

CT_EX 宏集手册

CTEX.ORG

2020/08/23 v2.5.4*

简介

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

CT_EX 宏集支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX、LuaL^AT_EX、upL^AT_EX 等多种不同的编译方式,并为它们提供了统一的界面。主要功能由宏包 `ctex` 以及中文文档类 `ctexart`、`ctexrep`、`ctexbook` 和 `ctexbeamer` 实现。

目录

第 1 节 介绍	2	第 7 节 章节标题样式设置	15
第 2 节 简明教程	3	7.1 编号相关	16
2.1 CT _E X 宏集的组成	3	7.2 格式相关	17
2.2 CT _E X 宏集的安装和更新	3	7.3 间距、缩进相关	22
2.3 使用 CT _E X 文档类	4	7.4 目录、附录相关	25
2.4 使用 <code>ctex</code> 宏包	5	7.5 辅助命令	26
第 3 节 宏包选项与 <code>\ctexset</code> 命令	5	7.6 示例	27
第 4 节 编译方式、编码与中文字库	6	第 8 节 实用命令	27
4.1 编译方式	6	8.1 字号与间距	27
4.2 中文编码	6	8.2 中文数字转换	27
4.3 中文字库	6	8.3 杂项	29
第 5 节 排版格式设定	8	第 9 节 LuaL ^A T _E X 下的中文支持方式	29
5.1 文档默认字号	8	9.1 LuaL ^A T _E X 下替代字体的设置	30
5.2 章节标题风格	9	第 10 节 CT _E X 宏集的配置文	31
5.3 排版方案选项	10	10.1 修改宏包默认选项	31
第 6 节 文档汉化	11	10.2 宏包载入后的配置	32
6.1 日期汉化	11	10.3 配置标题中文翻译	32
6.2 文档标题汉化	12	10.4 自定义字体集	32
6.3 页面格式设置与汉化	13	第 11 节 对旧版本的兼容性	32
		11.1 CT _E X 0.8a 及以前的版本	32
		11.2 CT _E X 0.9–CT _E X 1.0d	33
		11.3 CT _E X 1.02c 以后的 SVN 开发版	34
		11.4 CT _E X 2.2 之前的版本	35
		11.5 CT _E X 2.4.1 和 2.4.2	35

*`ctex-kit` rev. e805781.

11.6 C _T E _X 2.5 之前的版本	35	第 14 节 代码实现	38
第 12 节 宏集依赖情况与手工安装方法	36	版本历史	146
第 13 节 开发人员	37	代码索引	148

第 1 节 介绍

历史

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

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

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 [GitHub](#)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_AT_EX 对中文的支持一直不是很好。即使在 CJK 宏包解决了中文字符处理的问题以后，中文用户使用 L_AT_EX 仍然要面对许多困难。这些困难里，以章节标题的中文化为最。由于中文和西文书写习惯的差异，用户很难使用标准文档类中的代码结构来表达中文标题。于是，用户不得不对标准文档类做较大的修改。除此之外，日期格式、首行缩进、中文字号和字距等细节问题，也需要精细的调校。我们设计 C_TE_X 宏集的目的之一就是解决这些 L_AT_EX 文档的汉化难题。

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

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

关于宏集名字的说明

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

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

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

³比如：pdf_TE_X 引擎下的 CJK、zhmCJK 宏包，X_YL_AT_EX 引擎下的 xeCJK 宏包和 Lua_TE_X 引擎下的 Lua_TE_X-ja 宏包。

⁴2018 年，C_TE_X 论坛因故无限期关闭，此链接现已失效。

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

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

第 2 节 简明教程

2.1 CT_EX 宏集的组成

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

表 1 CT_EX 宏集的组成

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

2.2 CT_EX 宏集的安装和更新

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

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

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 CT_EX 宏集⁷。

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

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

⁷*nix 用户可能需要超级用户权限(sudo)才能正确安装宏集。

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

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

来安装 CT_EX 宏集。

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

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

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

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

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

来进行更新。

2.3 使用 CT_EX 文档类

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

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

例 1

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

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

以下是使用 ctexbeamer 文档类编写中文演示文稿的一个示例。

例 2

```
\documentclass{ctexbeamer}
\begin{document}
\begin{frame}{中文演示文档}
\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
\end{itemize}
\end{frame}
\end{document}
```

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

```

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

```

2.4 使用 `ctex` 宏包

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

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

例 3

```

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

```

第 3 节 宏包选项与 `\ctexset` 命令

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

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

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

New: 2014-03-18

`\ctexset {<键值列表>}`

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

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

例 4

```

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

```

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

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

表 2 \CTeX 宏集的中文支持方式

编译方式	(pdf) \LaTeX	\XeLaTeX	\LuaLaTeX	\upLaTeX [*]
支持宏包	CJK	xeCJK	LuaTeX-ja	原生

^{*} \pLaTeX-ng (或称 \ApLaTeX) 与 \upLaTeX 兼容。使用 \pLaTeX-ng 编译时, `ctex` 采用与 \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 \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。 <code>\CTeX</code> 宏集无法检测文档源文件的实际编码格式, 因此需要用户通过选项声明。如果没有显式指定, 则默认采用 UTF-8 编码。
UTF8	☆	
Updated: 2019-11-10		

使用 `XLaTeX`、`LuaLaTeX` 或 `upLaTeX` 编译时, `\CTeX` 宏集强制使用 UTF-8 编码, 此时 GBK

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

用户需要保证编译方式、源文件编码、宏包编码选项三者一致。

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

4.3 中文字库

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

zhmap	☆	<code>zhmap = <true false zhmCJK></code>
Updated: 2014-03-08		
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 <code>zhmetrics</code> 宏包, 将 CJK 字库通过 <code>\special</code> 命令映射到 .ttf 文件。

⁹ \LaTeX 、pdf \LaTeX 、 \XeLaTeX 、 \LuaLaTeX 及 \upLaTeX 。
¹⁰ \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。
¹¹ 对于以键值对形式提供的选项, 文档以**粗体**来表示其默认值。下同, 不再额外解释。

表 3 CT_EX 宏集自动配置字体策略

	macOS Old ¹	macOS New ²	Windows ³	其他
X _Y 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 + DVIPDFM _x	不可用	CJK + zhmetrics 华文字库 + 苹方	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	CJK + zhmetrics Fandol 字库
upLaTeX + DVIPDFM _x	不可用	zhmetrics-uptex 华文字库 + 苹方	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex Fandol 字库

1 Yosemite (10.10) 及以前的 macOS 系统。

2 El Capitan (10.11) 及以后的 macOS 系统。

3 仅支持 Windows Vista 及以后的 Windows 操作系统。

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

5 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。

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 = {adobe|fandol|founder|mac|macnew|macold|ubuntu|windows|none|...}
```

New: 2014-03-08

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 预定义了以下六种中文字库。

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

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

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

¹²使用 (pdf)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](#)。

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

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

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

例 5

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

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

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

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

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

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

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

在 `macnew` 字库中, 还定义了 `\pingfang`:

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

第 5 节 排版格式设定

5.1 文档默认字号

`zihao` ☆ `zihao = <-4|5|false>`

New: 2015-05-06

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

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

10pt ☆
11pt ☆
12pt ☆

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

表 4 标准字体命令与字号的对应

字体命令	zihao = 5		zihao = -4		10pt	11pt	12pt
	字号	bp	字号	bp	pt	pt	pt
<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

标准文档类的其他选项在 $\text{CT}_{\text{E}}\text{X}$ 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。 $\text{CT}_{\text{E}}\text{X}$ 会将这些选项传给标准文档类¹⁴。

5.2 章节标题风格

`heading` ☆ `heading = (true|false)`

New: 2014-03-08

本选项只能在调用 `ctex.sty` 时作为宏包选项使用。

$\text{CT}_{\text{E}}\text{X}$ 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。

$\text{CT}_{\text{E}}\text{X}$ 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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 标准文档类。若然，则该选项将会使得 `ctex.sty` 宏包的行为和 $\text{CT}_{\text{E}}\text{X}$ 宏集提供的四个中文文档类完全一致；若不然，则会根据 `\chapter` 是否有定义来使用 `ctexbook` 或者 `ctexart` 的标题设置。

`sub3section` ☆
`sub4section` ☆

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

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

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

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

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

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

5.3 排版方案选项

scheme ☆ `scheme = <chinese|plain>`

New: 2015-04-15

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

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

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

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

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

Updated: 2014-04-11

设置标点处理格式。预定义好的格式有:

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

banjiao 半角式: 所有标点占半个汉字宽度;

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

CCT CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

plain 原样 (不调整标点间距)。

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

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。该选项仅在使用 `X3LaTeX/(pdf)LaTeX` 编译时有效。

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

false 使用 `(pdf)LaTeX` 编译时: 总是忽略掉汉字后面的空格, 不论汉字后是什么; 使用 `X3LaTeX` 编译时, 等同于 `auto` 的效果。不建议使用该选项。

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

例如, 使用

例 6

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

将得到“汉字 分词 技术 English”; 使用

例 7

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

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

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

¹⁷`LaTeX` 将单个换行视作一个空格。

则会得到“汉字分词技术 English”。

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

`linespread` ☆ `linespread = <数值>`

New: 2014-04-23

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

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

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

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

New: 2014-03-13

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

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

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

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

例 8

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

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

New: 2014-03-26

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

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

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

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

第 6 节 文档汉化

6.1 日期汉化

`CTEX` 宏包对显示当前日期的 `\today` 命令进行了汉化, 使之以中文的方式显示今天的日期。如本文档编译时的日期是“2020 年 8 月 23 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“2020 年 8 月 23 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇二〇年八月二十三日”。使用全汉字的日期格式。

`old` 效果为“August 23, 2020”。使用文档原来的(英文)日期格式。

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

例 9

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

CT_{EX} 宏包的中文日期功能实际上是调用 `zhnumber` 宏包完成的。如果需要更多有关日期、时间的命令和更复杂的设置,可以查阅 `zhnumber` 宏包的文档。

6.2 文档标题汉化

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

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

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

设置目录标题名 `\contentsname`。中文默认为“目录”。

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

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

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

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

```
figurename * figurename = <名字>
```

设置图片环境标题名 `\figurename`。中文默认为“图”。

```
tablename * tablename = <名字>
```

设置表格环境标题名 `\tablename`。中文默认为“表”。

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

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

```
indexname * indexname = <名字>
```

设置索引标题名 `\indexname`。中文默认为“索引”。

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

设置附录标题名 `\appendixname`。中文默认为“附录”。

```
bibname * bibname = <名字>
```

设置参考文献标题名。中文默认为“参考文献”。

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

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

`proofname` ★ `proofname = <名字>`

设置证明环境的名称 `\proofname`。中文默认为“证明”。

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

`refname` ★ `refname = <名字>`

设置参考文献标题名 `\refname`。中文默认为“参考文献”。

注意, 三个标准文档类(及相应的 \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 \LaTeX 文档类)中使用 `refname` 选项会报错。

`algorithmname` ★ `algorithmname = <名字>`

设置算法环境标题名 `\algorithmname`。中文默认为“算法”。

`continuation` ★ `continuation = <名字>`

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

6.3 页面格式设置与汉化

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

\LaTeX 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。 \LaTeX 宏包同时也为这些自定义页面格式的包提供了以下宏供使用:

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

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

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

例 10

```

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

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

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

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

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

例 11

```

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

本例的页眉效果大致如下(有页眉线):

第 7 节 章节标题样式设置

CT_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。注意, 斜线 / 的前后不能有空格或者换行。

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

例 12

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

7.1 编号相关

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

numbering = true|false

控制是否对不带星号的章节标题进行编号。各级标题的默认值均为 true。

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,本选项仅仅是否对不带星号的章节标题进行编号。因此,当设置本选项为 false 时,除了不对标题编号以外,其余功能与正常标题一致: 可以编入目录,并生成正确的 hyperref 目录超链接位置和页眉页脚标记。例如:

例 13

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

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

注意, 章节标题是否编号还要受到 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	★
New: 2020-05-06	

secnumdepth = <整数或章节名称>

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

本选项对 beamer/ctexbeamer 文档类无效。

表 5 章节层次			表 6 章节层次的默认设置		
层次	名称	注	文档类	secnumdepth	tocdepth
-1	part	book/report 类	article	3 (subsubsection)	3 (subsubsection)
0	chapter	book/report 类	book/report	2 (subsection)	2 (subsection)
0	part	article 类	beamer	无效	3 (subsubsection)
1	section				
2	subsection				
3	subsubsection				
4	paragraph				
5	subparagraph				

如果没有特别说明, 以下将用“...”代表各级章节标题名。

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

Updated: 2014-03-08

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

例 14

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

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

表 7 name 选项的默认设置

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

.../number ★ number = {⟨数字输出命令⟩}

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

例 15

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

将会使 `\section` 的编号变为大写罗马数字（如 I、II 等）。

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

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

7.2 格式相关

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

表 8 number 选项的默认设置

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

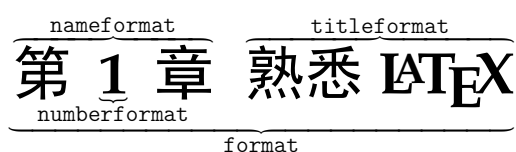


图 1 numberformat, nameformat, titleformat, format 几个选项的作用范围示意

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

Updated: 2020-04-22

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

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

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

例 16

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

`format` 选项的默认设置见表 9。

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

Updated: 2015-06-30

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

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

`nameformat` 选项的默认设置见表 10。

表 9 format 选项的默认设置

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

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

表 10 nameformat 选项的默认设置

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

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

Updated: 2015-06-19

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

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

例 17

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

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

第 4 章

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

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

Updated: 2015-06-30

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

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

例 18

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

上面的代码可以做出类似这样的章标题效果:

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

`titleformat` 选项的默认设置见表 11。

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

Updated: 2014-03-08

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

表 11 titleformat 选项的默认设置

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

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

New: 2015-06-19

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

表 12 aftername 选项的默认设置

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

表 13 aftertitle 选项的默认设置

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

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

New: 2014-03-21

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

表 14 pagestyle 选项的默认设置

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

7.3 间距、缩进相关

section/runin ★ runin = true|false

subsection/runin ★

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

paragraph/runin ★

subparagraph/runin ★

New: 2015-06-27

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

表 15 runin 选项的默认设置

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

.../hang ★ hang = true|false

Updated: 2020-04-23

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

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

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

表 16 hang 选项的默认设置

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

表 17 indent 选项的默认设置

标题名	默认值
part	0pt
chapter	0pt
section	0pt
subsection	0pt
subsubsection	0pt
paragraph	0pt
subparagraph	\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 19

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

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

Updated: 2016-05-10

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

.../afterskip ★ afterskip = {⟨弹性间距⟩}

Updated: 2015-06-27

afterskip 选项控制章节标题与后面下方之间的距离。

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

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

表 18 beforeskip 选项的默认设置

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

表 19 afterskip 选项的默认设置

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

.../fixskip ★

New: 2016-06-03

fixskip = true|false

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

.../break ★

.../break+ ★

New: 2016-09-19

break = {<格式命令>}

break+= {<格式命令>}

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

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

例 20

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

该选项的默认设置见表 20。

表 20 break 选项的默认设置

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

.../afterindent ★

New: 2015-06-27

afterindent = true|false

`afterindent` 选项用于设置章节标题后首段的缩进。

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

该选项的默认设置见表 21。

表 21 afterindent 选项的默认设置

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

7.4 目录、附录相关

`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}>`

New: 2016-10-25

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

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

`\CTEXifname{\protect\numberline{\csname CTExthe#1\endcsname}}{}#2`

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}>`

New: 2016-10-01

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

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

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

New: 2015-06-21

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}>`

Updated: 2014-03-08

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

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

该选项的默认设置见表 23。

表 23 appendix/name 选项的默认设置

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

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

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

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

该选项的默认设置见表 24。

表 24 appendix/number 选项的默认设置

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

7.5 辅助命令

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

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

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

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

New: 2016-09-18

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

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

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

例 21

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


7.6 示例

我们最后举一个稍微复杂的例子,来看看上述选项的综合应用。

例 22

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

本例的设置效果大致如下:

第
1
章

熟悉 \LaTeX

第 8 节 实用命令

8.1 字号与间距

$\backslash\text{zihao}$

$\backslash\text{zihao}$ {〈字号〉}

Updated: 2014-03-08

用于调整字号大小。其中〈字号〉的有效值共有 16 个,如表 25 所示。使用 $\backslash\text{zihao}$ 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。

$\backslash\text{ziju}$

$\backslash\text{ziju}$ {〈中文字符宽度的倍数〉}

Updated: 2014-03-28

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

这个命令会影响 $\backslash\text{ccwd}$ 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。

Updated: 2014-03-27

8.2 中文数字转换

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

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

$\backslash\text{chinese}$	$\backslash\text{chinese} \{ \langle \text{counter} \rangle \}$
Updated: 2016-05-01	$\backslash\text{pagenumbering} \{ \text{chinese} \}$
	$\backslash\text{chinese}$ 命令与 $\backslash\text{roman}$ 等命令的用法类似, 作用在一个 \LaTeX 计数器上, 将计数器的值以中文数字的形式输出。
$\backslash\text{zhnumber}$	$\backslash\text{zhnumber} \{ \langle \text{number} \rangle \}$
New: 2014-03-08	以中文格式输出数字。这里的数字可以是整数、小数和分数。
$\backslash\text{zhdigits}$	$\backslash\text{zhdigits} \{ \langle \text{number} \rangle \}$
New: 2014-03-08	将阿拉伯数字转换为中文数字串。
$\backslash\text{CTEXnumber}$	$\backslash\text{CTEXnumber} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$
	$\backslash \langle \text{macro} \rangle$ 必须是一个 \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 $\backslash \langle \text{macro} \rangle$ 里。对 $\backslash \langle \text{macro} \rangle$ 的定义是局部的, 将它展开一次就可以得到转换结果。
	一般来说, 并不需要使用 $\backslash\text{CTEXnumber}$, 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。
$\backslash\text{CTEXdigits}$	$\backslash\text{CTEXdigits} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$
	$\backslash\text{CTEXdigits}$ 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 CTeX 标志。
-------------------------	------------------------

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

在 $\text{Lua}\text{\LaTeX}$ 下, CTeX 宏集依赖 $\text{LuaTeX-j}\text{\LaTeX}$ 宏包来完成中文支持。该宏包是日本 \TeX 社区的北川弘典、前田一贵、八登崇之等人开发的, 设计目的主要是在 $\text{Lua}\text{\TeX}$ 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 CTeX 禁用了它在 \LaTeX 格式下的大部分设置, 只保留了必要的部分。同时修改了它的字体设置方式, 使得相关命令与 xeCJK 宏包大致相同。

20150420 版以后的 $\text{LuaTeX-j}\text{\LaTeX}$ 宏包开始支持竖排, 但 CTeX 暂不支持竖排。

9.1 *Lua*TeX 下替代字体的设置

AlternateFont <hr/> Updated: 2020-04-30	<pre> \setCJKfamilyfont {<family>} {<base font name>} [<base font features> , AlternateFont = { {<character range₁>} {<alternate font name₁>} , {<character range₂>} {<alternate font name₂>} {<alternate font features₂>} , }]</pre>
---	--

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 <hr/> New: 2014-04-14	<pre> \setCJKfamilyfont {<family>} {<alternate font name>} [CharRange = {<character range>} , <alternate font features>]</pre>
---	--

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

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

例 23

```

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

declarecharrange ★ <hr/> New: 2020-04-30	<pre> \ctexset { declarecharrange = { {<name₁>} {<character range₁>} , {<name₂>} {<character range₂>} , ... } }</pre>
--	---

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

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

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

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

10.1 修改宏包默认选项

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

例 24

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

如上例所示, 宏包选项通常使用 \LaTeX 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 $\text{CT}_{\text{E}}\text{X}$ 宏集的开始处, 定义过宏包选项之后, `\ProcessKeysOptions` 命令之前生效。最好只使用此配置文件修改宏包默认选项。

10.2 宏包载入后的配置

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

例 25

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

例 26

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

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

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

<hr/> cs4size c5size <hr/>	分别相当于 zihao=-4 和 zihao=5, 过时选项。
<hr/> CCT CCTfont <hr/>	相关选项已删除。
<hr/> indent noindent <hr/>	indent 和 noindent 什么也不做, 过时选项。 在中文版式下, ctex 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开。C _T E _X 文档类的相关功能由章节标题的 afterindent 选项的值来确定。
<hr/> zhmap nozhmap <hr/>	zhmap 宏包选项增加了参数, 扩充了功能, 除了支持真假值参数外, 还支持选择 zhmCJK 作为底层中文处理宏包。(4.3 节) nozhmap 选项相当于 zhmap=false。过时选项。
<hr/> winfonts adobefonts nofonts <hr/>	宏包选项 winfonts 相当于 fontset=windows, adobefonts 相当于 fontset=adobe, nofonts 相当于 fontset=none。这几个选项是过时选项, 对于新文档, 应使用 fontset 选项设置不同字体集。 另外, 第 2 版 C _T 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)。
<hr/> punct nopunct <hr/>	旧版本中宏包 punct 选项没有参数, 现在可以用参数设定标点风格(5.3 节)。原有无参形式的 punct 选项相当于 punct=quanjiao。 旧版宏包中 nopunct 选项的效果大致相当于 punct=plain。过时选项, 不推荐使用。
<hr/> cap nocap <hr/>	原有的 cap 和 nocap 选项由新的 scheme 选项代替。(5.3 节) cap 选项相当于 scheme = chinese, nocap 选项相当于 scheme = plain。它们均已过时, 仅因兼容性而保留。
<hr/> space nospace <hr/>	新版本宏包 space 选项增加真假值参数。(5.3 节) nospace 选项相当于 space=false, 成为过时选项。
<hr/> fancyhdr <hr/>	新版本宏包中总是自动处理对 fancyhdr 宏包的兼容性, 而由用户自己使用 \usepackage 载入 fancyhdr 宏包。 fancyhdr 选项过时, 因兼容性保留, 功能是载入 fancyhdr 宏包。

