默认行距。..... 144
General: 修复加载 <code>ctex</code> 宏包后章节标题后第一段无段首缩进的问题。..... 144		<code>beamer</code> 不调整默认字体大小。..... 143
v2.1	(2015/05/18 – 2015/06/19)	使用 <code>bootfont.bin</code> 判断 Windows XP 以避免权限问题。..... 155
General: <code>format+</code> , <code>nameformat+</code> 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。..... 58		随字体更新 <code>up$\TeX$</code> 的 <code>\xkanjiskip</code> 。..... 127
<code>nameformat</code> 可以接受章节名字为参数。..... 57		新的标题格式选项 <code>part/fixbefore skip</code> 和 <code>chapter/fixbefore skip</code> 。..... 59
不依赖 <code>ifpdf</code> 宏包。..... 45		正确更新 <code>up$\TeX$</code> 的 <code>\CJKfamilydefault</code> 。..... 122
不再设置 <code>hyperref</code> 宏包的 <code>colorlinks</code> 选项。..... 51		<code>\ccwd</code> : 正确设置 <code>up$\TeX$</code> 下的 <code>\ccwd</code> 。..... 124
给 <code>pdf$\TeX$</code> 下的非 UTF-8 编码 <code>CJK</code> 字体族加上 <code>CMap</code> 。88		<code>\chinese</code> : 支持 <code>\pagenumbering</code> 。..... 55
		<code>zihao</code> : 不允许无参 <code>zihao</code> 选项。..... 47

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。	100	<code>\CTEX@selectfont@hook</code> : 修正导言区 <code>\selectfont</code> 钩子位置。	125
<code>\em</code> : 兼容 <code>upL^AT_EX</code> 2016/05/07u00 的定义。	119	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。	150
General: 更新 <code>unicode-math</code> 补丁。	106	区分 <code>macold</code> 及 <code>macnew</code> 。	149
简化 <code>fontspec</code> 补丁。	100	为 <code>macnew</code> 配置字体命令。	156
确保 <code>\proofname</code> 非空。	56	v2.4.15	(2019/01/29 – 2019/04/05)
删除选项 <code>part/fixbefore skip</code> 和 <code>chapter/fixbefore skip</code> 。	59	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。	61
新的标题格式选项 <code>fixskip</code> 。	58	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> 。	119
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。	165
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	96	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。	76
改进 <code>hyperref</code> 宏包的标题锚点设置。	69	同步 <code>L^AT_EX3</code> 2019/03/05。	1
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	90	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。	147
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	72	显式补丁 <code>upL^AT_EX</code> 的 <code>\rmfamily</code> 等字体命令。	119
提供 <code>\CTEXifname</code> 。	60	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。	61
提供 <code>\partmark</code> 。	60	v2.4.16	(2019/05/11 – 2019/05/29)
新的标题格式选项 <code>break</code> 。	58	General: 更好地兼容 <code>nameref</code> 宏包。	70
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> 。	61
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	59	允许设置 <code>autoindent</code> 为 0。	45
新的标题格式选项 <code>hang</code> 。	58	v2.5	(2019/10/25 – 2020/04/30)
新的标题格式选项 <code>tocline</code> 。	58	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。	154
<code>\ps@headings</code> : 修复补丁失败。	75	标题选项 <code>format</code> 也可以接受参数。	61
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。	155