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

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

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

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

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

在新版本中, 上述由 `fntef` 衍生的相关命令和环境均被移除。

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

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

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

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

\CTEXoptions `\CTEXoptions[⟨选项⟩]`
相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

\Chinese `\Chinese{⟨counter⟩}`

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

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

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

例 27

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

11.3 CT_EX 1.02c 以后的 SVN 开发版

`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。

`CTEX` 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 C_TE_X 2.2 之前的版本

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

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

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

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`。

11.5 C_TE_X 2.4.1 和 2.4.2

```
part/fixbeforeskip
chapter/fixbeforeskip
```

这两个选项已经被删除, 相应功能由新的选项 `fixskip` 提供。

11.6 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 12 节 宏集依赖情况与手工安装方法

本节介绍 C_T_EX 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_T_EX 宏集有两个源文件：`ctex.dtx`、`ctexpunct.spa`。使用不同的编译方式时，C_T_EX 依赖的宏包略有不同。在手工安装 C_T_EX 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_T_EX 依赖宏包的详情叙述如下：

- `expl3`、`xparse` 和 `l3keys2e` 宏包。它们属于 `l3kernel` 和 `l3packages` 宏集。
- `indentfirst` 宏包，属于 `tools` 宏集。
- `everyysel` 宏包，属于 `ms` 宏集。
- `zhnumber` 宏包。
- ➔ 以上是各种编译方式都必需的依赖项。
- `CJK` 宏集。
- `CJKpunct` 宏包。
- `xCJK2uni` 宏包。
- `zhmetrics` 宏包。
- `zhmCJK` 宏包，它还依赖
 - `iftex` 宏包。
 - `ltxcmds` 宏包。
 - `kvoptions` 宏包。
 - `kvsetkeys` 宏包。
 - `keyval` 宏包，`graphics` 宏集。
- ➔ 以上是使用 pdf_L_AT_EX 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项，其中 `zhmCJK` 是可选的。
- `xeCJK` 宏集，它还依赖
 - `xtemplate` 宏包，它属于 `l3packages` 宏集。
 - `fontspec` 宏包。
- ➔ 以上是使用 X_L_AT_EX 编译时的依赖项。
- `luatexja` 宏包，它还依赖
 - `adobemapping` 宏包。
 - `luaotfload` 宏包，它还依赖 `lualibs` 宏包。
 - `luatexbase` 宏包，它还依赖 `ctablestack` 宏包。
 - `atbegshi` 宏包。
 - `etoolbox` 宏包。
 - `iftex` 宏包。
 - `infwarerr` 宏包。
 - `ltxcmds` 宏包。
 - `pdfptexcmds` 宏包。
 - `xkeyval` 宏包。
- `fontspec` 宏包。
- ➔ 以上是使用 Lua_L_AT_EX 编译时的依赖项。
- `pxeveryysel` 宏包，属于 `platex-tools` 宏集。
- `zhmetrics-uptex` 宏包。
- ➔ 以上是使用 up_L_AT_EX 编译时的依赖项。

出于一些原因，`zhmCJK` 尚未被收入 T_EX Live 和 MiK_T_EX。因此，若你希望使用 `zhmCJK` 作为 C_T_EX 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 发行版的 ls-R 数据库以完成安装。

其他细节,可参照其 **宏包手册** 中第 3 节的指导。

$\mathrm{CT}_{\mathrm{E}}\mathrm{X}$ 宏集已被 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live 和 $\mathrm{MiK}_{\mathrm{T}}\mathrm{E}_{\mathrm{X}}$ 收录,若无特别理由,我们强烈建议用户使用包管理器安装本宏集。

若要手工安装,请遵循如下步骤:

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

第 13 节 开发人员

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

参考文献

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

第 14 节 代码实现

1 <@@=ctex>

宏包载入检查。

```

2 <*class|ctex>
3 \tl_const:Nx \c__ctex_version_tl
4 { \cs_if_exist_use:cF { ver@ \@currname . \@currentx } { 9999/99/99 } }
5 <*class>
6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl
7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl
8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl
9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl
10 </class>
11 <*ctex>
12 \msg_new:nnnn { ctex } { subpackage-loaded }
13 { Package~`#1'~can~not~be~loaded~with~`ctex'. }
14 {
15   `#1'~is~actually~a~part~of~`ctex'.\\
16   It~is~not~necessary~to~load~it~separately.
17 }
18 \@ifpackageloaded { ctexsize }
19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }
20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }
21 \@ifpackageloaded { ctexheading }
22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }
23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }
24 </ctex>
25 </class|ctex>
26 <*class|style>
27 \RequirePackage { xparse , l3keys2e }
28 </class|style>
29 <*class|ctex>

```

检查 expl3 和 l3keys2e 的版本。

```

30 \msg_new:nnnn { ctex } { l3-too-old }
31 { Support~package~`#1'~too~old. }
32 {
33   Please~update~an~up-to-date~version~of~the~bundles\\
34   `l3kernel'~and~`l3packages'\\
35   using~your~TeX~package~manager~or~from~CTAN.
36 }
37 \@ifpackagelater { expl3 } { 2020/07/17 } { }
38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
39 <*class>
40 \@ifpackagelater { l3keys2e } { 2015/12/20 } { }
41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
42 </class>

```

\c__ctex_engine_str
\c__ctex_engine_file_str

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

```

43 \str_const:Nx \c__ctex_engine_str
44 { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }
45 \msg_new:nnnn { ctex } { engine-not-supported }
46 { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }
47 { You~can~switch~to~xelatex,~lualatex,~pdflatex,~uplatex,~or~aplatex. }
48 \file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }
49 {
50   \str_const:Nx \c__ctex_engine_file_str
51   { ctex-engine- \c__ctex_engine_str .def }
52 }
53 { \msg_critical:nnx { ctex } { engine-not-supported } { \c__ctex_engine_str } }
54 </class|ctex>
55 <*class|ctex|ctexheading|ctexsize>

```

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

```

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

```

14.1 内部函数与变量

临时变量。

```

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

```

在宏包内部使用的键值选项定义、设置命令。

```

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

```

输入文件, 关闭 L^AT_EX3 语法环境, 并设置 @ 为字母类。

```

\ctex_file_input:n
75 \cs_new_protected:Npn \ctex_file_input:n #1
76 {
77   \cctab_begin:N \c__ctex_package_cctab
78   \file_input:n {#1}
79   \cctab_end:
80 }
81 \cctab_const:Nn \c__ctex_package_cctab
82 {
83   \cctab_select:N \c_document_cctab
84   \char_set_catcode_letter:n { 64 }
85 }

```

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。

```

\ctex_scheme_input:n
86 <!*ctexsize>
87 \cs_new_protected:Npn \ctex_scheme_input:n #1
88 {
89   \cctab_begin:N \c__ctex_package_cctab
90   \tl_if_exist:NTF \c__ctex_class_tl
91   {
92     \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
93     { \file_input:n { ctex-scheme- #1 .def } }
94   }
95   { \file_input:n { ctex-scheme- #1 .def } }
96   \cctab_end:
97 }
98 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

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。

```

99 <!*beamer>

```



```

100 \int_new:N \g__ctex_section_depth_int
101 \int_gset:Nn \g__ctex_section_depth_int { 2 }
102 </!beamer>

103 </!ctexsize>
104 </class|ctex|ctexheading|ctexsize>
105 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

106 \msg_new:nnnn { ctex } { package-too-old }
107 { Support~package~`#1'~too~old. }
108 {
109   Please~update~an~up-to-date~version~of~the~package~`#1'\
110   using~your~TeX~package~manager~or~from~CTAN.
111 }

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

112 \sys_if_output_pdf:TF
113 { \cs_new_eq:NN \ifctexpdf \if_true: }
114 { \cs_new_eq:NN \ifctexpdf \if_false: }

```

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

```

115 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
116 \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。

```

117 \cs_new_protected:Npn \ctex_set_default_ccwd:Nn #1#2
118 { \tl_set:Nx #1 { \__ctex_default_ccwd_aux:n {#2} } }
119 \cs_new:Npn \__ctex_default_ccwd_aux:n #1
120 {
121   \exp_not:n {#1}
122   \exp_after:wN \__ctex_default_ccwd_aux:w
123   \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop
124 }
125 \exp_last_unbraced:NNNNo
126 \cs_new:Npn \__ctex_default_ccwd_aux:w #1 { \tl_to_str:n { pt } } #2 \q_stop
127 { \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。

```

128 \tl_new:N \g__ctex_encoding_tl

```

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

```

129 \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。

```

130 \tl_new:N \l__ctex_autoindent_tl

```

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

```

131 \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`。

```

132 \cs_new_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn

```

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

```
<class> 133 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfClass }
<ctex> 134 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfPackage }
```

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

```
135 <*class>
136 \clist_new:N \g__ctex_std_options_clist
137 </class>
```

对无效选项给出警告。

```
138 \msg_new:nnn { ctex } { invalid-option }
139 { Option~`\l_keys_key_tl'~is~invalid~in~current~mode. }
140 \msg_new:nnn { ctex } { invalid-value }
141 { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_tl'. }
```

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

```
\ctex_set_deprecated_option:n
\ctex_deprecated_command:Nn
142 \cs_new_protected:Npn \ctex_deprecated_option:n
143 { \msg_warning:nnn { ctex } { deprecated-option } }
144 \cs_new_protected:Npn \ctex_set_deprecated_option:n #1
145 {
146   \ctex_deprecated_option:n { Option~`#1'~is~set. }
147   \ctex_set:nn { option } { #1 }
148 }
149 \cs_new_protected:Npn \ctex_deprecated_command:Nn #1#2
150 {
151   \msg_warning:nxxx { ctex } { deprecated-command }
152   { \token_to_str:N #1 } { \exp_not:n { #2 } }
153 }
154 \msg_new:nnn { ctex } { deprecated-option }
155 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
156 \msg_new:nnn { ctex } { deprecated-command }
157 { Command~`#1'~is~deprecated.\\ #2 }
158 </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。

```
159 <*class|ctex|ctexsize>
160 \int_new:N \g__ctex_font_size_int
161 \int_gset:Nn \g__ctex_font_size_int { -1 }
162 </class|ctex|ctexsize>
```

14.2 宏包选项

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

```
179 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。

```
180 linespread .fp_set:N = \l__ctex_line_spread_fp ,
181 linespread .initial:n = { \c_nan_fp } ,
182 linespread .value_required:n = true ,
```

autoindent 自动调整段落的首行缩进功能。

```
183 autoindent .choice: ,
184 autoindent .default:n = { true } ,
185 autoindent / true .code:n =
186 {
187   \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
188   \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
189 } ,
190 autoindent / false .code:n =
191 {
192   \tl_clear:N \l__ctex_autoindent_tl
193   \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
194 } ,
195 autoindent / unknown .code:n =
196 {
197   \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
198   \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
199 } ,
```

indent 仅为兼容性保留, 已过时。

```
200 indent .code:n =
201 {
202   \ctex_deprecated_option:n
203   {
204     The~functionality~has~been~removed.\\
205     It's~better~to~set~the~heading~styles~via~`afterindent'~option.
206   }
207 } ,
208 indent .value_forbidden:n = true ,
209 noindent .code:n =
210 {
211   \ctex_deprecated_option:n
212   {
213     The~functionality~has~been~removed.\\
214     It's~better~to~set~the~heading~styles~via~`afterindent'~option.
215   }
216 } ,
217 noindent .value_forbidden:n = true ,
```

GBK 文档编码, 默认为 UTF-8。
UTF8

```
218 GBK .code:n =
219 {
220   \sys_if_engine_pdftex:TF
221   { \tl_gset:Nn \g__ctex_encoding_tl { GBK } }
222   {
223     \msg_warning:nn { ctex } { invalid-option }
224     \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
225   }
226 } ,
227 UTF8 .code:n = { \tl_gset:Nn \g__ctex_encoding_tl { UTF8 } } ,
228 GBK .value_forbidden:n = true ,
229 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。

```

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

```

zhmap

```

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

```

punct 设置标点符号输出格式。

```

257 punct .tl_set:N = \l__ctex_punct_tl ,
258 punct .default:n = { quanjiao } ,
259 punct .initial:n = { quanjiao } ,
260 nopunct .code:n = \ctex_set_deprecated_option:n { punct = plain } ,
261 nopunct .value_forbidden:n = true ,

```

space

```

262 space .choices:nn =
263 { true , auto , false }
264 {
265   \exp_args:Nx \ctex_at_end:n
266   { \ctex_set:n { space = \l_keys_choice_tl } }
267 } ,
268 space .default:n = { true } ,
269 nospace .code:n = { \ctex_deprecated_option:nn { space = false } } ,
270 nospace .value_forbidden:n = true ,

```

heading

```

271 heading .bool_set:N = \l__ctex_heading_bool ,

```

```

272 </class>ctex>
273 <*class>ctex|ctexheading>

```

sub3section

```

274 <!*beamer>

```

sub4section

```

275 sub3section .code:n =
276 { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
277 sub4section .code:n =
278 { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
279 sub3section .value_forbidden:n = true ,
280 sub4section .value_forbidden:n = true ,
281 <!/beamer>

```

```

scheme 282     scheme .tl_set:N = \l__ctex_scheme_tl ,
283 < *ctexheading >
284     scheme .default:n = { plain } ,
285     scheme .initial:n = { plain }
286 }
287 < /ctexheading >
288 < !ctexheading >
289     scheme .default:n = { chinese } ,
290     scheme .initial:n = { chinese } ,
291 < /!ctexheading >
292 < /classctexctexheading >
293 < *classctex >

```

cap 和 **nocap** 是过时选项。

```

294 cap .code:n = { \ctex_set_deprecated_option:n { scheme = chinese } } ,
295 nocap .code:n = { \ctex_set_deprecated_option:n { scheme = plain } } ,
296 cap .value_forbidden:n = true ,
297 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 298 fntef .code:n =
299 {
300     \sys_if_engine_xetex:TF
301     {
302         \ctex_deprecated_option:n { `xeCJKfntef'~package~is~loaded. }
303         \ctex_at_end:n { \RequirePackage { xCJKfntef } }
304     }
305     {
306         \sys_if_engine_pdftex:TF
307         {
308             \ctex_deprecated_option:n { `CJKfntef'~package~is~loaded. }
309             \ctex_at_end:n { \RequirePackage { CJKfntef } }
310         }
311         {
312             \ctex_deprecated_option:n
313             { Furthermore,~option``fntef'~is~invalid~in~current~mode. }
314         }
315     }
316 } ,

fancyhdr 317 fancyhdr .code:n =
318 {
319     \ctex_deprecated_option:n { `fancyhdr'~package~is~loaded. }
320     \ctex_at_end:n { \RequirePackage { fancyhdr } }
321 } ,

hyperref 322 hyperref .code:n =
323 {
324     \ctex_deprecated_option:n { `hyperref'~package~will~be~loaded. }
325     \ctex_at_end:n
326     {
327         \cs_if_exist:NF \hypersetup
328         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
329     }
330     \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
331 } ,
332 }
333 < /classctex >
334 < *classctexctexsize >

```

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

```

335 \tl_clear_new:N \l__ctex_tmp_tl

```

```

336 \clist_map_inline:nn
337 {
338     10pt , 11pt , 12pt ,
339     8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
340 }
341 {
342     \tl_put_right:Nn \l__ctex_tmp_tl
343     {
344         #1 .code:n =
345         < *class>
346         { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
347         < /class>
348         < *class>
349         {
350             \int_gset:Nn \g__ctex_font_size_int { 2 }
351             \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
352         } ,
353         < /class>
354         #1 .value_forbidden:n = true ,
355     }
356 }
357 \exp_args:No \ctex_define_option:n { \l__ctex_tmp_tl }
358 \tl_clear:N \l__ctex_tmp_tl

```

将未知选项传给标准文档类。

```

359 < *class>
360 \ctex_define_option:n
361 {
362     unknown .code:n =
363     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
364 }
365 < /class>

```

载入选项配置文件。

```

< !ctexsize> 366 \ctex_file_input:n { ctexopts.cfg }
367 < /class|ctex|ctexsize>

```

处理宏包选项。

```

368 < *class|style>
369 \ProcessKeysOptions { ctex / option }
370 < /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。

```

371 < *class|ctex>
372 \msg_new:nnn { ctex } { pdftex-utf8 }
373 { UTF8~will~be~used~as~the~default~encoding. }
374 \tl_if_empty:NT \g__ctex_encoding_tl
375 {
376     \sys_if_engine_pdftex:T
377     { \msg_warning:nn { ctex } { pdftex-utf8 } }
378     \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
379 }
380 < /class|ctex>
381 < *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。

```

382 \int_case:nn { \g__ctex_font_size_int }
383 {
384     { 0 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt } }
385     { 1 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt } }
386 }

```

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

```

387 <*article>
388 \tl_const:Nn \c__ctex_class_tl { article }
389 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
390 \LoadClass { article }
391 </article>
392 <*book>
393 \tl_const:Nn \c__ctex_class_tl { book }
394 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
395 \LoadClass { book }
396 </book>
397 <*report>
398 \tl_const:Nn \c__ctex_class_tl { report }
399 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
400 \LoadClass { report }
401 </report>
402 <*beamer>
403 \tl_const:Nn \c__ctex_class_tl { beamer }
404 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
405 \LoadClass { beamer }
406 </beamer>
407 </class>

```

14.3 特定引擎支持与设置

14.3.1 ctexbackend.cfg

对于 $\text{X}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{p}\text{d}\text{f}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{L}\text{u}\text{a}\text{L}\text{A}\text{T}\text{E}\text{X}$ 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 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{u}\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 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 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 或 $\text{u}\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 等编译时, 大多数涉及驱动的宏包选定的默认输出驱动都是 `Dvips`。考虑当前实际使用频率, 以及考虑到 $\text{C}\text{T}\text{E}\text{X}$ 宏集对中文支持的默认方式, 我们在用户使用 $\text{C}\text{T}\text{E}\text{X}$ 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。

```

408 <*backend>
409 \begingroup
410 \expandafter\ifx\csname Umathchardef\endcsname\relax
411 \else\expandafter\endgroup\expandafter\endinput\fi
412 \ifodd
413   \expandafter\ifx\csname pdfoutput\endcsname\relax
414   \expandafter\ifx\csname enablecjktoken\endcsname\relax 0\else 1\fi
415   \else\ifnum\pdfoutput>0 0\else 1\fi\fi\space
416   \def\x#1{%
417     \if\relax\detokenize{#1}\relax
418     \gdef\@classoptionslist{dvipdfmx}%
419   \else
420     \let\CTEX@add@ne
421     \@tfor\x:={dvips}{dvipdfmx}{dvisvgm}\do{%
422       \expandafter\in@\expandafter{\expandafter,\x,}{, #1,}%
423       \ifin@ \let\CTEX@add@tw@ \@break@tfor \fi}%
424     \ifodd\CTEX@add \gdef\@classoptionslist{dvipdfmx, #1}\fi
425   \fi}
426   \expandafter\x\expandafter{\@classoptionslist}
427 \fi
428 \endgroup

```


429 `</backend>`

14.3.2 ctex-engine-pdfTeX.def

`\ctex_set_zhmap:n` 设置 upTeX 字体映射,同时作用于 `\AtBeginDvi` 与 `\AtBeginShipoutFirst`。该宏对 pdfTeX 和 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。

```

430 <*pdfTeX|uptex|aptex>
431 \cs_new_protected:Npn \ctex_set_zhmap:n
432   { \tl_gput_right:Nx \g__ctex_zhmap_tl }
433 \cs_new_protected:Npn \ctex_use_zhmap:
434   { \tl_use:N \g__ctex_zhmap_tl }
435 \cs_new_protected:Npn \ctex_add_dvi_zhmap:
436   { \AtBeginDvi { \ctex_use_zhmap: } }
437 \ctex_after_end_preamble:n { \ctex_add_dvi_zhmap: }
438 \ctex_at_end_package:nn { atbegshi }
439 {
440   \AtBeginShipoutFirst { \ctex_use_zhmap: }
441   \cs_gset_eq:NN \ctex_add_dvi_zhmap: \prg_do_nothing:
442 }
443 \tl_new:N \g__ctex_zhmap_tl
444 \onlypreamble \ctex_set_zhmap:n
445 </pdfTeX|uptex|aptex>

446 <*pdfTeX>

```

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

```

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

```

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

```

449 \cs_new_protected:Npn \ctex_family_cmap:nn #1#2
450   {
451     \cs_if_free:cF { #1 + #2 }
452     {
453       \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
454       { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
455     }
456   }
457 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
458 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn

```

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

```

459 \cs_new_protected:Npn \ctex_add_cmap:n #1
460   {
461     \cs_if_free:NF \CJK@plane
462     { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
463   }
464 \cs_new_protected:Npn \ctex_add_cmap:Nn #1#2
465   {
466     \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
467     #1
468   }
469 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
470 \cs_new_protected:Npn \__ctex_save_cmap:Nn #1#2
471   {
472     \tl_set:Nx \l__ctex_tmp_tl { \str_lowercase:n {#2} \CJK@plane }
473     \tex_immediate:D \tex_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
474     \cs_new_protected:Npx #1
475     {
476       \exp_not:N \tex_pdffontattr:D \exp_not:N \tex_font:D

```

```

477     { /ToUnicode ~ \int_use:N \tex_pdflastobj:D \c_space_tl 0 ~ R }
478   }
479 }

```

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

```

480 \group_begin:
481 \char_set_catcode_other:N \#
482 \sys_if_output_pdf:TF
483 {
484   \group_end:
485   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
486     { \CTEX@Family@CMap {#1} {#2} }
487     {
488       \ctex_at_end_package:nn { cmap }
489       { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
490     }
491     { \ctex_patch_failure:N \DeclareFontFamily }
492   }
493 { \group_end: }

```

首先检查选项,决定是否载入 zhmCJK 宏包。

```

494 \if_bool:N \g__ctex_zhmCJK_bool
495   \PassOptionsToPackage { encoding = \g__ctex_encoding_tl } { zhmCJK }
496   \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

497 \else:
498   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
499     { \RequirePackage { CJK } }
500     { \RequirePackage { CJKutf8 } }
501   \RequirePackage { CJKpunct , CJKspace }

```

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

```

502 \cs_new_protected:Npn \ctex_load_zhmap:nnnn #1#2#3#4
503 {
504   \tl_set:Nn \CJKrmdefault {#1}
505   \tl_set:Nn \CJKsfdefault {#2}
506   \tl_set:Nn \CJKttdefault {#3}
507   \ctex_set_zhmap:n { \ctex_file_input:n { ctex-zhmap- #4 .tex } }
508 }
509 \@onlypreamble \ctex_load_zhmap:nnnn

```

zhmCJK 判断结束。

```

510 \fi:

```

`\ctex_CJK_input:n`
`\CJK@input`

breqn 包可能会在正文中将 ~ 的 \catcode 改为 12 或 13,这将破坏 CJK 对汉字的首字节的定义(\CJK@loadBinding 和 \CJK@loadEncoding)。因此需要确保载入 .enc 和 .bdg 文件时,~ 的 \catcode 为 7。同时要设置 \endlinechar 为 -1。

```

511 \cs_new_protected:Npn \ctex_CJK_input:n #1
512 {
513   \cctab_begin:N \c__ctex_CJK_cctab
514   \file_input:n {#1}
515   \cctab_end:
516 }
517 \cctab_const:Nn \c__ctex_CJK_cctab
518 {
519   \cctab_select:N \c__ctex_package_cctab
520   \int_set:Nn \tex_endlinechar:D { -1 }
521 }
522 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

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

```

523 \if_cs_exist:N \CJK@surr
524   \cs_new_protected:Npn \ctex_plane_to_utfxvibe:Nn #1#2
525   {
526     \tl_set:Nx \l__ctex_tmp_tl {#2}
527     \int_set:Nn \l__ctex_tmp_int
528     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
529     \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
530     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
531     {
532       \int_sub:Nn \l__ctex_tmp_int { 256 }
533       \tl_gset:Nx #1
534       {
535         \int_to_Hex:n
536         { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
537         \int_to_Hex:n
538         { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
539       }
540     }
541   }
542   \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
543 \fi:

```

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

```

544 \AtBeginDocument
545 {
546   \str_if_eq:eeF { \l__ctex_punct_tl } { quanjiao }
547   { \punctstyle { \l__ctex_punct_tl } }
548 }

```

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

```

549 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

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

```

550 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
551   \str_if_eq:onF { \g__ctex_encoding_tl } { UTF8 }
552   {
553     \CJK@makeActive
554     \CJK@loadBinding { standard }
555   }

```

在导言区结束时调用 \CJK@envStart 启用完整的中文功能。

\CJK@envStart 的定义是

```

\def\CJK@envStart#1#2#3{
  \CJK@upperReset
  \ifCJK@lowercase@
    \CJK@lowerReset
  \fi%
  \CJK@makeActive%
  \CJK@global\let\CJK@selectFamily \CJK@selFam
  \CJK@global\let\CJK@selectEnc \CJK@selEnc%
  \def\CJK@@@enc{#2}
  \ifx\CJK@@@enc \@empty
    \PackageInfo{CJK}{

```

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

```

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

```

556 \exp_args:Nx \ctex_at_end_preamble:n
557 {
558   \exp_not:N \CJK@envStart
559   { } { \g__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
560   \exp_not:N \CJKtilde
561 }

```

zhmCJK 判断结束。

```
562 \fi:
```

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

```
563 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces
```

\ctex_ignorespaces_case:N \ctex_set_ignorespaces: 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 \CJKhook 备用。

```

564 \cs_new_protected:Npn \ctex_ignorespaces_case:N #1
565 {
566   \cs_set_protected:Npn \ctex_set_ignorespaces:
567   { \cs_set_eq:NN \CJK@ignorespaces #1 }
568   \ctex_set_ignorespaces:
569 }
570 \cs_new_protected:Npn \ctex_set_ignorespaces:
571 { \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。

```
572 \ctex_gadd_hook: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。

```

573 \cs_new_protected:Npn \ctex_punct_set:n #1
574 {
575   \clist_map_inline:Nn \c__ctex_punct_family_clist
576   {
577     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
578     {
579       \cs_set_eq:cc
580       { CJKpunct@ ##1 @spaces }
581       { c__ctex_ #1 ##1 _punct_spaces_tl }
582     }
583   }
584 }
585 \clist_const:Nn \c__ctex_punct_family_clist
586 {
587   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,

```

```

588     zhsongb , zhheil , zhheib , zhyoub ,
589     zhyahei , zhyaheib , zhpfb , zhpfb
590 }

```

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

```

591 \cs_new_protected:Npn \ctex_punct_map_family:nn #1#2
592 {
593     \cs_if_free:cF { CJKpunct@ #2 @spaces }
594     { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
595 }

```

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

```

596 \cs_new_protected:Npn \ctex_punct_map_bfseries:nn #1#2
597 {
598     \clist_map_inline:nn {#1}
599     {
600         \ctex_punct_map_series:nnn { ##1 } { b } {#2}
601         \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
602     }
603 }
604 \cs_new_protected:Npn \ctex_punct_map_series:nnn #1#2#3
605 {
606     \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
607     \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
608     \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
609     \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
610     \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
611     \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
612 }

```

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

```

613 \cs_new_protected:Npn \ctex_punct_map_itshape:nn #1#2
614 {
615     \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
616     \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
617     \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
618     \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
619     \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
620     \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
621 }

```

`\ctex_punct_space:nn` 定义标点的边界信息。

`\ctexspadef`

```

622 \cs_new_protected:Npn \ctex_punct_space:nn #1#2
623 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
624 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

625 \ctex_file_input:n { ctexspa.def }

```

```

626 </pdfTeX>

```

14.3.3 ctex-engine-xetex.def

```

627 <*xetex>

```

```

628 \RequirePackage { xeCJK }
629 \exp_args:Nx \xeCJKsetup
630 {
631     LoadFandol = false ,
632     PunctStyle = \l__ctex_punct_tl
633 }

```

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`。

```
634 \@ifpackagelater { fontspec } { 2014/05/25 } { }
635 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
636 </xetex>
```

14.3.4 ctex-engine-luatex.def

```
637 <*/luatex>
```

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

```
638 \msg_new:nnn { ctex } { luatexja-loaded }
639 {
640   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
641   Loading~file~`#1'~will~abort!
642 }
643 \@ifpackagelater { luatexja }
644 { \msg_critical:nxx { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
645 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
646 \RequirePackage { luatexja }
647 \@ifpackagelater { luatexja } { 2020/04/12 } { }
648 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
649 \RequirePackage { fontspec }
650 \@ifpackagelater { fontspec } { 2020/02/21 } { }
651 { \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。

```
652 \ctex_at_end:n { \char_set_catcode_comment:n { \ltjlineendcomment } }
```

14.3.4.1 LuaTeX-ja 的默认设置

```
653 \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, 我们把它归入字符范围 3, 改成 ALchar。

```
654 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
655 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
656 \ltjdefcharrange{3}{%
657   "2000-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
658 \ltjdefcharrange{4}{%
659   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
660   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A95F, "A980-"ABFF, "E000-"F8FF,
661   "FB00-"FE0F, "FE20-"FE2F, "FE70-"FEFF, "10000-"1AFF, "1B170-"1FOFF,
662   "1F300-"1FFFF, "2000-"206F}
663 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
664 \ltjdefcharrange{6}{%
665   "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
666   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE1F, "FE30-"FE6F, "FF00-"FFEF,
667   "1B000-"1B16F, "1F100-"1F2FF, "20000-"3FFFF, "E0100-"E01EF}
668 \ltjdefcharrange{7}{%
669   "1100-"11FF, "2F00-"2FFF, "3100-"318F, "31A0-"31EF, "A000-"A4CF,
670   "A960-"A97F, "AC00-"D7FF}
671 \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。

```
672 \ltjdefcharrange{9}{%
673   "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。

```

674 \ltjsetparameter{jacharrange={-1, -2, -3, -4, -5, +6, +7, -8, +9}}
675 \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`。

```

676 \@ifpackagelater{luatexja}{2020/08/08}
677 { \ltjsetparameter { autospacing, autoxspacing, differentjfm = paverage } }
678 {
679   \directlua{
680     local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
681     luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
682   }
683   \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
684     xkanjiskip=.25\zw plus 1pt minus 1pt,
685     autospacing, autoxspacing, jacharrange={-1},
686     yalbaselineshift=\z@, yjabaselineshift=\z@,
687     jcharwidowpenalty=500, differentjfm=paverage
688   }
689 }
690 \ExplSyntaxOn

```

14.3.4.2 LuaTeX-ja 的补丁

```

691 <@@=ctex_ltj>

```

分组中的字符都是 ALchar 类。

```

\CTEX@alchar
\CTEX@beginallalchar
\CTEX@endallalchar
692 \cs_new_protected:Npn \CTEX@alchar #1
693 { \CTEX@beginallalchar #1 \CTEX@endallalchar }
694 \cs_new_protected:Npn \CTEX@beginallalchar
695 {
696   \group_begin:
697   \ctex_ltj_zero_globaldefs:
698   \ltj@allalchar
699 }
700 \cs_new_protected:Npn \CTEX@endallalchar
701 { \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。

```

702 \cs_new_protected:Npn \CTEX@chardef@text@cmd #1
703 {
704   \cs_set_eq:NN \@ifdefinable \@ifdefinable
705   \tl_set:Nn \l__ctex_ltj_cmd_tl {#1}
706   \tex_afterassignment:D \__ctex_ltj_chardef_text_cmd:
707   \tex_chardef:D #1
708 }
709 \tl_new:N \l__ctex_ltj_cmd_tl
710 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd:
711 { \exp_after:wN \__ctex_ltj_chardef_text_cmd_aux:N \l__ctex_ltj_cmd_tl }
712 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd_aux:N #1
713 {
714   \int_compare:nNf {#1} < { "80 }
715   { \cs_set_protected:Npx #1 { \CTEX@alchar { \tex_Uchar:D #1 } } }
716 }

```

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

```

717 \cs_new_protected:Npn \CTEX@text@composite@x #1#2
718 {
719   \CTEX@beginallalchar
720   \cs_if_exist_use:NF #1 {#2}
721   \CTEX@endallalchar
722 }

```


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

```

723 \cs_new_protected:Npx \CTEX@add@unicode@accent #1#2
724 {
725   \CTEX@beginallalchar
726   \exp_not:N \tl_if_blank:nTF {#2} { \tex_Uchar:D "A0 ~ } {#2}
727   \exp_not:N \tex_Uchar:D \tex_numexpr:D #1 \scan_stop:
728   \CTEX@endallalchar
729 }

```

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

```

730 \cs_new_protected:Npn \CTEX@patch@text@cmd #1
731 {
732   \exp_args:Nnc \__ctex_ltj_patch_text_cmd:NN #1
733   { \UnicodeEncodingName \token_to_str:N #1 }
734 }
735 \cs_new_protected:Npn \__ctex_ltj_patch_text_cmd:NN #1#2
736 {
737   \cs_set_eq:NN \CTEX@textcmd #2
738   \ctex_preto_cmd:NnnTF \CTEX@textcmd
739   { \ExplSyntaxOff \makeatletter }
740   { \CTEX@beginallalchar }
741   {
742     \ctex_appto_cmd:NnnTF \CTEX@textcmd
743     { \ExplSyntaxOff \makeatletter }
744     { \CTEX@endallalchar }
745     { \cs_set_eq:NN #2 \CTEX@textcmd }
746     { \ctex_patch_failure:N #1 }
747   }
748   { \ctex_patch_failure:N #1 }
749 }

```

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

```

750 \cs_new_protected:Npn \CTEX@patch@tunec
751 {
752   \cs_set_eq:NN \chardef@text@cmd \CTEX@chardef@text@cmd
753   \ctex_file_input:n { tuenc.def }
754   \cs_set_eq:NN \@text@composite@x \CTEX@text@composite@x
755   \cs_set_eq:NN \add@unicode@accent \CTEX@add@unicode@accent
756   \CTEX@patch@text@cmd \textasteriskcentered
757 }
758 \ifpackageloaded { xunicode }
759 { }
760 { \CTEX@patch@tunec }

```

在 \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`。

```

761 \ctex_at_end_package:nn { xunicode }
762 {
763   \RequirePackage { xunicode-addon }
764   \AtBeginUTFCommand { \CTEX@beginallalchar }
765   \AtEndUTFCommand { \CTEX@endallalchar }
766 }

```

对 `listings` 的补丁是让代码环境支持 `J`Achar 类。

```

767 \ctex_at_end_package:nn { listings }
768 { \RequirePackage { lltjp-listings } }

```

14.3.4.3 Lua 函数

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

```

769 \group_begin:
770 \char_set_catcode_space:n { 32 }
771 \lua_now:e
772 {
773   local func = lua.get_functions_table()
774   local ltjfont = luatexja.jfont
775   local getattribute = tex.getattribute
776   local set_lua = token.set_lua
777   local scan_int, scan_arg = token.scan_int, token.scan_argument
778   local new_luafunction = luatexbase.new_luafunction

```

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

```

779   local id = new_luafunction("ctex")
780   func[id] = ltjfont.add_kyenc_list
781   set_lua("ctex_ltj_add_kyenc:n", id, "global", "protected")

```

`\ctex_ltj_is_kenc:n` 判断编码是否属于 jfont。

```

782   local id = new_luafunction("ctex")
783   func[id] = ltjfont.is_kenc
784   set_lua("ctex_ltj_is_kenc:n", id, "global")

```

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

```

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

```

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

```

785   local id = new_luafunction("ctex")
786   func[id] = function ()
787     local s = scan_arg()
788     local is_braced, is_quoted
789     if s:sub(1,1) == '{' and s:sub(-1)=='}' then
790       is_braced = true; s = s:sub(2,-2)
791     end
792     if s:sub(1,1) == '"' and s:sub(-1) == '"' then
793       is_quoted = true; s = s:sub(2,-2)
794     end
795     s = s .. ltjfont.print_aftl_address()
796     if is_braced then s = '{'..s..'}'
797     elseif is_quoted then s = '"'..s..'"'
798     end
799     tex.sprint(-2, s)
800   end
801   set_lua("ctex_ltj_patch_external_font:n", id, "global")

```

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

```

802   local id = new_luafunction("ctex")
803   func[id] = function ()
804     font.current(getattribute("ltj@curjfnt"))
805   end
806   set_lua("ctex_ltj_use_jfont:", id, "global", "protected")

```

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

```

807   local id = new_luafunction("ctex")

```

```

808     func[id] = function ()
809         local b = tonumber(scan_arg())
810         local e = tonumber(scan_arg())
811         local alt = scan_arg()
812         local base = scan_arg()
813         ltjfont.set_alt_font_latex(b, e, alt, base)
814     end
815     set_lua("ctex_ltj_set_alt_font:nnnn", id, "global", "protected")

```

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

```

816     local id = new_luafunction("ctex")
817     func[id] = function ()
818         local base = scan_arg()
819         ltjfont.clear_alt_font_latex(base)
820     end
821     set_lua("ctex_ltj_clear_alt_font:n", id, "global", "protected")

```

`\ctex_ltj_pickup_alt_font:nn` 定义 #1 的替代字体, #2 是字体大小。

会在内部执行 `\ltj@pickup@altfont@auxy` 和 `\ltj@pickup@altfont@copy`。

```

822     local id = new_luafunction("ctex")
823     func[id] = function ()
824         local base = scan_arg()
825         local size = scan_arg()
826         ltjfont.output_alt_font_cmd("y", base)
827         ltjfont.pickup_alt_font_a(size)
828     end
829     set_lua("ctex_ltj_pickup_alt_font:nn", id, "global", "protected")

```

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

```

830     local id = new_luafunction("ctex")
831     func[id] = function ()
832         local num = scan_int()
833         local base = scan_arg()
834         ltjfont.pickup_alt_font_b(num, base)
835     end
836     set_lua("__ctex_ltj_pickup_alt_font:Nn", id, "global", "protected")

```

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

```

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

```

luatexja 20200412.0 以后的定义为:

```

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

837     local id = new_luafunction("ctex")
838     func[id] = \LuaTeXjavirusion > 20200412.0
839         and ltjfont.does_alt_set
840         or function ()
841             local base = scan_arg()
842             ltjfont.does_alt_set(base)
843         end
844     set_lua("__ctex_ltj_if_alt_set:nT", id, "global")

```

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

```

845     local id = new_luafunction("ctex")

```

```

846     func[id] = function ()
847         tex.set("globaldefs", 0)
848     end
849     set_lua("ctex_ltj_zero_globaldefs:", id, "global", "protected")
850 }
851 \group_end:

```

14.3.4.4 字体切换方式

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

```

852 \cs_new_protected:Npn \ctex_ltj_select_font:
853 {
854     \group_begin: \exp_args:Nnc \group_end:
855     \cs_if_exist_use:NF { \l__ctex_ltj_current_font_tl }
856     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
857 }
858 \tl_new:N \CJK@family
859 \tl_new:N \l__ctex_ltj_current_font_tl
860 \tl_set:Nn \l__ctex_ltj_current_font_tl
861 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

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

```

862 \cs_new_protected:Npn \__ctex_ltj_select_font_aux:
863 {
864     \group_begin:
865     \tl_set_eq:NN \f@encoding \CJK@encoding
866     \tl_set_eq:NN \f@family \CJK@family
867     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
868     \ctex_ltj_pickup_font:
869     \group_end:
870     \font@name
871     \__ctex_ltj_pop_fontname:

```

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

```

872     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
873     { \__ctex_ltj_select_font_aux: }
874 }
875 \cs_new_protected:Npn \__ctex_ltj_push_fontname:n #1
876 {
877     \seq_gpush:No \g__ctex_ltj_fontname_seq { \font@name }
878     \cs_gset_nopar:Npx \font@name {#1}
879 }
880 \cs_new_protected:Npn \__ctex_ltj_pop_fontname:
881 {
882     \seq_gpop:NNT \g__ctex_ltj_fontname_seq \l__ctex_ltj_tmp_tl
883     { \cs_gset_eq:NN \font@name \l__ctex_ltj_tmp_tl }
884 }
885 \seq_new:N \g__ctex_ltj_fontname_seq

```

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

```

886 \cs_new_protected:Npn \ctex_ltj_pickup_font:
887 {
888     \exp_after:wN \cs_if_exist:NF \font@name
889     {
890         \group_begin:
891         \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
892         \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
893         \define@newfont

```

```

894     \group_end:
895   }
896 }
897 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

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

```

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

```

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

```

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

```

898 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
899 {
900   \get@external@font
901   \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
902   {
903     \tl_set:Nx \external@font
904     { \exp_after:wN __ctex_ltj_patch_external_font:w \external@font }
905   }
906   \exp_after:wN \globaljfont \font@name \external@font \scan_stop:

```

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

```

907   \font@name
908   \ctex_ltj_use_jfont:
909   \use:c { \f@encoding + \f@family }
910   \use:c { \curr@fontshape }
911 }

```

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

```

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

```

我们在这里不需要定义新字体,而是设置对应字体的命令。

```

912 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
913 {
914   \ctex_ltj_if_altername_shape_exist:nF { \curr@fontshape }
915   {
916     \group_begin:
917     \tl_set_eq:NN \CJK@family \f@family
918     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
919     {
920       \cs_gset_protected_nopar:Npx \subst@correction
921       {
922         \cs_new_eq:NN
923         \exp_not:c { \l__ctex_ltj_current_font_tl }
924         \font@name
925       }
926       \group_insert_after:N \group_insert_after:N
927       \group_insert_after:N \subst@correction
928     }
929     \group_end:
930   }
931 }