General: <code>\CTEXifname</code> 初始为假。	60	不再自动载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。	124
支持字体属性可选项在后的新语法。	111	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。	123
重新初始化 <code>\ifthechapter</code> 等。	74	处理 <code>\ctex_file_input:n</code> 在 <code>ctexsize</code> 中未定义的错误。	43
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> 。	87
General: 依赖 <code>pxeverysel</code> 宏包。	121	更新 <code>LuaT_EX-j_a</code> 支持(20200412.0)。	96
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	75	兼容 <code>titletoc</code> 宏包。	74
v2.4.8	(2017/02/23)	删除 <code>fontspec</code> 补丁。	100
General: 解决与 <code>fontspec</code> 2017/01/24 v2.5d 的字体族匹配兼容问题。	112	使用环境变量代替绝对路径查找字体。	155
v2.4.9	(2017/02/27)	所有引擎下默认编码均设为 UTF-8。	47
General: 调整 <code>unicode-math</code> 补丁的代码顺序。	106	为 <code>macnew</code> 增加粗楷体、隶书和圆体的定义。	150
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。	124
General: 常数 <code>\c_minus_one</code> 已过时。	43	应用新内核中的 <code>\labelformat</code> 。	79
定义 <code>\cht</code> 、 <code>\cdp</code> 和 <code>\cwo</code> 。	127	允许 <code>macnew</code> 在 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下使用。	150
使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 (<code>L^AT_EX3</code> 2017/07/19)。	43	在 <code>ctexsize</code> 也载入 <code>fix-cm</code> 。	43
v2.4.11	(2017/08/17 – 2017/11/21)	增加宏包开头钩子。	131
General: 不把 Enclosed Alphanumerics 设置为 <code>JAchar</code> 。	97	增加字体映射文件 <code>zhmacfonts.tex</code> 。	160
不把希腊和西里尔字母设置为 <code>JAchar</code> 。	98	重构标题选项 <code>indent</code> 和 <code>hang</code> 。	59
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。	58	重构字体选项 <code>AlternateFont</code> 。	113
<code>\ps@headings</code> : 补充页眉空格。	75	<code>\CTEX@selectfont@hook</code> : 兼容 KOMA-Script 的 <code>\selectfont</code> 补丁。	125
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。	86
General: 同步 <code>L^AT_EX3</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。	49
修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。	52		
正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	119		
<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	85		

v2.5.1	(2020/05/02)	Breaking: 移除对 L ^A T _E X 2020/10/01 之前版本的字体钩子兼容代码。现在需要 L ^A T _E X 2020/10/01 或更新版本 (#746)。 121
General: zhconv 更名为 ctex-zhconv。 1		4 字节 UTF-8 字符也优先使用
v2.5.2	(2020/05/05 – 2020/05/06)	\DeclareUnicodeCharacter 的定义 (#815)。 92
General: ctexmakespa.tex 更名为 ctex-spa-make.tex。 162		补全 upL ^A T _E X 字体编码 JY2 和 JT2 的 Fallback 机制。 . . 166
ctexspamacro.tex 更名为 ctex-spa-macro.tex。 . . 162		将 experiment/font-size-system 的 traditional 选项更名为 letterpress (#813)。 46
zhadobefonts.tex 等字体映射文件更名为		使用 \l_keys_key_str 和 \l_keys_choice_str 替代已废弃的 \t_l 版本 (#806)。 46
ctex-zhmap-*.tex。 157		提升 L ^A T _E X3 版本至 2022/10/09。 43
兼容 L ^A T _E X 2020-02-02 之前的版本。 121		提升 L ^A T _E X3 最低版本要求至 2025/10/09。 43
新增标题选项 secnumdepth 和 tocdepth。 77		文档: 在标准字体命令、中文字号表附近提示 experiment/font-size-system 的影响; 说明 letterpress 只是金属活字排印字号体系之一 (#871)。 46