```

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

```

932 \prg_new_conditional:Npnn \ctex_ltj_if_altername_shape_exist:n #1 { T , F , TF }
933 {
934   \__ctex_ltj_if_alt_set:nT {#1} { \prg_return_true: \use_none:n }
935   \prg_return_false:
936 }

```

```

\__ctex_ltj_patch_external_font:w 937 \cs_new:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
938 { \ctex_ltj_patch_external_font:n {#1} ~ at }

```

在 `\selectfont` 中更新替代字体。

```

939 \cs_new_protected:Npn \ctex_ltj_select_altername_font:
940 {
941   \ctex_ltj_if_altername_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
942   {
943     \ctex_ltj_pickup_alt_font:nn
944     { \l__ctex_ltj_current_shape_tl } { \f@size }
945   }
946 }
947 \tl_new:N \l__ctex_ltj_current_shape_tl
948 \tl_set:Nn \l__ctex_ltj_current_shape_tl
949 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

被用在函数 `output_alt_font_cmd` 中,作用是定义替代字体。

```

950 \cs_new_protected:Npn \ltj@pickup@altfont@auxy #1
951 {
952   \cs_if_exist:cF { #1/\f@size }
953   {
954     \group_begin:
955     \use:e { \exp_not:N \split@name #1 / \f@size } \@nil
956     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
957     \ctex_ltj_pickup_font:
958     \group_end:
959     \__ctex_ltj_pop_fontname:
960   }
961 }

```

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。

```

962 \cs_new_protected:Npn \ltj@pickup@altfont@copy #1#2

```

```

963 {
964   \ltj@@getjfontnumber #1
965   \__ctex_ltj_pickup_alt_font:Nn \ltj@tempcntc {#2}
966 }

```

14.3.4.5 数学字体族

以下内容来自 `l1tjfont.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`。

```

967 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
968 {
969   \ctex_ltj_is_kenc:n { \__ctex_ltj_ltj_if_jfont:w #1 / \q_stop }
970   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
971 }
972 \cs_new:Npn \__ctex_ltj_ltj_if_jfont:w #1 / #2 \q_stop
973 {#1}

```

`\ctex_ltj_if_jfont_math:NTF` #1 是一个形式为 $\text{\M@}\langle encoding \rangle$ 的命令, 它由 `\DeclareFontEncoding` 的第三个参数来定义。

```

974 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
975 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
976 \group_begin:
977   \char_set_catcode_other:N M
978   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
979     { \ctex_ltj_if_jfont:nTF {#3} }
980 \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`。

```

981 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
982 {
983   \exp_args:No \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
984   { \ctex_ltj_get_and_define_fonts_ja:nN }
985   { \ctex_ltj_get_and_define_fonts_al:nN }
986   {#1} #2
987 }
988 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
989 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
990 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
991 {
992   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
993   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
994   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
995   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
996   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
997   \ctex_ltj_pickup_font:
998   \tl_put_right:Nx \math@fonts
999   {
1000     \ltj@setpar@global
1001     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
1002     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
1003     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
1004   }
1005 }

```


`\use@mathgroup`
`\ctex_ltj_use_math_group:Nn`

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

```

1006 \cs_new_protected:Npn \ctex_ltj_use_math_group:Nn #1#2
1007 {
1008   \mode_if_math:T
1009   {
1010     \math@bgroup
1011     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
1012     \ctex_ltj_math_group_hook:
1013     \ctex_ltj_if_jfont_math:NTF #1
1014     { \jfam } { \mathgroup } #2 \scan_stop:
1015     \math@egroup
1016   }
1017 }
1018 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
1019 \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。

```

1020 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
1021 {
1022   \group_begin:
1023   \cs_set_protected:Npn #1 ##1##2##3
1024   { \ltjsetmathletter { ##1 } }
1025   #2
1026   \group_end:
1027 }
1028 \ctex_at_end_package:nn { unicode-math }
1029 {
1030   \cs_if_exist:NTF \um_input_math_symbol_table:
1031   {
1032     \ctex_ltj_set_math_letter:NN
1033     \um_sym:nnn
1034     \um_input_math_symbol_table:
1035   }
1036   {
1037     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
1038     \cs_set_protected:Npn \ctex_ltj_math_group_hook:
1039     { \__um_switchto_literal: }
1040     \ctex_ltj_set_math_letter:NN
1041     \__um_sym:nnn
1042     \__um_input_math_symbol_table:
1043   }
1044 }

```

14.3.4.6 字体族的定义与使用

`\ctex_mono_jfm:n`
`\l__ctex_ltj_jfm_tl`

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

```

1045 \cs_new_protected:Npn \ctex_mono_jfm:n #1
1046 {
1047   \str_if_eq:nnTF {#1} { plain }
1048   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
1049   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
1050 }
1051 \tl_new:N \l__ctex_ltj_jfm_tl
1052 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
1053 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

`\CJK@encoding`

在 \LaTeX 下, LuaTeX-ja 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。

```

1054 \tl_const:Nn \CJK@encoding { LTJY3 }
1055 \DeclareFontEncoding { \CJK@encoding } { } { }
1056 \use:e
1057 {
1058   \exp_not:N \DeclareFontSubstitution
1059     { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1060 }
1061 \ctex_ljtj_add_kyenc:n { \CJK@encoding }
1062 \DeclareFontFamily { \CJK@encoding } { song } { }
1063 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1064   { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ljtj_jfm_tl } { }
1065 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1066   { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ljtj_jfm_tl } { }
1067 \tl_const:Nn \c__ctex_ljtj_math_tl { CJKmath }
1068 \DeclareSymbolFont { \c__ctex_ljtj_math_tl }
1069   { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1070 \SetSymbolFont { \c__ctex_ljtj_math_tl } { bold }
1071   { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1072 \int_const:Nn \c__ctex_ljtj_math_fam_int { \use:c { sym \c__ctex_ljtj_math_tl } }
1073 \jfam \c__ctex_ljtj_math_fam_int

```

这是 luatexja-fontspec 中新增的一些字体选项。

```

1074 \newfontfeature { CID }      { cid = #1 }
1075 \newfontfeature { JFM }      { jfm = #1 }
1076 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

1077 \keys_define:nn { fontspec-preparse-external }
1078 {
1079   NoEmbed .code:n =
1080     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ljtj_noembed_wrap:n }
1081 }
1082 \cs_new:Npn \__ctex_ljtj_noembed_wrap:n #1 { psft: #1 }

```

\ctex_ljtj_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。

```

1083 \cs_new_protected:Npn \ctex_ljtj_set_family:nnn #1#2#3
1084 {
1085   \group_begin:
1086   \clist_clear:N \l__ctex_ljtj_char_range_clist
1087   \prop_clear:N \l__ctex_ljtj_alternate_prop
1088   \tl_set:Nn \l__ctex_ljtj_base_CJKfamily_tl {#1}
1089   \keys_set_known:nnN { ctex_ljtj / fontspec } {#2} \l__ctex_ljtj_tmp_tl
1090   \clist_set:No \l__ctex_ljtj_font_options_clist { \l__ctex_ljtj_tmp_tl }
1091   \ctex_ljtj_set_alternate_family:nnF {#1} {#3}
1092   {
1093     \prop_gput:Nnn \g__ctex_ljtj_family_font_name_prop {#1} {#3}
1094     \prop_gput:Nno \g__ctex_ljtj_family_font_options_prop
1095       {#1} { \l__ctex_ljtj_font_options_clist }
1096     \__ctex_ljtj_update_family_uid:N \l__ctex_ljtj_font_options_clist
1097     \__ctex_ljtj_use_global_options:N \l__ctex_ljtj_font_options_clist
1098     \__ctex_ljtj_gset_family_cs:nn {#1} {#3}
1099   }
1100   \group_end:
1101 }
1102 \tl_new:N \l__ctex_ljtj_base_CJKfamily_tl
1103 \clist_new:N \l__ctex_ljtj_font_options_clist

```

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

```

1104 \cs_new_protected:Npn \__ctex_ljtj_use_global_options:N #1
1105 {
1106   \clist_concat:NNN #1 \g__ctex_ljtj_default_features_clist #1

```

```

1107 \clist_put_left:Nx #1
1108 { NFSSEncoding = \CJK@encoding , JFM = \l__ctex_ltj_jfm_tl }
1109 }

```

分别保存 **fontspec** 设置的字体族名、字体名称和字体选项。

```

1110 \prop_new:N \g__ctex_ltj_family_name_prop
1111 \prop_new:N \g__ctex_ltj_family_font_name_prop
1112 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

删除重复的定义,清除替代字体的先前设置。

```

1113 \cs_new_protected:Npn \__ctex_ltj_check_family:n #1
1114 {
1115   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1116   {
1117     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1118     \cs_undefine:c { \__ctex_ltj_altername_cs:n {#1} }
1119     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1120     {
1121       \use:c { \__ctex_ltj_altername_cs:n { clear / #1 } }
1122       \cs_undefine:c { \__ctex_ltj_altername_cs:n { clear / #1 } }
1123       \cs_undefine:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1124       \prop_gremove:Nn \g__ctex_ltj_reset_altername_prop {#1}
1125     }
1126     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1127   }
1128 }
1129 \tl_new:N \l__ctex_ltj_tmp_tl
1130 \msg_new:nnn { ctex } { redefine-family }
1131 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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**。

```

1132 \cs_new_protected:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1133 {
1134   \cs_gset_protected:cpx { \__ctex_ltj_family_csname:n {#1} }
1135   {
1136     \group_begin:
1137     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1138     \exp_not:n { \fontspec_gset_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1139     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1140     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1141     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1142     \__ctex_ltj_set_altername_family:n {#1}
1143     \group_end:
1144   }
1145 }
1146 \tl_new:N \l__ctex_ltj_base_family_tl
1147 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1148 \cs_new:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1149 \cs_new_protected:Npn \__ctex_ltj_set_altername_family:n #1
1150 {
1151   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1152   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1153   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1154   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n {#1} }
1155 }
1156 \cs_new:Npn \__ctex_ltj_altername_cs:n #1 { ctex_ltj/altername_family/#1 }

```

\CJKfamily 切换字体。

```

1157 \NewDocumentCommand \CJKfamily { m }
1158 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1159 \cs_new_protected:Npn \ctex_ltj_switch_family:n #1

```

```

1160 {
1161   \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1162   {
1163     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1164     \selectfont
1165   }
1166   { \__ctex_ltj_family_unknown_warning:n {#1} }
1167 }
1168 \tl_new:N \l_ctex_ltj_family_tl
1169 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

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。

```

1170 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
1171 {
1172   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1173   { \prg_return_true: }
1174   {
1175     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1176     {
1177       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1178       \prg_return_true:
1179     }
1180     { \prg_return_false: }
1181   }
1182 }
1183 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { x } { T , F , TF }

```

```

\__ctex_ltj_family_unknown_warning:n 1184 \cs_new_protected:Npn \__ctex_ltj_family_unknown_warning:n #1
1185 {
1186   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1187   {
1188     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1189     {
1190       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1191       \msg_warning:nnn { ctex } { family-unknown } {#1}
1192     }
1193   }
1194 }
1195 \seq_new:N \g__ctex_ltj_unknown_family_seq
1196 \msg_new:nnn { ctex } { family-unknown }
1197 {
1198   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1199   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1200 }
1201 \cs_new:Npn \__ctex_ltj_msg_def_family_map:n #1
1202 {
1203   \str_case_e:nnF {#1}
1204   {
1205     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1206     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1207     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1208   }
1209   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1210   [...] \{ ... \}
1211 }
1212 \cs_new:Npn \__ctex_ltj_msg_family_map:n #1
1213 {
1214   \str_case_e:nnF {#1}
1215   {
1216     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1217     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1218     \CJKttdefault { \token_to_str:N \CJKttdefault }
1219   }
1220   {#1}
1221 }

```

```

\ctex_ltj_fontspec:nn 1222 \cs_new_protected:Npn \ctex_ltj_fontspec:nn #1#2
1223 {
1224   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1225   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1226   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1227   {
1228     \int_gincr:N \g__ctex_ltj_family_int
1229     \__ctex_ltj_fontspec:enn
1230     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1231     {#1} {#2}
1232   }
1233 }
1234 \cs_new_protected:Npn \ctex_ltj_fontspec:ee #1#2
1235 { \use:e { \ctex_ltj_fontspec:nn {#1} {#2} } }
1236 \cs_new_protected:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1237 {
1238   \bool_if:NT \l__ctex_ltj_add_alterate_bool
1239   {
1240     \cs_if_free:cF
1241     { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1242     {
1243       \cs_gset_eq:cc
1244       { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1245       { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1246       \cs_gset_eq:cc
1247       { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1248       { \__ctex_ltj_alterate_cs:n { clear / \l_ctex_ltj_family_tl } }
1249     }
1250     \bool_set_false:N \l__ctex_ltj_add_alterate_bool
1251   }
1252   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1253   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1254   \ctex_ltj_switch_family:n {#1}
1255 }
1256 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { e }
1257 \prop_new:N \g__ctex_ltj_fontspec_prop

\ctex_ltj_add_font_features:n 1258 \cs_new_protected:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1259 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1260 \cs_new_protected:Npn \ctex_ltj_add_font_features:nn #1#2
1261 {
1262   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1263   {#1} \l__ctex_ltj_tmp_tl
1264   {
1265     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1266     {#1} \l__ctex_ltj_font_options_clist
1267     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1268     \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1269     \ctex_ltj_fontspec:ee
1270     { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1271     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1272   }
1273   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1274 }
1275 \bool_new:N \l__ctex_ltj_add_alterate_bool
1276 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1277 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1278 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1279 {
1280   \token_to_str:N \addCJKfontfeature (s)~ignored.\
1281   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1282 }

```

__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。

```

1283 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1284 {

```

```

1285     \tl_if_novalue:nTF {#2}
1286     { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1287     {
1288         \use:e { #1 {#2} {#3} }
1289         #4
1290     }
1291 }
1292 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1293 {
1294     \use:e { #1 {#4} {#2} }
1295     #3
1296 }

\setCJKfamilyfont 1297 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 1298 {
    \CJKfontspec 1299     \__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 1300     { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1301     { }
1302 }
1303 \NewDocumentCommand \newCJKfontfamily { o m o m }
1304 {
1305     \tl_set:Nx \l__ctex_ltj_tmp_tl
1306     { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
1307     \cs_new_protected:Npx #2
1308     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1309     \__ctex_ltj_pass_args:nnnn
1310     { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1311     { }
1312 }
1313 \NewDocumentCommand \CJKfontspec { o m }
1314 {
1315     \__ctex_ltj_pass_args:nnnn
1316     { \ctex_ltj_fontspec:nn } {#1} {#2}
1317     { \tex_ignorespaces:D }
1318 }
1319 \NewDocumentCommand \addCJKfontfeatures { m }
1320 {
1321     \ctex_ltj_add_font_features:x {#1}
1322     \tex_ignorespaces:D
1323 }
1324 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\setCJKmainfont 1325 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1326 {
\setCJKmonofont 1327     \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1328     { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1329     { \normalfont }
1330 }
1331 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1332 \NewDocumentCommand \setCJKsansfont { o m }
1333 {
1334     \__ctex_ltj_pass_args:nnnn
1335     { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1336     { \normalfont }
1337 }
1338 \NewDocumentCommand \setCJKmonofont { o m }
1339 {
1340     \__ctex_ltj_pass_args:nnnn
1341     { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1342     { \normalfont }
1343 }
1344 \NewDocumentCommand \setCJKmathfont { o m }
1345 {
1346     \__ctex_ltj_pass_args:nnnn
1347     { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1348     { }

```

```

1349 }
1350 \NewDocumentCommand \defaultCJKfontfeatures { m }
1351 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1352 \clist_new:N \g__ctex_ltj_default_features_clist
1353 \@onlypreamble \setCJKmainfont
1354 \@onlypreamble \setCJKsansfont
1355 \@onlypreamble \setCJKmonofont
1356 \@onlypreamble \setCJKmathfont
1357 \@onlypreamble \setCJKromanfont
1358 \@onlypreamble \defaultCJKfontfeatures

```

在导言区结束确认 \CJKfamilydefault 确实存在。

```

1359 \cs_new_protected:Npn \ctex_ltj_ensure_default_family:
1360 {
1361   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1362   {
1363     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1364     {
1365       \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
1366       { \use:n }
1367       {
1368         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1369         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1370         { \use:n }
1371       }
1372     }
1373     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1374     {
1375       \prop_map_break:n
1376       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1377     }
1378   }
1379   \normalfont
1380   \ctex_ltj_update_mathfont:
1381 }
1382 }
1383 }

```

更新数学字体为实际的字体。

```

1384 \cs_new_protected:Npn \ctex_ltj_update_mathfont:
1385 {
1386   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1387   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1388   {
1389     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1390     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1391   }
1392 }
1393 \cs_new_protected:Npn \ctex_ltj_update_mathfont:n #1
1394 {
1395   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1396   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1397   { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1398   \cs_if_free:cTF
1399   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\shapedefault }
1400   {
1401     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1402     { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1403   }
1404   {
1405     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1406     { \c__ctex_ltj_math_family_tl } { \bfdefault } { \shapedefault }
1407   }
1408 }

```

14.3.4.7 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1409 \keys_define:nn { ctex_ltj / fontspec }
1410 {
1411   AlternateFont .code:n = \ctex_ltj_set_alternate_prop:n {#1} ,
1412   AlternateFont .value_required:n = true ,
1413   CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1414   CharRange .value_required:n = true
1415 }

```

\ctex_ltj_set_alternate_prop:n

保存替代字体序列。

```

1416 \cs_new_protected:Npn \ctex_ltj_set_alternate_prop:n #1
1417 { \clist_map_function:nN {#1} \__ctex_ltj_push_alternate_prop:n }
1418 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:n #1
1419 {
1420   \clist_set:Nx \l__ctex_ltj_tmp_clist { \tl_head:n {#1} }
1421   \tl_remove_all:Nn \l__ctex_ltj_tmp_clist { ~ }
1422   \exp_args:No \__ctex_ltj_push_alternate_prop:nn
1423     { \l__ctex_ltj_tmp_clist } {#1}
1424 }
1425 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:nn #1
1426 {
1427   \prop_remove:Nn \l__ctex_ltj_alternate_prop {#1}
1428   \prop_put:Nnn \l__ctex_ltj_alternate_prop {#1}
1429 }
1430 \clist_new:N \l__ctex_ltj_tmp_clist
1431 \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。

```

1432 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnF
1433 {
1434   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1435     { \__ctex_ltj_set_family_aux:nnn }
1436     { \__ctex_ltj_set_alternate_family_aux:nnn }
1437 }
1438 \cs_new_protected:Npn \__ctex_ltj_set_family_aux:nnn #1#2#3
1439 {
1440   \__ctex_ltj_check_family:n {#1}
1441   \prop_if_empty:NF \l__ctex_ltj_alternate_prop
1442     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1443     #3
1444 }
1445 \cs_new_protected:Npn \__ctex_ltj_set_alternate_family_aux:nnn #1#2#3
1446 { \ctex_ltj_set_alternate_family:nn {#1} {#2} }

```

\ctex_ltj_save_alternate_seq:Nn

保存由 AlternateFont 设置的替代字体序列。

```

1447 \cs_new_protected:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1448 {
1449   \prop_map_inline:Nn \l__ctex_ltj_alternate_prop
1450     { \__ctex_ltj_save_alternate_auxi:w ##2 { } \q_mark #1 {#2} }
1451 }
1452 \cs_new_protected:Npn \__ctex_ltj_save_alternate_auxi:w #1#2#
1453 {
1454   \tl_if_blank:nTF {#2}
1455     { \__ctex_ltj_save_alternate_auxii:w {#1} }
1456     { \__ctex_ltj_save_alternate_auxii:w {#1} {#2} }
1457 }
1458 \cs_new_protected:Npn \__ctex_ltj_save_alternate_auxii:w #1#2#3 #4 \q_mark #5#6
1459 {
1460   \clist_set:Nn \l__ctex_ltj_char_range_clist {#1}
1461   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#3}
1462   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist

```



```

1463 \tl_if_blank:nTF {#2}
1464 { \tl_set:Nn \l__ctex_ltj_tmp_tl {#6} }
1465 {
1466   \tl_set:Nx \l__ctex_ltj_tmp_tl { \tl_trim_spaces:n {#2} }
1467   \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#6}
1468 }
1469 \use:e
1470 {
1471   \ctex_ltj_save_alterate_family:Nnnn \exp_not:N #5
1472   { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1473   { \exp_not:o { \l__ctex_ltj_alterate_options_clist } }
1474   { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1475 }
1476 }
1477 \clist_new:N \l__ctex_ltj_alterate_options_clist
1478 \cs_generate_variant:Nn \ctex_ltj_save_alterate_seq:Nn { c }

```

`\ctex_ltj_set_alterate_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。

```

1479 \cs_new_protected:Npn \ctex_ltj_set_alterate_family:nn #1#2
1480 {
1481   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1482   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1483   \ctex_ltj_set_alterate_family:coonn
1484   { \__ctex_ltj_alterate_cs:n {#1} }
1485   { \l__ctex_ltj_char_range_clist }
1486   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1487 }
1488 \cs_new_protected:Npn \ctex_ltj_set_alterate_family:Nnnnn #1#2#3#4#5
1489 {
1490   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1491   { \ctex_ltj_set_alterate_family:nnn {#2} {#3} {#4} }
1492   \ctex_ltj_save_alterate_family:Nnnn #1 {#2} {#3} {#4}
1493 }
1494 \cs_generate_variant:Nn \ctex_ltj_set_alterate_family:Nnnnn { coo }

```

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

```

1495 \cs_new_protected:Npn \ctex_ltj_save_alterate_family:Nnnn #1#2#3#4
1496 {
1497   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1498   \cs_gset_protected:Npx #1
1499   { \exp_not:o { #1 \ctex_ltj_set_alterate_family:nnn {#2} {#3} {#4} } }
1500 }

```

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

```

1501 \cs_new_protected:Npn \ctex_ltj_set_alterate_family:nnn #1#2#3
1502 {
1503   \group_begin:
1504   \cs_set_eq:NN \CJKfamily \use_none:n
1505   \ctex_ltj_swap_cs:NN
1506   \DeclareFontShape@ \ctex_ltj_declare_alterate_shape:nnnnnn
1507   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1508   \fontspec_set_family:Nnn \l__ctex_ltj_alterate_family_tl {#2} {#3}
1509   \group_end:
1510 }
1511 \tl_new:N \l__ctex_ltj_alterate_family_tl

```

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

```

1512 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1513 {
1514   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1515   \cs_set_eq:NN #1 #2

```

```

1516 \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1517 \cs_undefine:N \__ctex_ltj_tmp:w
1518 }

```

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。

```

1519 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1520 \cs_new_protected:Npn \__ctex_ltj_update_family_uid:N #1
1521 {
1522   \int_gincr:N \g__ctex_ltj_family_int
1523   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1524 }
1525 \int_new:N \g__ctex_ltj_family_int

```

\ctex_ltj_declare_alter_nate_shape:nnnnnn

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

```

1526 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1527 {
1528   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1529   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1530   { \l__ctex_ltj_base_family_tl } {#3} {#4} {#2} {#3} {#4}
1531 }

```

\ctex_ltj_set_alter_nate_shape:Nnnnnnn

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

```

1532 \cs_new_protected:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1533 {
1534   \clist_map_inline:Nn #1
1535   {
1536     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1537     {
1538       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1539       \l__ctex_ltj_char_range_tl
1540     }
1541     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1542   }
1543   \__ctex_ltj_save_alter_nate_shape:cn
1544   { \__ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1545   { \ctex_ltj_clear_alt_font:n { \CJK@encoding/#2/#3/#4 } }
1546 }

```

\ctex_ltj_set_alter_nate_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。

```

1547 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1548 { m m > { \SplitArgument { 1 } { -> } } m }
1549 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1550 \cs_new_protected:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1551 {
1552   \ctex_ltj_set_alter_nate_shape:e
1553   {
1554     \__ctex_ltj_range_normalization:nn {#3} {#4}
1555     { \CJK@encoding / \exp_not:n {#2} }
1556     { \CJK@encoding / \exp_not:n {#1} }
1557   }
1558 }
1559 \cs_new_protected:Npn \ctex_ltj_set_alter_nate_shape:n #1
1560 {
1561   \ctex_ltj_set_alt_font:nnnn #1

```

```

1562 \__ctex_ltj_save_altername_shape:cn
1563 { \__ctex_ltj_altername_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1564 { \ctex_ltj_set_alt_font:nnnn #1 }
1565 }
1566 \cs_generate_variant:Nn \ctex_ltj_set_altername_shape:n { e }

```

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

```

1567 \cs_new_protected:Npn \ctex_ltj_set_altername_shape:nnN #1#2#3
1568 {
1569   \tl_map_inline:Nn #3
1570   {
1571     \ctex_ltj_set_altername_shape:n
1572     {
1573       ##1
1574       { \CJK@encoding/#2 }
1575       { \CJK@encoding/#1 }
1576     }
1577   }
1578 }

```

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。

```

1579 \cs_new_protected:Npn \__ctex_ltj_save_altername_shape:Nn #1#2
1580 {
1581   \group_begin:
1582   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1583   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1584   \cs_gset_protected:Npx #1 { \exp_not:o {#1} #2 }
1585   \group_end:
1586 }
1587 \cs_generate_variant:Nn \__ctex_ltj_save_altername_shape:Nn { c }

```

`clearalternamefont`
`resetalternamefont`

清除和重置操作总是全局的。

```

1588 \ctex_define:n
1589 {
1590   clearalternamefont .code:n =
1591   { \clist_map_function:eN {#1} \ctex_ltj_clear_altername_font:n } ,
1592   resetalternamefont .code:n =
1593   { \clist_map_function:eN {#1} \ctex_ltj_reset_altername_font:n } ,
1594   clearalternamefont .default:n = \l_ctex_ltj_family_tl ,
1595   resetalternamefont .default:n = \l_ctex_ltj_family_tl
1596 }
1597 \cs_new_protected:Npn \ctex_ltj_clear_altername_font:n #1
1598 {
1599   \group_begin:
1600   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1601   {
1602     \cs_if_exist_use:cT
1603     { \__ctex_ltj_altername_cs:n { clear / #1 } }
1604     {
1605       \prop_gput:Nno \g__ctex_ltj_reset_altername_prop
1606       {#1} { \l__ctex_ltj_base_family_tl }
1607       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1608       \selectfont
1609     }
1610   }
1611   { \__ctex_ltj_family_unknown_warning:n {#1} }
1612   \group_end:
1613 }
1614 \cs_new_protected:Npn \ctex_ltj_reset_altername_font:n #1
1615 {
1616   \group_begin:

```

```

1617 \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1618 {
1619 \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1620 \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1621 \selectfont
1622 }
1623 \group_end:
1624 }
1625 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1626 \cs_generate_variant:Nn \clist_map_function:nN { e }

```

declarecharrange 预先声明字符范围。

```

1627 \ctex_define:n
1628 {
1629 declarecharrange .code:n = \ctex_ltj_declare_char_range:e {#1} ,
1630 declarecharrange .value_required:n = true
1631 }
1632 \cs_new_protected:Npn \ctex_ltj_declare_char_range:n #1
1633 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1634 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { e }
1635 \cs_new_protected:Npn \__ctex_ltj_declare_char_range:nn #1
1636 { \tl_trim_spaces_apply:nN {#1} \ctex_ltj_declare_char_range:nn }

```

\ctex_ltj_declare_char_range:nn
\g__ctex_ltj_char_range_prop

#1 是名字, #2 是范围。

```

1637 \cs_new_protected:Npn \ctex_ltj_declare_char_range:nn #1#2
1638 {
1639 \tl_clear:N \l__ctex_ltj_char_range_tl
1640 \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1641 \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1642 \ctex_ltj_def_char_range_key:n {#1}
1643 \tl_clear:N \l__ctex_ltj_char_range_tl
1644 }
1645 \tl_new:N \l__ctex_ltj_char_range_tl
1646 \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。

```

1647 \NewDocumentCommand \ctex_ltj_save_char_range:n
1648 { > { \SplitArgument { 1 } { -> } } m }
1649 { \ctex_ltj_save_char_range:nn #1 }
1650 \cs_new_protected:Npn \ctex_ltj_save_char_range:nn #1#2
1651 {
1652 \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1653 { { \__ctex_ltj_range_normalization:nn {#1} {#2} } }
1654 }
1655 \cs_new:Npn \__ctex_ltj_range_normalization:nn #1#2
1656 {
1657 \tl_if_novalue:nTF {#2}
1658 {
1659 { \int_eval:n {#1} }
1660 { \int_eval:n {#1} }
1661 }
1662 {
1663 { \int_eval:n { \tl_if_blank:nTF {#1} { \c__ctex_ltj_range_min_int } {#1} } }
1664 { \int_eval:n { \tl_if_blank:nTF {#2} { \c__ctex_ltj_range_max_int } {#2} } }
1665 }
1666 }
1667 \int_const:Nn \c__ctex_ltj_range_min_int { "80 }
1668 \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。

```

1669 \cs_new_protected:Npn \ctex_ltj_def_char_range_key:n #1
1670 {
1671 \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}

```

```

1672     {
1673       \keys_define:nn { ctex_ltj / fontspec }
1674       { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1675     }
1676   }

```

`\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。

```

1677 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1678 {
1679   \tl_if_blank:nTF {#2}
1680   { \clist_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1681   { \__ctex_ltj_push_alterate_prop:nn {#1} { {#1} #2 } }
1682 }

```

14.3.4.8 其他设置

在抄录环境中禁用 `autospacing` 和 `autoxspacing`。然而, `LuaTeX-j` 还是会使 `J`Achar 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 `J`Achar 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
}

1683 \AtBeginDocument
1684 {
1685   \ctex_appto_cmd:NnnTF \verbatim@font
1686   { \char_set_catcode_letter:n { 64 } }
1687   { \CTEX@verbatim@font@hook }
1688   { }
1689   { \ctex_patch_failure:N \verbatim@font }
1690 }
1691 \cs_new_protected:Npn \CTEX@verbatim@font@hook
1692 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

```

`\@@italiccorr` \LaTeX 的倾斜校正也要重新定义。

```

1693 \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`。

```

1694 \cs_new_protected:Npn \ctex_ltj_set_kanjiskip:N
1695 { \ltj@setpar@global \ltjsetkanjiskip }
1696 \cs_new_protected:Npn \ctex_ltj_set_xkanjiskip:N
1697 { \ltj@setpar@global \ltjsetxkanjiskip }

1698 <@@=ctex>
1699 </luatex>

```

14.3.5 ctex-engine-uptex.def

```

1700 <*uptex|aptex>

```

按 `CJK` 的命名习惯模拟 `\CJKfamily`。

```

1701 \NewDocumentCommand \CJKfamily { m }
1702 { \kanjifamily {#1} \selectfont }

```

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

```

1703 \DeclareErrorKanjifont {JY2}{zhrm}{m}{n}{10}
1704 \DeclareKanjifontSubstitution {JY2}{zhrm}{m}{n}
1705 \DeclareKanjifontSubstitution {JT2}{zhrm}{m}{n}
1706 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
1707 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
1708 \jfam \symmincho

```

\em 取消 $\text{up}\text{\LaTeX}$ 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。 $\text{up}\text{\LaTeX}$ 2016/05/07 的定义有所变化, 这一行为可以由用户通过 **platexrelease** 包改变, 需要分支处理。

```

1709 \ctex_patch_cmd_once:NnnTF \em
1710 { \ExplSyntaxOff }
1711 { \eminnershapes \else \gtfamily \itshape }
1712 { \eminnershapes \else \itshape }
1713 { }
1714 {
1715   \ctex_patch_cmd:Nnn \em
1716     { \mcfamily \upshape \else \gtfamily \itshape }
1717     { \eminnershapes \else \itshape }
1718 }
1719 \cs_set_nopar:Npn \eminnershapes { \upshape }

```

$\text{\ctex_set_upfamily:nnn}$ 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。

```

1720 \cs_new_protected:Npn \ctex_set_upfamily:nnn #1#2#3
1721 {
1722   \DeclareKanjifont{JY2}{#1}{}
1723   \DeclareKanjifont{JT2}{#1}{}
1724   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1725   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1726   \tl_if_empty:nF {#3}
1727   {
1728     \DeclareFontShape{JY2}{#1}{b}{n}{<->~ #3-h}{}
1729     \DeclareFontShape{JT2}{#1}{b}{n}{<->~ #3-v}{}
1730     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1731     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1732   }
1733 }

```

$\text{\ctex_set_upmap:nnn}$ 设置 $\text{up}\text{\TeX}$ 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。

```

1734 \cs_new_protected:Npn \ctex_set_upmap:nnn #1#2#3
1735 {
1736   \ctex_set_zhmap:n
1737   {
1738     \special { pdf:mapline~#1-h~UniGB-UTF16-H~#2 }
1739     \special { pdf:mapline~#1-v~UniGB-UTF16-V~#2 }
1740     \tl_if_empty:nF {#3}
1741     {
1742       \special { pdf:mapline~#1b-h~UniGB-UTF16-H~#3 }
1743       \special { pdf:mapline~#1b-v~UniGB-UTF16-V~#3 }
1744     }
1745   }
1746 }

```

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

```

1747 \cs_new_protected:Npn \ctex_set_upmap_unicode:nnn #1#2#3
1748 {
1749   \ctex_set_zhmap:n
1750   {
1751     \special { pdf:mapline~#1-h~unicode~#2 }
1752     \special { pdf:mapline~#1-v~unicode~#2 }
1753     \tl_if_empty:nF {#3}
1754     {
1755       \special { pdf:mapline~#1b-h~unicode~#3 }
1756       \special { pdf:mapline~#1b-v~unicode~#3 }
1757     }
1758   }
1759 }

```

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

```

1760 \cs_new_protected:Npn \ctex_set_upfonts:nnnnnn #1#2#3#4#5#6
1761 {
1762   \ctex_set_upmap:nnn { upserif } {#1} {#2}
1763   \ctex_set_upmap:nnn { upserifit } {#3} {}
1764   \ctex_set_upmap:nnn { upsans } {#4} {#5}
1765   \ctex_set_upmap:nnn { upmono } {#6} {}
1766 }

```

以下命令只能在导言区使用。

```

1767 \@onlypreamble \ctex_set_upfamily:nnn
1768 \@onlypreamble \ctex_set_upmap:nnn
1769 \@onlypreamble \ctex_set_upmap_unicode:nnn
1770 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

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

```

1771 \RequirePackage { pxeveryysel }
1772 </uptex|aptex>

```

14.3.6 修改主要字体命令

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

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

```

1773 < *pdf|luatex|uptex|aptex>
<pdf|aptex> 1774 \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
1775 \cs_new_protected:Npn \ctex_provide_font_hook:NNN #1#2
1776 {
1777   \exp_args:Nc \__ctex_provide_font_hook_aux:NNNN
1778   { CTEX \cs_to_str:N #2 } #1#2
1779 }
1780 \cs_new_protected:Npn \__ctex_provide_font_hook_aux:NNNN #1#2#3#4
1781 {
1782   \tl_new:N #1
1783   \cs_if_exist:NTF #3
1784   { \tl_gput_right:Nn #3 {#1} }

```

注意此处不能用 `\ctex_patch_cmd:Nnn` 来打补丁,因其会关闭 \LaTeX 3 语法,但 `fontspec` 会在 `\rmfamily` 的命令中相应加入 `\l__fontspec_rmfamily_encoding_tl` 等,导致补丁失败。

```

1785     { \ctex_parse_name:NN \tl_replace_once:Nnn #2 {#4} { #1#4 } }
1786   }
1787 \ctex_provide_font_hook:NNN \rmfamily \@rmfamilyhook \selectfont
1788 \ctex_provide_font_hook:NNN \sffamily \@sffamilyhook \selectfont
1789 \ctex_provide_font_hook:NNN \ttfamily \@ttfamilyhook \selectfont
<pdf|luatex> 1790 \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。

```

1791 \tl_if_exist:NF \CJKfamilydefault
1792 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1793 <*pdf|luatex>
1794 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1795 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1796 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1797 \tl_gput_right:Nn \CTEX@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
1798 \tl_gput_right:Nn \CTEX@sffamilyhook { \CJKfamily { \CJKsfdefault } }
1799 \tl_gput_right:Nn \CTEX@ttfamilyhook { \CJKfamily { \CJKttdefault } }
1800 \tl_gput_right:Nn \CTEX@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }
1801 </pdf|luatex>

```

up \LaTeX 不需要补丁 `\normalfont`,只需要修改 `\kanjifamilydefault`。

```

1802 <*uptex|aptex>
1803 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { zhrm } }
1804 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { zhsf } }
1805 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { zhzt } }
1806 \tl_gput_right:Nn \CTEX@rmfamilyhook { \kanjifamily { \CJKrmdefault } }
1807 \tl_gput_right:Nn \CTEX@sffamilyhook { \kanjifamily { \CJKsfdefault } }
1808 \tl_gput_right:Nn \CTEX@ttfamilyhook { \kanjifamily { \CJKttdefault } }
1809 \tl_gset:Nn \kanjifamilydefault { \CJKfamilydefault }
1810 </uptex|aptex>

```

zhmCJK 判断结束。

```
<pdf|tex> 1811 \fi:
```

使修改立刻生效,保证导言区字体族正确。

```
1812 \normalfont
```

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

```
<pdf|tex> 1813 \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。

```

1814 \cs_new_protected:Npn \ctex_update_default_family:
1815 {
1816   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1817   {
1818     \group_begin:
1819     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1820     \tl_gset:Nx \CJKfamilydefault
1821     {
1822       \str_case:onF { \familydefault }
1823       {
1824         { \rmdefault } { \exp_not:N \CJKrmdefault }
1825         { \sfdefault } { \exp_not:N \CJKsfdefault }
1826         { \ttdefault } { \exp_not:N \CJKttdefault }
1827       }
1828       { \CJKfamilydefault }
1829     }
1830     \group_end:

```



```
1831 }
```

使用 Lua_{La}T_EX 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_{La}T_EX 时, C19rm、JY2rm 等总是有定义的, 不需要确认。

```
<luatex> 1832 \ctex_ljtj_ensure_default_family:
1833 }
```

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

```
1834 \tl_new:N \l__ctex_family_default_init_tl
1835 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1836 \tl_set:Nx \l__ctex_family_default_init_tl
1837 {
1838   \exp_not:N \__ctex_family_default_wrap:n
1839   { \exp_not:o { \CJKfamilydefault } }
1840 }
1841 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1842 <pdf|luatex|uptex|aptex>
```

14.3.7 hyperref 兼容性处理

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

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

```
1843 <*class|ctex>
1844 \@ifpackageloaded { hyperref }
1845 {
1846   \cs_new_protected:Npn \ctex_hypersetup:n #1
1847   { \hypersetup {#1} }
1848 }
1849 {
1850   \cs_new_protected:Npn \ctex_hypersetup:n #1
1851   { \PassOptionsToPackage {#1} { hyperref } }
1852 }
1853 </class|ctex>
```

在 pdf_TE_X 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。

```
1854 <*pdf|ctex>
1855 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1856 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
1857 {
1858   \ctex_hypersetup:n { CJKbookmarks = true }
1859   \sys_if_output_pdf:TF
1860   { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1861   {
1862     \ctex_at_end_package:nn { hyperref }
1863     {
1864       \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1865       {
1866         \AtBeginShipoutFirst
1867         { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1868       }
1869     }
1870   }
1871 }
```

```

1869         { \RequirePackage { xCJK2uni } }
1870     }
1871 }
1872 }
1873 { \ctex_hypersetup:n { pdfencoding = unicode } }
1874 </pdftex>

```

X_YTeX 和 LuaTeX 统一设置 pdfencoding 为 unicode。

```

1875 <*xetex|luatex>
1876 \ctex_hypersetup:n { pdfencoding = unicode }
1877 </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。

```

1878 <*uptex|aptex>
1879 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1880 \ctex_at_end_package:nn { hyperref }
1881 { \AtBeginShipoutFirst { \special { pdf:tounicode~UTF8-UTF16 } } }
1882 </uptex|aptex>

1883 <*pdftex|xetex|luatex|uptex|aptex>

```

14.3.8 CJKfntef、xeCJKfntef 相关设置

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

```

1884 <*pdftex>
1885 \ctex_at_end_package:nn { CJKfntef }
1886 {
1887     \normalem
1888     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1889     { \tl_clear:c { CJK#1color } }
1890 </pdftex>
1891 <*xetex>
1892 \ctex_at_end_package:nn { xeCJKfntef }
1893 {
1894     \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
1895     {
1896         \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1897         { \xeCJKsetup { #1 / format = { } } }
1898     }
1899     {
1900         \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1901         { \tl_clear:c { CJK#1color } }
1902     }
1903 </xetex>
1904 <*pdftex|xetex>
1905 \clist_map_inline:nn
1906 { underdot , underline , underdblline , underwave , sout , xout }
1907 { \__ctex_clear_fntef_color:n {#1} }
1908 }
1909 </pdftex|xetex>

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1910 \cs_new_protected:Npn \ctex_update_ccwd:
\ccwd 1911 <*pdftex|xetex>
1912 {
1913     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1914     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1915 }
1916 </pdftex|xetex>

```

```

1917 <*luatex>
1918 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1919 </luatex>
1920 <*uptex|aptex>
1921 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
1922 </uptex|aptex>
1923 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1924 \cs_new_protected:Npn \ctex_update_ccglue:
1925 <*pdf|tex|xetex>
1926 {
1927   \cs_set_protected:Npn \CJKglue
1928   { \skip_horizontal:N \l__ctex_ccglue_skip }
1929 }
1930 </pdf|tex|xetex>
1931 <*luatex>
1932 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
1933 </luatex>
1934 <*uptex|aptex>
1935 { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
1936 </uptex|aptex>
1937 \skip_new:N \l__ctex_ccglue_skip

```

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

\ctex_if_ccglue_touched: TF