修正 macnew 和 ubuntu 字库的 CJKpunct 标点信息。 . 149		文档说明 runin 与 aftertitle 的交互 (#574)。 23
v2.5.3	(2020/05/31 – 2020/06/06)	文档字体统一为 Noto CJK 系列 (#686)。 1
General: 不再依赖 xunicode, 单独补丁 tuenc.def。 98		新增实验性 experiment/CJKecglue 选项 (#717)。 . . . 128
兼容 cleveref。 78		新增实验性 experiment/font-size-system 选项 (#543)。 46
正确关闭和恢复 L ^A T _E X3 语法环境。 44		修复 hyperref 在 ctex 之前被加载时 driverfallback 选项重复设置的警告 (#715)。 123
v2.5.4	(2020/06/07 – 2020/08/16)	修复参数签名, @@_char_auxi:NNNN 原误为 NNN。 92
General: 更新 LuaTeX-ja 支持 (20200808.0)。 96		修复了 T _E X tree 字体无法制作 .spa 文件的问题。 162
兼容 L ^A T _E X 2020/10/01 的钩子机制。 131		修复内联 \verb 前 xkanjiskip 丢失的问题 (移植 lltjcore 对 \verb 和 \do@noligs 的修复)。 119
同时兼容 cleveref 和 hyperref。 78		\ctex_undecore_unicode_character:n: 替换废弃命令 \char_to_utfviii_bytes:n。 90
修正主要字体命令补丁。 121		v2.6.1
应用 l3cctab。 44		(2026/07/01)
v2.5.5	(2020/10/06 – 2020/10/17)	General: 补全字库选项 windows 的 pdfT _E X 分支中, \ctex_zhmap_case:nnn 的第三个子分支 zhmap = false 缺失的报错。 155
General: 放弃应用 l3cctab。 44		v2.6.2
进一步应用 L ^A T _E X 2020/10/01 的新钩子。 121		(2026/07/02 – 2026/07/05)
\CJKhook: 不再通过旧的钩子命令来定义。 95		General: ubuntu 字库新增仿宋配置和 \fangsong 命令。使用 X ₃ L ^A T _E X 或 LuaL ^A T _E X 编译时, 按朱雀仿宋、FandolFang、思源宋体的顺序自动选用 (#908)。 154
v2.5.6	(2021/01/11 – 2021/02/16)	订正 ctex-heading-*.def 中 \ProvidesExplFile 使用与实际文件名不匹配的 ctex-*.def。 43
General: 使用正确的导言区末尾钩子。 132		将单一 ctex.dtx 拆分为 ctex.dtx (安装脚本与用户文档) 与 ctex-kernel.dtx、ctex-engine.dtx、ctex-scheme.dtx、ctex-auxpkg.dtx、ctex-fontset.dtx 六个源文件, 由夏明宇完成重构 (#937)。 1
\f@nch@initialise: 更新 fancyhdr 宏包的补丁。 75		文档等宽字体改用朱雀仿宋, 意大利体改用霞鹜文楷 GB Lite (#908)。 1
v2.5.7	(2021/06/04 – 2021/06/12)	文档字体启用 Language 与 PoZheHaoLigature, 破折号按全角字形合字输出 (#382)。 1
General: 更好地兼容 cmap 包。 89		v2.6.3
兼容 L ^A T _E X 2021/06/01 的字体钩子。 43		(2026/07/13 – 2026/07/15)
禁用 DVIPDFMx 驱动的 unicode 书签设置。 123		General: 按宏包整理第三方兼容说明, 并增加与 babel、biblatex 联用的内容 (#986, #987)。 35
使用 \disable@package@load 禁止宏包载入。 131		禁止在 LuaTeX 后端载入不兼容的 CJKfntef, 并推荐使用 lua-ul (#381)。 96
同时兼容 cleveref 和 beamer。 78		说明保留零段首缩进的兼容行为 (#402)。 12
重新应用 l3cctab。 131		提供按层级查询标题编号、完整标签和编号开关的接口, 使 Beamer 主题作者可以通过公开接口读取并响应 CTeX 本地化编号设置, 而无需访问私有变量 (#275)。 . . 59
\CJK@addcmap: 应用 \pdfnombuiltintounicode。 91		
\ctex_add_cmap:n: 确保 cmap 文件存在。 89		
v2.5.8	(2021/11/18 – 2021/12/04)	
General: 兼容 L ^A T _E X 2021/11/15。 132		
兼容 microtype。 102		
兼容 titlesec 包和 \CTEXifname。 72		
简化部分 Lua 函数。 100		
v2.5.9	(2022/05/26 – 2022/05/27)	
General: 设置消息模块的名字和类型。 43		
依赖 chinese-jfm 宏包。 106		
v2.5.10	(2022/06/10 – 2022/07/11)	
General: 不直接依赖 xparse 和 l3keys2e。 43		
更新一些内部函数。 43		
解决 CJK 包与 \text_uppercase:n 等转化函数的冲突。 92		
取消 L ^A T _E X 2022-06-01 对书名号的定义。 90		
展开传递 pagestyle 的值。 60		
\ctex_add_cmap:n: 使用封装好的函数。 89		
v2.6.0	(2024/04/20 – 2026/06/23)	
General: macnew 增加 macOS 15+ 兼容, 字体运行时检测。 149		
Breaking: \newCJKfontfamily 定义的字体切换命令改为局部定义, 与 \newcommand 行为一致。在分组内调用时, 命令不再泄漏到分组外 (#751)。 111		