```

1938 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1939 {
1940 <*pdf|tex|xetex>
1941   \if_meaning:w \CJKglue \__ctex_ccglue:
1942   \prg_return_false: \else: \prg_return_true: \fi:
1943 </pdf|tex|xetex>
1944 <*luatex>
1945   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1946   { \prg_return_false: } { \prg_return_true: }
1947 </luatex>
1948 <*uptex|aptex>
1949   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
1950   { \prg_return_false: } { \prg_return_true: }
1951 </uptex|aptex>
1952 }
<pdf|tex|xetex> 1953 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }

```

\ctex_update_em_unit: 将当前汉字的宽度保存到 \ccwd 中备用。不采用 1em, 因为这时的 1em 实际上来自西文字体的信息, 未必等于汉字的宽度, 这似乎在传统的 .tfm 字体上表现更明显。在 pdf_{TeX} 和 X_{TeX} 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_{TeX} 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-j_a 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。

```

1954 \cs_new_protected:Npn \ctex_update_em_unit:
<pdf|tex|xetex> 1955 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1956 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1957 { \dim_set:Nn \ccwd { 1zw } }

```

14.3.10 其他

\ctex_add_to_selectfont:n \CTEX@selectfont@hook

\EverySelectfont 直到文档开始时才有效。为了 \ccwd 和 LuaTeX-j_a 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。`pxeveryysel` 宏包取消了检查,但也需要恢复定义,避免重复使用钩子。`KOMA-Script` 宏包也会进行检查,我们需要小心处理。

```

1958 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1959 {
1960   \cs_set_protected:Npx \CTEX@selectfont@hook
1961   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1962 }
1963 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1964 \if_cs_exist:N \@EverySelectfont@Init
1965 \group_begin:
1966   \cs_set:Npn \__ctex_tmp:N #1
1967   {
1968     \tl_set:Nn \l__ctex_tmp_tl {#1}
1969     \cs_new_eq:NN \CTEX@selectfont@save #1
1970     \cs_new_protected:Npn \__ctex_restore_selectfont:
1971     {
1972       \cs_if_free:NF \scr@new@selectfont
1973       {

```

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

```

1974 <*pdfTeX|uptex|aptex>
1975   \cs_if_free:NF \par@update
1976 <*uptex|aptex>
1977   {
1978     \tl_put_right:Nn \@EverySelectfont@Init
1979     { \tl_put_right:Nn #1 { \par@update } }
1980   }
1981 </uptex|aptex>
1982 <*pdfTeX>
1983   { \tl_put_right:Nn #1 { \par@update } }
1984   \cs_set_eq:NN \scr@selectfont \CTEX@selectfont@save
1985 </pdfTeX>
1986 </pdfTeX|uptex|aptex>
1987   \cs_set_eq:NN \scr@new@selectfont #1
<|pdfTeX> 1988   \cs_set_eq:NN \CTEX@selectfont@save \scr@selectfont
1989   }
1990   \tl_put_left:Nn \@EverySelectfont@Init
1991   { \cs_set_eq:NN #1 \CTEX@selectfont@save }
1992   \cs_undefine:N \__ctex_restore_selectfont:
1993 }
1994 }
1995 \ctex_parse_name:NN \__ctex_tmp:N \selectfont
1996 \exp_last_unbraced:NNo \group_end:
1997 \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
1998 { \ExplSyntaxOff }
1999 { \size@update }
2000 { \CTEX@selectfont@hook \size@update }
2001 { \__ctex_restore_selectfont: }
2002 { \ctex_patch_failure:N \selectfont }
2003 \fi:

```

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

```

2004 <*pdfTeX>
2005 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
2006 </pdfTeX>
2007 <*xetex|luatex|uptex|aptex>
2008 \EverySelectfont { \CTEX@selectfont@hook }
2009 </xetex|luatex|uptex|aptex>

```

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

```

2010 <*luatex>
2011 \ctex_add_to_selectfont:n

```

```

2012 {
2013   \ctex_ltj_select_font:
2014   \ctex_ltj_select_alterate_font:
2015 }
2016 \tl_set:Nn \CJK@family { song } \selectfont
2017 \tl_clear:N \CJK@family
2018 </luatex>

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

2019 <*luatex|uptex|aptex>
2020 \cs_new_protected:Npn \ctex_update_xkanjiskip:
2021 {
2022   \skip_if_eq:nnT
2023   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
2024   { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2025   {
2026     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
2027     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
2028     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2029   }
2030 }
2031 \tl_new:N \l__ctex_xkanjiskip_tl
2032 \tl_set:Nn \l__ctex_xkanjiskip_tl
2033 { .25\zw plus 1pt minus 1pt }
2034 { .25zw plus 1pt minus 1pt }
2035 \skip_new:N \l__ctex_xkanjiskip_skip
2036 \skip_set:Nn \l__ctex_xkanjiskip_skip
2037 { \ltjgetparameter { xkanjiskip } }
2038 { \tex_xkanjiskip:D }

2039 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2040 </luatex|uptex|aptex>

```

\cht 分别从 .jfm 中读取字符高度、深度和宽度，目前仅考虑横排的情况。

\cdp
\c wd

\ctex_update_kanjisize:

```

2041 <*luatex>
2042 \dim_new:N \cht
2043 \dim_new:N \cdp
2044 \dim_new:N \c wd
2045 \group_begin:
2046 \char_set_catcode_space:n { 32 }
2047 \lua_now:e
2048 {
2049   local nulltable = { }
2050   local fmt = luatexja.jfont.font_metric_table
2051   local getattribute = tex.getattribute
2052   local setdimen = tex.setdimen
2053   local func = lua.get_functions_table()
2054   local id = luatexbase.new_luafunction("ctex")
2055   func[id] = function ()
2056     local ft = fmt[getattribute("ltj@curjfont")] or nulltable
2057     local ft = ft and ft.char_type or nulltable
2058     local fk = ft and ft[0] or nulltable
2059     setdimen("cht", fk.height or 0)
2060     setdimen("cdp", fk.depth or 0)
2061     setdimen("c wd", fk.width or ft.zw or 0)
2062   end
2063   token.set_lua("ctex_update_kanjisize:", id, "global", "protected")
2064 }
2065 \group_end:
2066 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2067 </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。

```

2068 \ctex_define:n
2069 {
2070 <*pdf|xetex>
2071   space .choice: ,
2072   space / true .code:n =
2073     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
2074 <xetex>   { \xeCJKsetup { CJKspace = true } } ,
2075   space / auto .code:n =
2076 <pdf|xetex>   { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
2077 <xetex>   { \xeCJKsetup { CJKspace = false } } ,
2078   space / false .code:n =
2079 <pdf|xetex>   { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
2080 <xetex>   { \xeCJKsetup { CJKspace = false } } ,
2081   space .default:n = { true } ,
2082   space .initial:n = { auto }
2083 </pdf|xetex>
2084 <*luatex|uptex|aptex>
2085   space .code:n =
2086     { \msg_warning:nn { ctex } { invalid-option } }
2087 </luatex|uptex|aptex>
2088 }

```

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

```

2089 \ctex_define:n
2090 {
2091   punct .code:n =
2092     {
2093       \tl_set:Nx \l__ctex_punct_tl {#1}
2094 <pdf|xetex>   \punctstyle { \l__ctex_punct_tl }
2095 <xetex>   \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2096 <luatex>   \ctex_mono_jfm:o { \l__ctex_punct_tl }
2097 <uptex|aptex>   \msg_warning:nn { ctex } { invalid-option }
2098     } ,
2099   punct .default:n = { quanjiao } ,
2100 }
2101 </pdf|xetex|luatex|uptex|aptex>

```

14.3.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

<class|ctex> 2102 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.4 用户设置接口

```

\ctexset 2103 <*class|ctex|ctexheading>
2104 \NewDocumentCommand \ctexset { } { \ctex_set:n }
2105 </class|ctex|ctexheading>

```

\CTEXsetup
\CTEXoptions 过时命令。

```

2106 <*class|ctex>
2107 \NewDocumentCommand \CTEXsetup { +0 { } } > { \TrimSpaces } m {
2108   {
2109     \tl_if_blank:nTF {#1}
2110     { \ctex_deprecated_command:Nn \CTEXsetup { } }
2111     {
2112       \ctex_deprecated_command:Nn \CTEXsetup

```

```

2113         { \ctexset {~#2~={~#1~}}~is~set. }
2114         \ctex_set:nn {#2} {#1}
2115     }
2116 }
2117 \NewDocumentCommand \CTEXoptions { +0 { } }
2118 {
2119     \tl_if_blank:nTF {#1}
2120     { \ctex_deprecated_command:Nn \CTEXoptions { } }
2121     {
2122         \ctex_deprecated_command:Nn \CTEXoptions
2123         { \ctexset {~#1~}~is~set. }
2124         \ctex_set:n {#1}
2125     }
2126 }

```

14.5 字距与缩进

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

```

2127 \ctex_define:n
2128 {
2129     autoindent .choice: ,
2130     autoindent .default:n = { true } ,
2131     autoindent / true .code:n =
2132     {
2133         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2134         \ctex_select_size:
2135     } ,
2136     autoindent / false .code:n =
2137     { \tl_clear:N \l__ctex_autoindent_tl } ,
2138     autoindent / unknown .code:n =
2139     {
2140         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2141         \ctex_select_size:
2142     }
2143 }

```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2144 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2145 \cs_new_protected:Npn \ctex_select_size:
2146 { \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。

```

2147 \cs_new_protected:Npn \ctex_update_size:
2148 {
2149     \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2150     {
2151         \ctex_update_stretch:
2152         \ctex_update_parindent:
2153     }
2154     { \ctex_update_ziju: }
2155 }
2156 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2157 \tl_new:N \l__ctex_ziju_tl
2158 \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。

```

2159 \ctex_add_to_selectfont:n
2160 { \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`。

```

2161 \ctex_define:n
2162 {
2163   linestretch .code:n =
2164   {
2165     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2166     \ctex_select_size:
2167   },
2168   linestretch .value_required:n = true
2169 }
2170 \tl_new:N \l__ctex_line_stretch_tl
2171 \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 -TeX 的除法是四舍五入,而我们这里应该用截断。由于没有可展性的要求,直接用原语 `\tex_divide:D` 要比 `\int_div_truncate:nn` 快一些。下面的算法还兼顾到了 `\linewidth` 不为汉字字宽的整数倍的情况。若用户禁用 `linestretch` 并且修改过 `\CJKglue`,则只更新 `\ccwd`,否则设置伸展量为 0.08 倍 `\baselineskip`。注意 `everysel` 的钩子位于 `\size@update` 之前,`\baselineskip` 还未更新,不能直接使用它。

```

2172 \cs_new_protected:Npn \ctex_update_stretch:
2173 {
2174   \ctex_update_em_unit:
2175   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2176   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2177   { \__ctex_update_stretch_auxi: }
2178   { \__ctex_update_stretch_auxii: }
2179 }
2180 \cs_new_protected:Npn \__ctex_update_stretch_auxi:
2181 {
2182   \ctex_if_ccglue_touched:TF
2183   { \ctex_update_ccwd: }
2184   {
2185     \dim_set:Nn \l__ctex_tmp_dim
2186     { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
2187     \skip_set:Nn \l__ctex_ccglue_skip
2188     { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2189     \ctex_update_ccglue:
2190   }
2191 }
2192 \cs_new_protected:Npn \__ctex_update_stretch_auxii:
2193 {
2194   \int_set:Nn \l__ctex_tmp_int
2195   { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2196   \tex_divide:D \l__ctex_tmp_int \ccwd
2197   \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
2198   {
2199     \skip_set:Nn \l__ctex_ccglue_skip
2200     {
2201       \c_zero_dim plus \dim_eval:n
2202       {
2203         ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2204         \l__ctex_tmp_int
2205       }
2206     }
2207   }
2208   { \skip_zero:N \l__ctex_ccglue_skip }
2209   \ctex_update_ccglue:
2210 }

```


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

```

2211 \cs_new_protected:Npn \ctex_update_parindent:
2212 {
2213   \tl_if_empty:NF \l__ctex_autoindent_tl
2214   {
2215     \dim_compare:nNnF \parindent = \c_zero_dim
2216     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2217   }
2218 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2219 \NewDocumentCommand \ziju { m }
2220 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2221 \cs_new_protected:Npn \ctex_ziju:n #1
2222 {
2223   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2224   \ctex_select_size:
2225 }

```

`\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。

```

2226 \cs_new_protected:Npn \ctex_update_ziju:
2227 {
2228   \ctex_update_em_unit:
2229   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2230   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2231   \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 决定伸展还是收缩。

```

2232 {
2233   \dim_set:Nn \l__ctex_tmp_dim
2234   { \linewidth - \ccwd + \l__ctex_ziju_dim }
2235   \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2236   \tex_divide:D \l__ctex_tmp_int \ccwd
2237   \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。

```

2238   \dim_compare:nNnF \parindent = \c_zero_dim
2239   {
2240     \int_compare:nNnF \l__ctex_tmp_int < 3
2241     { \int_sub:Nn \l__ctex_tmp_int { 2 } }
2242   }
2243   \skip_set:Nn \l__ctex_ccglue_skip
2244   {
2245     \l__ctex_ziju_dim
2246     plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2247     minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2248     { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
2249   }
2250 }
2251 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2252 \ctex_update_ccglue:

```

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

```

2253   \ctex_update_parindent:
2254 }
2255 \dim_new:N \l__ctex_ziju_dim

```

`\CTEXindent`
`\CTEXnoindent` 过时命令。

```
2256 \NewDocumentCommand \CTEXindent { }
2257 {
2258   \ctex_deprecated_command:Nn \CTEXindent
2259   { \parindent is~set~to~2\ccwd. }
2260   \ctex_update_ccwd:
2261   \dim_set:Nn \parindent { 2 \ccwd }
2262 }
2263 \NewDocumentCommand \CTEXnoindent { }
2264 {
2265   \ctex_deprecated_command:Nn \CTEXnoindent
2266   { \parindent is~set~to~0pt. }
2267   \dim_zero:N \parindent
2268 }
```

14.6 中文数字与日期

```
2269 \PassOptionsToPackage { encoding = \g__ctex_encoding_tl } { zhnumber }
2270 \RequirePackage { zhnumber }
```

`\chinese`

```
2271 \cs_new:Npn \chinese { \zhnum_counter:n }
2272 \cs_new_eq:NN \@chinese \@zhnum
2273 \cs_new_eq:NN \Chinese \chinese
2274 \cs_new_eq:NN \CTEXcounter \use_none:n
```

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

```
2275 \ctex_at_end_package:nn { enumitem }
2276 {
2277   \cs_if_free:NF \AddEnumerateCounter
2278   {
2279     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2280     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2281     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2282   }
2283 }
```

`\CTEXnumber`
`\CTEXdigits`

```
2284 \NewDocumentCommand \CTEXnumber { m m }
2285 { \protected@edef #1 { \zhnumber {#2} } }
2286 \NewDocumentCommand \CTEXdigits { m m }
2287 { \protected@edef #1 { \zhdigits {#2} } }
```

`today`

```
2288 \cs_set_eq:NN \CTEX@todayold \today
2289 \ctex_define:n
2290 {
2291   today .choice: ,
2292   today / old .code:n =
2293   { \cs_set_eq:NN \today \CTEX@todayold } ,
2294   today / small .code:n =
2295   {
2296     \cs_set_eq:NN \today \zhtoday
2297     \zhnumsetup { time = Arabic }
2298   } ,
2299   today / big .code:n =
2300   {
2301     \cs_set_eq:NN \today \zhtoday
2302     \zhnumsetup { time = Chinese }
2303   } ,
2304   today / unknown .code:n =
2305   { \msg_error:nnx { ctex } { today-undef } {#1} }
2306 }
2307 \msg_new:nnnn { ctex } { today-undef }
2308 { Today~format~`#1'~is~undefined. }
2309 { Available~today~formats~are~`old',~`small',~and~`big'. }
```

14.7 其他中文标题定义

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

```

2310 \tl_if_exist:NF \proofname
2311 {
2312   \tl_new:N \proofname
2313   \tl_set:Nn \proofname { Proof }
2314 }

2315 \ctex_define:n
2316 {
2317   contentsname .tl_set:N = \contentsname ,
2318   listfigurename .tl_set:N = \listfigurename ,
2319   listtablename .tl_set:N = \listtablename ,
2320   figurename .tl_set:N = \figurename ,
2321   tablename .tl_set:N = \tablename ,
2322   abstractname .tl_set:N = \abstractname ,
2323   indexname .tl_set:N = \indexname ,
2324   appendixname .tl_set:N = \appendixname ,
2325   proofname .tl_set:N = \proofname ,
2326   bibname .tl_set:N = \refname
2327   bibname .tl_set:N = \bibname
2328 }
2329 \algorithmname .tl_set:N = \algorithmname ,
2330 \bibname .tl_set:N = \bibname ,
2331 \refname .tl_set:N = \refname ,
2332 \continuation .tl_set:N = \insertcontinuationtext
2333 }

2335 \ctex
2336 \msg_new:nnn { ctex } { ctexbibname }
2337 {
2338   Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.~\
2339   The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2340 }
2341 \tl_if_exist:NTF \insertcontinuationtext
2342 {
2343   \ctex_define:n
2344   {
2345     algorithmname .tl_set:N = \algorithmname ,
2346     bibname .tl_set:N = \bibname ,
2347     refname .tl_set:N = \refname ,
2348     continuation .tl_set:N = \insertcontinuationtext
2349   }
2350 }
2351 {
2352   \tl_if_exist:NTF \bibname
2353   { \ctex_define:n { bibname .tl_set:N = \bibname } }
2354   {
2355     \tl_if_exist:NTF \refname
2356     { \ctex_define:n { bibname .tl_set:N = \refname } }
2357     {
2358       \msg_warning:nn { ctex } { ctexbibname }
2359       \ctex_define:n { bibname .tl_set:N = \ctexbibname }
2360     }
2361   }
2362 }
2363 }
2364 \class{ctex}

```

14.8 中文化的标题结构

本节内容在 C_T_E_X 文档类或打开 heading 选项下生效。

2365 `<*class|heading>`

14.8.1 定义标题格式选项

保存 \section 级以下标题名字。

```
2366 <*article|book|report>
2367 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
2368 { section , subsection , subsubsection , paragraph , subparagraph }
2369 </article|book|report>
```

```
\c__ctex_headings_seq 2370 <*article|book|report>
2371 \seq_new:N \c__ctex_headings_seq
2372 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2373 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2374 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2375 </article|book|report>
2376 <*beamer>
2377 \seq_const_from_clist:Nn \c__ctex_headings_seq
2378 { part , section , subsection }
2379 </beamer>
```

```
\__ctex_initial_heading:n 2380 \cs_new_protected:Npn \__ctex_initial_heading:n #1
2381 {
2382   \tl_new:c { CTEX@pre#1 }
2383   \tl_new:c { CTEX@post#1 }
2384   \tl_const:cx { CTEX@the#1 }
2385   {
2386     \exp_not:c { CTEX@pre#1 }
2387     \exp_not:c { CTEX@the#1 }
2388     \exp_not:c { CTEX@post#1 }
2389   }
2390   \tl_const:cx { CTEX@#1name }
2391   {
2392     \group_begin:
2393     \exp_not:c { CTEX@#1@nameformat }
2394     {
2395       \exp_not:c { CTEX@pre#1 }
2396       \exp_not:N \tl_if_empty:NTF
2397       \exp_not:c { CTEX@#1@numberformat }
2398       { \exp_not:c { CTEX@the#1 } }
2399       {
2400         \group_begin:
2401         \exp_not:c { CTEX@#1@numberformat }
2402         \exp_not:c { CTEX@the#1 }
2403         \group_end:
2404       }
2405       \exp_not:c { CTEX@post#1 }
2406     }
2407   \group_end:
2408 }
2409 }
```

```
\__ctex_def_heading_keys:n 2410 \cs_new_protected:Npn \__ctex_def_heading_keys:n #1
2411 {
2412   \tl_put_right:Nx \l__ctex_tmp_tl
2413   {
2414     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2415     #1 / name .code:n =
2416     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
```

```

2417      #1 / number      .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2418      #1 / before skip .tl_set:N = \exp_not:c { CTEX@#1@before skip } ,
2419      #1 / after skip  .tl_set:N = \exp_not:c { CTEX@#1@after skip } ,
2420      #1 / indent      .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2421      #1 / numbering   .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2422      #1 / numbering   .initial:n = true ,
2423      #1 / before skip .initial:n = \c_zero_skip ,
2424      #1 / after skip  .initial:n = \c_zero_skip ,
2425      #1 / indent      .initial:n = \c_zero_dim ,
2426      #1 / before skip .value_required:n = true ,
2427      #1 / after skip  .value_required:n = true ,
2428      #1 / indent      .value_required:n = true ,
2429  <*article|book|report>
2430      #1 / after indent .bool_set:N = \exp_not:c { CTEX@#1@after indent } ,
2431      #1 / fix skip     .bool_set:N = \exp_not:c { CTEX@#1@fix skip } ,
2432      #1 / hang         .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2433      #1 / hang         .initial:n = true ,
2434      #1 / run in       .bool_set:N = \exp_not:c { CTEX@#1@run in } ,
2435      #1 / to cline     .cs_set:Np = \exp_not:c { CTEX@#1@to cline }
2436                          \exp_not:n { ##1##2 } ,
2437      \__ctex_plus_key_aux:nn {#1} { break } ,
2438  </article|book|report>
2439      \__ctex_plus_key_aux:nn {#1} { format } ,
2440      \__ctex_plus_key_aux:nn {#1} { name format } ,
2441      \__ctex_plus_key_aux:nn {#1} { number format } ,
2442      \__ctex_plus_key_aux:nn {#1} { title format } ,
2443      \__ctex_plus_key_aux:nn {#1} { after name } ,
2444      \__ctex_plus_key_aux:nn {#1} { after title } ,
2445  }
2446 }
2447 \cs_new:Npn \__ctex_plus_key_aux:nn #1#2
2448 {
2449     #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2450     #1 / #2 + .code:n =
2451     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2452     #1 / #2 ~ + .code:n =
2453     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2454 }

```

```

\ctex_assign_heading_name:nn
\__ctex_assign_heading_name:nnn

```

name 的值是一个至多两个元素的逗号分隔列表。由于 \LaTeX 的 `clist` 总是会自动忽略空元素，所以设置 `name={, 章}` 后，第一个元素将会是“章”，必须用空的分组保护空元素：`name={ {}, 章 }`，这在使用中有些许不便。我们可以改用 `seq` 或者手写函数解析参数来加以改进。为实现的简单起见，这里用了 `xparse` 的 `\SplitArgument`，它带有参数的长度检查。

```

2455 \NewDocumentCommand \ctex_assign_heading_name:nn
2456 { m > { \SplitArgument { 1 } { , } } +m }
2457 { \__ctex_assign_heading_name:nnn {#1} #2 }
2458 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2459 {
2460     \tl_set:cn { CTEX@pre#1 } {#2}
2461     \tl_if_novalue:nTF {#3}
2462     { \tl_clear:c { CTEX@post#1 } }
2463     { \tl_set:cn { CTEX@post#1 } {#3} }
2464 }

```

```

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

```

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2465 \group_begin:
2466 <*book|report>
2467 \tl_set:Nn \l__ctex_tmp_tl
2468 {
2469     part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2470     chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2471     chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2472     chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,

```

```

2473     chapter / lofskip .initial:n = \c_zero_skip ,
2474     chapter / lotskip .initial:n = \c_zero_skip ,
2475     chapter / lofskip .value_required:n = true ,
2476     chapter / lotskip .value_required:n = true ,
2477   }
2478 </book|report>
2479 <*article|beamer>
2480 \tl_clear:N \l__ctex_tmp_tl
2481 </article|beamer>

```

定义标题键值选项。

```

2482 \seq_map_inline:Nn \c__ctex_headings_seq
2483 {
2484   \__ctex_initial_heading:n {#1}
2485   \__ctex_def_heading_keys:n {#1}
2486 }
2487 \exp_args:NNo \group_end: \ctex_define:n { \l__ctex_tmp_tl }

```

\CTEX@heading@format@initial

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

```

2488 \cs_new_protected:Npn \CTEX@heading@format@initial
2489 {
2490   \normalfont
2491   \tl_clear:N \l__ctex_autoindent_tl
2492   \int_set:Nn \tex_interlinepenalty:D { 10 000 }
2493   \tex_noindent:D
2494 }

```

\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。

```

2495 \cs_new_protected:Npn \ctex_indent_box:n #1
2496 {
2497   \dim_set:Nn \tex_parindent:D {#1}
2498   \__ctex_insert_indent:
2499 }
2500 \cs_new_protected:Npn \__ctex_insert_indent:
2501 {
2502   \dim_compare:nNnF \tex_parindent:D = \c_zero_dim
2503     { \tex_indent:D }
2504 }
2505 \cs_new_eq:NN \CTEX@indentbox \ctex_indent_box:n

```

14.8.2 标准标题命令的修改

```

2506 <*article|book|report>

```

\CTEX@fixtopskip

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

```

2507 <*book|report>
2508 \cs_new_protected:Npn \CTEX@fixtopskip
2509 {
2510   \CTEX@fixheadingskip
2511   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2512     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2513 }
2514 </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），T_EX 就不会加上 \parskip。这时候就需要用户把 \parskip 加到 beforeskip 或者 afterskip 作为修正。

```

2515 \cs_new_protected:Npn \CTEX@fixheadingskip

```

```

2516 {
2517   \par
2518   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2519   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2520 }
2521 \skip_new:N \l__ctex_heading_skip
2522 \cs_new_protected:Npn \CTEX@setheadingskip
2523 { \skip_set:Nn \l__ctex_heading_skip }
2524 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

\partmark 提供 \partmark。

```

2525 \ProvideDocumentCommand \partmark { m }
2526 { \markboth { } { } }

```

\CTEXifname 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2527 \cs_new_eq:NN \CTEXifname \use_ii:nn
2528 \cs_new_protected:Npn \CTEX@ifnametrue
2529 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2530 \cs_new_protected:Npn \CTEX@ifnamefalse
2531 { \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。

```

2532 <*book|report>
2533 \cs_new_protected:Npn \CTEX@addloflotskip #1
2534 {
2535   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2536   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2537   {
2538     \addtocontents { lof }
2539     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2540   }
2541   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2542   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2543   {
2544     \addtocontents { lot }
2545     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2546   }
2547 }
2548 </book|report>

```

```

\CTEX@addtocline 2549 \cs_new_protected:Npn \CTEX@addtocline #1#2
2550 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

\part 2551 <*article>
2552 \renewcommand\part{%
2553   \if@noskipsec \leavevmode \fi
2554   \par
2555   \CTEX@part@break
2556   % \addvspace{4ex}%
2557   \CTEX@setheadingskip \CTEX@part@beforeskip
2558   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2559   \addvspace \CTEX@headingskip
2560   \ifodd \CTEX@part@afterindent
2561     \@afterindenttrue
2562   \else
2563     \@afterindentfalse
2564   \fi
2565   \secdef\@part\@spart}

```

```

2566 </article>
2567 <*book|report>
2568 \renewcommand\part{%
2569 % \if@openright
2570 % \cleardoublepage
2571 % \else
2572 % \clearpage
2573 % \fi
2574 \CTEX@part@break
2575 % \thispagestyle{plain}%
2576 \thispagestyle{\CTEX@part@pagestyle}%
2577 \if@twocolumn
2578 \onecolumn
2579 \@tempwatrue
2580 \else
2581 \@tempwafalse
2582 \fi
2583 % \null\vfil
2584 \CTEX@setheadingskip \CTEX@part@beforeskip
2585 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2586 \vspace*{\CTEX@headingskip}%
2587 \secdef\@part\@spart}
2588 </book|report>

\@part 2589 <*article>
2590 \def\@part[#1]#2{%
2591 \ifnum \c@secnumdepth >\m@ne
2592 \ifodd \CTEX@part@numbering
2593 \CTEX@ifnametrue
2594 \refstepcounter{part}%
2595 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2596 \else
2597 \CTEX@ifnamefalse
2598 \CTEX@makeanchor{part*}%
2599 % \addcontentsline{toc}{part}{#1}%
2600 \fi
2601 \else
2602 \CTEX@ifnamefalse
2603 \CTEX@makeanchor{part*}%
2604 % \addcontentsline{toc}{part}{#1}%
2605 \fi
2606 \CTEX@getttitle{#1}%
2607 \CTEX@addtoctline{part}{#1}%
2608 \partmark{#1}%
2609 \begingroup
2610 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2611 \CTEX@heading@format@initial
2612 \CTEX@part@format{%
2613 % \ifnum \c@secnumdepth >\m@ne
2614 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2615 % \fi
2616 \CTEX@headinghang{part}%
2617 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2618 % \huge\bfseries #2%
2619 \CTEX@part@titleformat{#2}%
2620 % \markboth{}{}\par
2621 \CTEX@part@aftertitle}\par
2622 \endgroup
2623 \nobreak
2624 % \vskip 3ex
2625 \CTEX@setheadingskip \CTEX@part@afterskip
2626 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2627 \vskip \CTEX@headingskip
2628 \@afterheading}
2629 </article>
2630 <*book|report>
2631 \def\@part[#1]#2{%

```



```

2632 \ifnum \c@secnumdepth >-2\relax
2633 \ifodd \CTEX@part@numbering
2634 \CTEX@ifnametrue
2635 \refstepcounter{part}%
2636 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2637 \else
2638 \CTEX@ifnamefalse
2639 \CTEX@makeanchor{part*}%
2640 % \addcontentsline{toc}{part}{#1}%
2641 \fi
2642 \else
2643 \CTEX@ifnamefalse
2644 \CTEX@makeanchor{part*}%
2645 % \addcontentsline{toc}{part}{#1}%
2646 \fi
2647 \CTEX@getttitle{#1}%
2648 \CTEX@addtocline{part}{#1}%
2649 % \markboth{}{}%
2650 \partmark{#1}%
2651 \begingroup
2652 % \centering \interlinepenalty \@M \normalfont
2653 \CTEX@heading@format@initial
2654 \CTEX@part@format{%
2655 % \ifnum \c@secnumdepth >-2\relax
2656 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2657 % \fi
2658 \CTEX@headinghang{part}%
2659 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2660 % \Huge\bfseries #2\par
2661 \CTEX@part@titleformat{#2}%
2662 \CTEX@part@aftertitle}\par
2663 \endgroup
2664 \@endpart}
2665 </book|report>

\@spart 2666 <*article>
2667 \def\@spart#1{%
2668 \CTEX@ifnamefalse
2669 \CTEX@makeanchor@spart{part*}%
2670 \CTEX@getttitle{#1}%
2671 \begingroup
2672 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2673 \CTEX@heading@format@initial
2674 \CTEX@part@format{%
2675 \CTEX@headinghang{part}{}%
2676 % \huge \bfseries #1\par
2677 \CTEX@part@titleformat{#1}%
2678 \CTEX@part@aftertitle}\par
2679 \endgroup
2680 \nobreak
2681 % \vskip 3ex
2682 \CTEX@setheadingskip \CTEX@part@afterskip
2683 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2684 \vskip \CTEX@headingskip
2685 \@afterheading}
2686 </article>
2687 <*book|report>
2688 \def\@spart#1{%
2689 \CTEX@ifnamefalse
2690 \CTEX@makeanchor@spart{part*}%
2691 \CTEX@getttitle{#1}%
2692 \begingroup
2693 % \centering \interlinepenalty \@M \normalfont
2694 \CTEX@heading@format@initial
2695 \CTEX@part@format{%
2696 \CTEX@headinghang{part}{}%
2697 % \Huge \bfseries #1\par%

```

```

2698      \CTEX@part@titleformat{#1}%
2699      \CTEX@part@aftertitle}\par
2700  \endgroup
2701  \@endpart}
2702  </book|report>

\@endpart 2703  <*book|report>
2704  \def\@endpart{%
2705  % \vfil
2706  \CTEX@setheadingskip \CTEX@part@afterskip
2707  \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2708  \vskip \CTEX@headingskip
2709  \newpage
2710  \if@twoside
2711    \if@openright
2712      \null
2713      \thispagestyle{empty}%
2714      \newpage
2715    \fi
2716  \fi
2717  \if@tempswa
2718    \twocolumn
2719  \fi}
2720  </book|report>

```

14.8.2.2 chapter 的标题

```

2721  <*book|report>

\chapter 2722  \renewcommand\chapter{%
2723  % \if@openright\cleardoublepage\else\clearpage\fi
2724  % \thispagestyle{plain}%
2725  \CTEX@chapter@break
2726  \thispagestyle{\CTEX@chapter@pagestyle}%
2727  \global\@topnum\z@
2728  % \@afterindentfalse
2729  \ifodd \CTEX@chapter@afterindent
2730    \@afterindenttrue
2731  \else
2732    \@afterindentfalse
2733  \fi
2734  \secdef\@chapter\@schapter}

\@chapter 2735  \def\@chapter[#1]#2{%
2736  \ifnum \c@secnumdepth >\m@ne
2737  <*book>
2738    \if@mainmatter
2739  </book>
2740    \ifodd \CTEX@chapter@numbering
2741      \CTEX@ifnametrue
2742      \refstepcounter{chapter}%
2743      % \typeout{\@chapapp\space\thechapter.}%
2744      \typeout{\CTEXthechapter}%
2745      % \addcontentsline{toc}{chapter}
2746      % {\protect\numberline{\thechapter}#1}%
2747    \else
2748      \CTEX@ifnamefalse
2749      \CTEX@makeanchor{\Hy@chapapp*}%
2750      % \addcontentsline{toc}{chapter}{#1}%
2751    \fi
2752  <*book>
2753    \else
2754      \CTEX@ifnamefalse
2755      \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2756      % \addcontentsline{toc}{chapter}{#1}%
2757    \fi
2758  </book>

```

```

2759 \else
2760 \CTEX@ifnamefalse
2761 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2762 % \addcontentsline{toc}{chapter}{#1}%
2763 \fi
2764 \CTEX@getttitle{#1}%
2765 \CTEX@addtocline{chapter}{#1}%
2766 \chaptermark{#1}%
2767 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2768 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2769 \CTEX@addloflotskip{chapter}%
2770 \if@twocolumn
2771 \topnewpage[\@makechapterhead{#2}]%
2772 \else
2773 \@makechapterhead{#2}%
2774 \@afterheading
2775 \fi}

\@schapter 2776 \def\@schapter#1{%
2777 \CTEX@ifnamefalse
2778 \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2779 \CTEX@getttitle{#1}%
2780 \if@twocolumn
2781 \topnewpage[\@makeschapterhead{#1}]%
2782 \else
2783 \@makeschapterhead{#1}%
2784 \@afterheading
2785 \fi}

\@makechapterhead 2786 \def\@makechapterhead#1{%
2787 % \vspace*{50\p@}%
2788 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2789 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2790 \vspace*{\CTEX@headingskip}%
2791 \begingroup
2792 % \parindent \z@ \raggedright \normalfont
2793 \CTEX@heading@format@initial
2794 \CTEX@chapter@format{%
2795 % \ifnum \c@secnumdepth >\m@ne
2796 % \if@mainmatter
2797 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2798 % \fi
2799 % \fi
2800 \CTEX@headinghang{chapter}%
2801 {\CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
2802 % \Huge \bfseries #1\par\nobreak
2803 \CTEX@chapter@titleformat{#1}%
2804 \CTEX@chapter@aftertitle}\par
2805 \endgroup
2806 \nobreak
2807 % \vskip 40\p@
2808 \CTEX@setheadingskip \CTEX@chapter@afterskip
2809 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2810 \vskip \CTEX@headingskip}

\@makeschapterhead 2811 \def\@makeschapterhead#1{%
2812 % \vspace*{50\p@}%
2813 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2814 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2815 \vspace*{\CTEX@headingskip}%
2816 \begingroup
2817 % \parindent \z@ \raggedright \normalfont \interlinepenalty\@M
2818 \CTEX@heading@format@initial
2819 \CTEX@chapter@format{%
2820 \CTEX@headinghang{chapter}{}}%
2821 % \Huge \bfseries #1\par\nobreak
2822 \CTEX@chapter@titleformat{#1}%

```

```

2823 \CTEX@chapter@aftertitle}\par
2824 \endgroup
2825 \nobreak
2826 % \vskip 40\p@
2827 \CTEX@setheadingskip \CTEX@chapter@afterskip
2828 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2829 \vskip \CTEX@headingskip}

2830 </book|report>

```

14.8.2.3 section 类的标题

\startsection L^AT_EX 的标准参数是：

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

```

2831 \def\startsection#1#2#3#4#5#6{%
2832 \if@noskipsec \leavevmode \fi
2833 \par
2834 % \@tempskipa #4\relax
2835 % \@afterindenttrue
2836 % \ifdim \@tempskipa <\z@
2837 % \@tempskipa -\@tempskipa \@afterindentfalse
2838 % \fi
2839 \CTEX@update@sectionformat@n{#1}%
2840 \ifodd \CTEX@afterindent
2841 \@afterindenttrue
2842 \else
2843 \@afterindentfalse
2844 \fi
2845 \if@nobreak
2846 \everypar{}%
2847 \else
2848 % \addpenalty\@secpenalty\addvspace\@tempskipa
2849 \csname CTEX@#1@break\endcsname
2850 \CTEX@setheadingskip{#4}%
2851 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2852 \addvspace \CTEX@headingskip
2853 \fi
2854 \@ifstar
2855 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2856 {\dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

\seccntformat 2857 \def\seccntformat#1{%
2858 % \csname the#1\endcsname\quad}%
2859 \csname CTEX@#1name\endcsname
2860 \csname CTEX@#1aftername\endcsname}

\@sect 2861 \def\@sect#1#2#3#4#5#6[#7]#8{%
2862 \ifnum #2>\c@secnumdepth
2863 \CTEX@ifnamefalse
2864 \CTEX@makeanchor@sect{#1*}%
2865 \let\@svsec\@empty
2866 \else
2867 \ifodd \csname CTEX@#1@numbering\endcsname
2868 \CTEX@ifnametrue
2869 \refstepcounter{#1}%
2870 \protected@edef\@svsec{\@seccntformat{#1}\relax}%
2871 \else
2872 \CTEX@ifnamefalse
2873 \CTEX@makeanchor{#1*}%
2874 \let\@svsec\@empty
2875 \fi
2876 \fi
2877 \CTEX@getttitle{#7}%
2878 % \@tempskipa #5\relax
2879 % \ifdim \@tempskipa>\z@

```

```

2880 \unless \ifodd \CTEX@runin
2881 \begingroup
2882 \CTEX@heading@format@initial
2883 #6{%
2884 % \hangfrom{\hskip #3\relax\@svsec}%
2885 % \interlinepenalty \M #8\@par
2886 \CTEX@sectionhang{#3}{\@svsec}%
2887 \csname CTEX@#1@titleformat\endcsname{#8}%
2888 \csname CTEX@#1@aftertitle\endcsname}\par
2889 \endgroup
2890 \csname #1mark\endcsname{#7}%
2891 % \addcontentsline{toc}{#1}{%
2892 % \ifnum #2>\c@secnumdepth \else
2893 % \protect\numberline{\csname the#1\endcsname}%
2894 % \fi
2895 % #7}%
2896 \CTEX@addtocline{#1}{#7}%
2897 \else
2898 \def\@svsechd{%
2899 #6{%
2900 % \hskip #3\relax \@svsec #8
2901 % {\CTEX@indentbox{#3}}\@svsec
2902 % \csname CTEX@#1@titleformat\endcsname{#8}%
2903 % \csname CTEX@#1@aftertitle\endcsname}%
2904 % \csname #1mark\endcsname{#7}%
2905 % \addcontentsline{toc}{#1}{%
2906 % \ifnum #2>\c@secnumdepth \else
2907 % \protect\numberline{\csname the#1\endcsname}%
2908 % \fi
2909 % #7}%
2910 \CTEX@addtocline{#1}{#7}}%
2911 \fi
2912 \@xsect{#5}}

\@ssect 2913 \def\@ssect#1#2#3#4#5{%
2914 \CTEX@ifnamefalse
2915 \CTEX@getttitle{#5}%
2916 % \@tempskipa #3\relax
2917 % \ifdim \@tempskipa>\z@
2918 \unless \ifodd \CTEX@runin
2919 \begingroup
2920 \CTEX@heading@format@initial
2921 #4{%
2922 % \hangfrom{\hskip #1}%
2923 % \interlinepenalty \M #5\__ctexpar
2924 \CTEX@sectionhang{#1}{}%
2925 \CTEX@titleformat{n}{#5}%
2926 \CTEX@aftertitle}\par
2927 \endgroup
2928 \else
2929 % \def\@svsechd{#4{\hskip #1\relax #5}}%
2930 \def\@svsechd{#4{{\CTEX@indentbox{#1}}}%
2931 \CTEX@titleformat{n}{#5}\CTEX@aftertitle}}%
2932 \fi
2933 \@xsect{#3}}

\@xsect 2934 \def\@xsect#1{%
2935 % \@tempskipa #1\relax
2936 % \ifdim \@tempskipa>\z@
2937 \unless \ifodd \CTEX@runin
2938 \par \nobreak
2939 % \vskip \@tempskipa
2940 \CTEX@setheadingskip{#1}%
2941 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2942 \vskip \CTEX@headingskip
2943 \@afterheading
2944 \else

```

```

2945 \nobreakfalse
2946 \global\@noskipsectrue
2947 \everypar{%
2948   \if@noskipsec
2949     \global\@noskipsecfalse
2950     {\setbox\z@\lastbox}%
2951     \clubpenalty\@M
2952     \begingroup \svsechd \endgroup
2953     \unskip
2954   % \@tempskipa #1\relax
2955   % \hskip -\@tempskipa
2956   \CTEX@heading@glue{#1}%
2957   \else
2958     \clubpenalty \clubpenalty
2959     \everypar{}%
2960   \fi}%
2961 \fi
2962 \ignorespaces}

```

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

```

2963 \cs_new_protected:Npn \CTEX@headinghang #1
2964 {
2965   \ctex_heading_hang:cnn
2966   { \CTEX@#1@hang }
2967   { \use:c { \CTEX@#1@indent } }
2968 }
2969 \cs_new_protected:Npn \CTEX@sectionhang
2970 { \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

```

2971 \cs_new_protected:Npn \ctex_heading_hang:Nnn #1#2#3
2972 {
2973   \dim_set:Nn \tex_parindent:D {#2}
2974   \bool_if:NTF #1
2975   { \ctex_hang_from:n }
2976   { \use:n }
2977   { \__ctex_insert_indent: #3 }
2978 }
2979 \cs_new_protected:Npn \ctex_hang_from:n #1
2980 {
2981   \tex_noindent:D
2982   \hbox_set:Nn \l__ctex_tmp_box {#1}
2983   \tex_hangindent:D = \box_wd:N \l__ctex_tmp_box
2984   \box_use_drop:N \l__ctex_tmp_box
2985 }
2986 \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

```

2987 \cs_new_protected:Npn \ctex_heading_glue:n #1
2988 {
2989   \group_begin:
2990   \skip_set:Nn \l__ctex_heading_skip {#1}
2991   \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
2992   { \skip_horizontal:N \l__ctex_heading_skip }
2993   \group_end:
2994 }
2995 \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。

```

2996 \cs_new_protected:Npn \CTEX@update@sectionformat@n #1
2997 {
2998   \cs_set_eq:Nc \CTEX@titleformat@n { \CTEX@#1@titleformat }
2999   \cs_set_eq:Nc \CTEX@aftertitle { \CTEX@#1@aftertitle }

```

```

3000 \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
3001 \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
3002 \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
3003 \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
3004 }
3005 \cs_new_eq:NN \CTEX@titleformat@n \use:n
3006 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
3007 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
3008 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
3009 \cs_new_eq:NN \CTEX@hang \c_true_bool
3010 \cs_new_eq:NN \CTEX@runin \c_false_bool

\CTEX@part@tocline 3011 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 3012 {
3013 \CTEXifname
3014 { \CTEXthepart \hspace { 1em } }
3015 { }
3016 #2
3017 }
3018 <*book|report>
3019 \cs_new:Npn \CTEX@chapter@tocline #1#2
3020 {
3021 \CTEXifname
3022 { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
3023 { }
3024 #2
3025 }
3026 </book|report>

\CTEXnumberline 3027 \cs_new:Npn \CTEXnumberline #1
3028 {
3029 \CTEXifname
3030 { \protect \numberline { \use:c { CTEXthe #1 } } }
3031 { }
3032 }

3033 \int_zero:N \l__ctex_tmp_int
3034 \seq_map_inline:Nn \c__ctex_section_headings_seq
3035 {
3036 \int_incr:N \l__ctex_tmp_int
3037 \cs_gset_protected:cpx {#1}
3038 {
3039 \exp_not:N \@startsection {#1}
3040 { \int_use:N \l__ctex_tmp_int }
3041 { \exp_not:c { CTEX@#1@indent } }
3042 { \exp_not:c { CTEX@#1@beforeskip } }
3043 { \exp_not:c { CTEX@#1@afterskip } }
3044 { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
3045 }
3046 \cs_new:cpn { CTEX@#1@tocline } ##1##2
3047 { \CTEXnumberline { ##1 } ##2 }
3048 }

```

14.8.2.4 附录标题

```

appendix/name 3049 \ctex_define:n
appendix/number 3050 {
appendix/numbering 3051 appendix .meta:nn = { ctex / appendix } {#1} ,
3052 appendix / name .code:n =
3053 { \ctex_assign_heading_name:nn { appendix } {#1} } ,
3054 appendix / number .tl_set:N = \CTEX@appendix@number ,
3055 appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
3056 appendix / numbering .initial:n = true
3057 }
3058 \tl_new:N \CTEX@preappendix
3059 \tl_new:N \CTEX@postappendix

```

```

\appendix 3060 \cs_new_eq:NN \CTEX@save@appendix \appendix
3061 \cs_gset_protected:Npn \appendix
3062 {
3063   \CTEX@save@appendix
3064   <*article>
3065   \gdef \CTEX@presection { \CTEX@preappendix }
3066   \gdef \CTEX@thesection { \CTEX@appendix@number }
3067   \gdef \CTEX@postsection { \CTEX@postappendix }
3068   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
3069   </article>
3070   <*book|report>
3071   \gdef \CTEX@prechapter { \CTEX@preappendix }
3072   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3073   \gdef \CTEX@postchapter { \CTEX@postappendix }
3074   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3075   </book|report>
3076 }

```

14.8.2.5 设置 hyperref 宏包的标题锚点

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

```

3077 \cs_new_protected:Npn \CTEX@makeanchor #1
3078 { }

```

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

```

3079 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
<article> 3080 { part , spart , sect , ssect }
<book|report> 3081 { part , spart , chapter , schapter , sect , ssect }
3082 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3083 {
3084   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3085   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3086 }

```

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

```

3087 \cs_new_protected:Npn \CTEX@hyperheadinghook
3088 {
3089   \group_begin:
3090   \ifHy@implicit
3091     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3092     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3093     {
3094       \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3095       {
3096         \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3097         \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3098       }
3099     }
3100   \else:
3101     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3102     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3103   \fi:
3104   \group_end:
3105 }

3106 \ctex_at_end_package:n { hyperref }
3107 {
3108   \cs_gset_protected:Npn \CTEX@makeanchor #1
3109   {
3110     \Hy@MakeCurrentHrefAuto {#1}
3111     \Hy@raisedlink

```



```

3112      {
3113        \hyper@anchorstart { \@currentHref }
3114        \hyper@anchorend
3115      }
3116    }
3117    \CTEX@hyperheadinghook
3118  }

```

14.8.2.6 兼容 nameref 宏包

`\CTEX@getttitle` 在 `nameref` 载入后才有意义,与上述 `hyperref` 的处理类似。

```

3119 \cs_new_protected:Npn \CTEX@getttitle #1
3120 { }
3121 \ctex_at_end_package:nn { nameref }
3122 {
3123   \cs_gset_protected:Npn \CTEX@getttitle { \NR@getttitle }
3124   \seq_map_inline:Nn \c__ctex_headings_cs_seq
3125   {
3126     \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
3127     { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
3128   }
3129 }

```

14.8.2.7 兼容 titlesec 宏包

我们修改了 `\@startsection` 的定义,它的第四个(`\<beforeskip>`)和第五个(`\<afterskip>`)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 `beforeskip` 和 `afterskip` 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>`。

```

3130 \cs_new_protected:Npn \ctex_titlesec_hook:
3131 {
3132   \@ifpackagewith { titlesec } { explicit }
3133   {
3134     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3135     \__ctex_titlesec_format_explicit:Nn
3136   }
3137   { }
3138   \clist_map_inline:nn
3139   { indentafter , noindentafter , indentfirst , nonindentfirst }
3140   {
3141     \@ifpackagewith { titlesec } { ##1 }
3142     {
3143       \clist_map_break:n

```

```

3144         { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3145     }
3146     { }
3147 }
3148 \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3149 }
3150 \cs_new_protected:Npn \__ctex_titlesec_hook:n #1
3151 {
3152     \__ctex_titlesec_format:n {#1}
3153     \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3154 }
3155 \cs_new_protected:Npn \__ctex_titlesec_format:n #1
3156 {
3157     \cs_if_free:cF { ttlf@#1 }
3158     { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3159 }
3160 \cs_new_protected:Npn \__ctex_titlesec_format:Nn #1#2
3161 {
3162     \tl_set:Nx #1
3163     {
3164         \bool_if:cTF { CTEX@#2@runin }
3165         { \exp_not:N \ttlh@runin }
3166         { \exp_not:N \ttlh@hang }
3167         \tl_tail:N #1
3168     }
3169 }
3170 \cs_new_protected:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3171 {
3172     \cs_set_nopar:Npx #1 ##1
3173     {
3174         \bool_if:cTF { CTEX@#2@runin }
3175         { \exp_not:N \ttlh@runin }
3176         { \exp_not:N \ttlh@hang }
3177         \exp_args:No \tl_tail:n { #1 { } }
3178     }
3179 }
3180 \cs_new_protected:Npn \__ctex_titlesec_spacing:Nn #1#2
3181 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3182 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3183 {
3184     \exp_not:n { {#1} {#2} {#3} {#4} }
3185     { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3186 }

3187 \@ifpackageloaded { titlesec }
3188 { }
3189 {
3190     \ctex_at_end_package:nn { titlesec }
3191     {
3192         \@ifpackagewith { titlesec } { loadonly }
3193         { }
3194         { \ctex_titlesec_hook: }
3195     }
3196 }

```

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

```

3197 \ctex_at_end_package:nn { titlesec }
3198 {
3199     <*book|report>
3200     \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3201     </book|report>
3202     \cs_if_free:NF \ttl@tocpart
3203     {
3204         \cs_set_protected:Npn \ttl@tocpart
3205         { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3206     }
3207     \seq_map_inline:Nn \c__ctex_headings_seq

```

```

3208     {
3209         \cs_if_exist:cF { ttl@toc #1 }
3210         {
3211             \cs_new_protected:cpx { ttl@toc #1 }
3212             {
3213                 \tl_set:Nn \exp_not:N \ttl@a
3214                 {
3215                     \exp_not:N \protect
3216                     \exp_not:N \numberline { \exp_not:c { CTEXthe #1 } }
3217                 }
3218             }
3219         }
3220     }
3221 }

```

14.8.2.8 兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题¹⁹。

\ctex_titleps_hook: 我们修改 titleps 包的内部命令 \ttl@settopmark 和 \ttl@setsubmark, 将 \CTEXtheXXX 等加入更新队列中。

```

3222 \group_begin:
3223 \char_set_catcode_other:N \#
3224 \cs_new_protected:Npn \ctex_titleps_hook:
3225 {
3226     \ctex_patch_cmd:Nnn \ttl@settopmark
3227     { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3228     {
3229         \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3230         \CTEX@titlepslabel@set {#1}
3231     }
3232     \ctex_patch_cmd:Nnn \ttl@setsubmark
3233     { \protect \@namedef { the#1 } { } }
3234     {
3235         \protect \@namedef { the#1 } { }
3236         \CTEX@titlepslabel@clear {#1}
3237     }
3238     \ctex_patch_cmd:Nnn \ttl@setsubmark
3239     { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3240     {
3241         \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3242         \CTEX@titlepslabel@set {#2}
3243     }
3244 }
3245 \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。

```

3246 \cs_new:Npn \CTEX@titlepslabel@set #1
3247 {
3248     \cs_if_free:cF { CTEXthe#1 }
3249     { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3250 }
3251 \cs_new:Npn \CTEX@titlepslabel@clear #1
3252 {
3253     \cs_if_free:cF { CTEXthe#1 }
3254     { \protect \@namedef { CTEXthe#1 } { } }
3255 }

```

¹⁹<https://github.com/CTeX-org/ctex-kit/issues/217>

`titles` 宏包的功能可以由 `titlesec` 的选项 `pagetypes` 引入。

```
3256 \ctex_at_end_package:nn { titlesec }
3257 { \cs_if_free:NF \ttl@settopmark { \ctex_titles_hook: } }
3258 \ctex_at_end_package:nn { titles } { \ctex_titles_hook: }
```

除此之外,也可以使用 `titles` 提供的命令 `\newtitlemark` 来完成:

```
\newtitlemark { \CTEXthechapter }
\newtitlemark { \CTEXthesection }
```

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

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

```
3259 \ctex_at_end_package:nn { titles }
3260 {
3261   \cs_set_protected:Npn \ttl@setifthe #1
3262   {
3263     \exp_args:Nco \cs_set:Npn { ifthe #1 }
3264     {
3265       \CTEXifname
3266       { \protect \@firstoftwo }
3267       { \protect \@secondoftwo }
3268     }
3269   }
3270   \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3271 }
```

14.8.3 目录标签的宽度

```
\CTEX@toc@width@n 3272 \cs_new_protected:Npn \CTEX@toc@width@n #1
3273 {
3274   \hbox_set:Nn \l__ctex_tmp_box {#1}
3275   \dim_set:Nn \@tempdima
3276   {
3277     \dim_max:nn { \@tempdima }
3278     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
3279   }
3280 }
```

`\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。

```
3281 \group_begin:
3282 \char_set_catcode_other:N \#
3283 \use:n
3284 {
3285   \group_end:
3286   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3287   { \CTEX@toc@width@n {#1} }
3288   { }
3289   { \ctex_patch_failure:N \numberline }
3290   \cs_new_protected:Npn \__ctex_patch_toc_width:n \c_parameter_token 1
3291   {
3292     \ifpackageloaded { \c_parameter_token 1 }
3293     { }
3294     {
3295       \ctex_at_end_package:nn { \c_parameter_token 1 }
3296       {
3297         \ctex_preto_cmd:NnnTF \numberline
3298         { \char_set_catcode_letter:n { 64 } }
3299         { \CTEX@toc@width@n {#1} }
3300         { }
3301       }
3302     }
3303   }
```

```

3301             { \ctex_patch_failure:N \numberline }
3302         }
3303     }
3304 }
3305 }
3306 \__ctex_patch_toc_width:n { tocloft }
3307 \__ctex_patch_toc_width:n { titletoc }

```

14.8.4 页眉信息的修改

```

\ps@headings 3308 <*article>
3309 \if@twoside
3310   \ctex_patch_cmd:Nnn \ps@headings
3311     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3312     { \CTEXifname { \CTEXthesection \quad } { } }
3313   \ctex_patch_cmd:Nnn \ps@headings
3314     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3315     { \CTEXifname { \CTEXthesubsection \quad } { } }
3316 \else:

```

不知为何,标准文档类此处对 secnumdepth 的判断为 0,与 \section 的层次 1 不符。

```

3317   \ctex_patch_cmd:Nnn \ps@headings
3318     { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3319     { \CTEXifname { \CTEXthesection \quad } { } }
3320 \fi:
3321 </article>
3322 <*book|report>
3323 \ctex_patch_cmd:Nnn \ps@headings
3324   {
3325     \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3326     \ifnum \c@secnumdepth > \m@ne
3327       \@chapapp \ \thechapter . ~ \ %
3328     \fi
3329     \fi \fi
3330   }
3331   { \CTEXifname { \CTEXthechapter \quad } { } }
3332 \if@twoside
3333   \ctex_patch_cmd:Nnn \ps@headings
3334     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3335     { \CTEXifname { \CTEXthesection \quad } { } }
3336 \fi:
3337 </book|report>

```

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

```

3338 \ctex_at_end_package:nn { fancyhdr }
3339 {
3340 <*article>
3341   \ctex_patch_cmd:Nnn \ps@fancy
3342     { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3343     { \CTEXifname { \CTEXthesection \quad } { } }
3344   \ctex_patch_cmd:Nnn \ps@fancy
3345     { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3346     { \CTEXifname { \CTEXthesubsection \quad } { } }
3347 </article>
3348 <*book|report>
3349   \ctex_patch_cmd:Nnn \ps@fancy
3350     { \ifnum \c@secnumdepth > \m@ne \@chapapp \ \thechapter . ~ \ \fi }
3351     { \CTEXifname { \CTEXthechapter \quad } { } }
3352   \ctex_patch_cmd:Nnn \ps@fancy
3353     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3354     { \CTEXifname { \CTEXthesection \quad } { } }
3355 </book|report>
3356 }

```

3357 `</article|book|report>`

14.8.5 beamer 标题页模板的修改

3358 `<*beamer>`

3359 `\ExplSyntaxOff`

对应 `\partpage`。

```
3360 \defbeamertemplate*{part page}{CTEX}[1][]{%
3361   \begingroup
3362   % \centering
3363   % {\usebeamerfont{part name}%
3364   % \usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3365   % \vskip1em\par
3366   \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3367   \CTEX@heading@format@initial
3368   \CTEX@part@format{%
3369     \CTEX@indentbox{\CTEX@part@indent}%
3370     \ifodd \CTEX@part@numbering
3371       \CTEX@partname \CTEX@part@aftername
3372     \fi
3373     \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3374   % \usebeamerfont{part title}\insertpart\par
3375     \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3376     \end{beamercolorbox}}%
3377   \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3378   \endgroup}
```

对应 `\sectionpage`。

```
3379 \defbeamertemplate*{section page}{CTEX}[1][]{%
3380   \begingroup
3381   % \centering
3382   % {\usebeamerfont{section name}%
3383   % \usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}
3384   % \vskip1em\par
3385   \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3386   \CTEX@heading@format@initial
3387   \CTEX@section@format{%
3388     \CTEX@indentbox{\CTEX@section@indent}%
3389     \ifodd \CTEX@section@numbering
3390       \CTEX@sectionname \CTEX@section@aftername
3391     \fi
3392     \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3393   % \usebeamerfont{section title}\insertsection\par
3394     \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3395     \end{beamercolorbox}}%
3396   \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3397   \endgroup}
```

对应 `\subsectionpage`。

```
3398 \defbeamertemplate*{subsection page}{CTEX}[1][]{%
3399   \begingroup
3400   % \centering
3401   % {\usebeamerfont{subsection name}%
3402   % \usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
3403   % \vskip1em\par
3404   \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3405   \CTEX@heading@format@initial
3406   \CTEX@subsection@format{%
3407     \CTEX@indentbox{\CTEX@subsection@indent}%
3408     \ifodd \CTEX@subsection@numbering
3409       \CTEX@subsectionname \CTEX@subsection@aftername
3410     \fi
3411     \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3412   % \usebeamerfont{subsection title}\insertsubsection\par
```

```

3413 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3414 \end{beamercolorbox}}%
3415 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3416 \endgroup}

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3417 \defbeamertemplatealias{part page}{default}{CTEX}
3418 \defbeamertemplatealias{section page}{default}{CTEX}
3419 \defbeamertemplatealias{subsection page}{default}{CTEX}
3420 \ExplSyntaxOn
3421 </beamer>

```

14.8.6 标题编号和目录的层次设置

secnumdepth **tocdepth** secnumdepth 在 beamer 下无意义。

```

3422 \ctex_define:n
3423 {
3424 <!*beamer>
3425   secnumdepth .code:n = \ctex_heading_depth:ne { secnumdepth } {#1} ,
3426   secnumdepth .value_required:n = true ,
3427 </!beamer>
3428   tocdepth .code:n = \ctex_heading_depth:ne { tocdepth } {#1} ,
3429   tocdepth .value_required:n = true
3430 }

```

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

```

3431 \cs_new_protected:Npn \ctex_heading_depth:nn #1#2
3432 {
3433   \prop_get:NnNTF \c__ctex_heading_level_prop {#2} \l__ctex_tmp_tl
3434   { \setcounter {#1} { \l__ctex_tmp_tl } }
3435   { \setcounter {#1} { \int_eval:n {#2} } }
3436 }
3437 \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。

```

3438 \prop_const_from_keyval:Nn \c__ctex_heading_level_prop
3439 {
3440 <*article|beamer>
3441   part = 0 ,
3442 </article|beamer>
3443 <*book|report>
3444   part = -1 ,
3445   chapter = 0 ,
3446 </book|report>
3447   section = 1 ,
3448   subsection = 2 ,
3449   subsubsection = 3 ,
3450   paragraph = 4 ,
3451   subparagraph = 5
3452 }

```

14.8.7 标签引用数字的汉化

`\refstepcounter` 对标题进行引用时,设置标签为通过 number 选项设置的形式。

```

3453 \cs_new_protected:Npn \CTEX@setcurrentlabel@n #1
3454 {
3455   \protected@edef \@currentlabel
3456   {
3457     \cs_if_exist:CTF { CTEX@the#1 }
3458     { \exp_args:cc { p@#1 } { CTEX@the#1 } }

```

```

3459     { \exp_not:o { \@currentlabel } }
3460   }
3461 }

```

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

```

3462 \cs_new_protected:Npn \ctex_varioref_hook:
3463 {
3464   \seq_map_inline:Nn \c__ctex_headings_seq
3465     { \ctex_fix_varioref_label:n { ##1 } }
3466   \ctex_at_end_package:nn { cleveref } { \ctex_cleveref_hook: }
3467 }

```

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

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

它的第二个参数必须放在花括号内,否则将会被作为宏的定界符号。

```

3468 \cs_new_protected:Npn \ctex_fix_varioref_label:n #1
3469 {
3470   \tl_if_empty:cT { p@#1 }
3471   { \exp_args:Nnx \labelformat {#1} { \exp_not:c { CTEX@the#1 } } }
3472 }

```

`\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`

```

3473 \cs_new_protected:Npn \ctex_cleveref_hook:
3474 {
3475   \@ifpackageloaded { hyperref }
3476   { \__ctex_cleveref_hook_aux:N \H@refstepcounter }
3477   {
3478     \__ctex_cleveref_hook_aux:N \refstepcounter@noarg
3479     \__ctex_cleveref_hook_aux:N \refstepcounter@optarg
3480   }
3481   \__ctex_cleveref_hook_aux:N \appendix
3482 }
3483 \cs_new_protected:Npn \__ctex_cleveref_hook_aux:N #1
3484 {
3485   \ctex_patch_cmd_all:NnnnTF #1
3486   {
3487     \ExplSyntaxOff
3488     \char_set_catcode_letter:n { 64 }
3489   }
3490   { \endcsname \csname the }
3491   { \expandafter \endcsname \csname the }
3492   { }
3493   { \ctex_patch_failure:N #1 }
3494 }

```

如果 `varioref` 已经被载入,则使用它来设置。

```

3495 \cs_if_exist:NTF \labelformat
3496 { \ctex_varioref_hook: }
3497 {
3498   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3499   \RenewDocumentCommand \refstepcounter { m }
3500   {

```



```

3501         \CTEX@save@refstepcounter {#1}
3502         \CTEX@setcurrentlabel@n {#1}
3503     }
3504     \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3505 }

```

14.8.8 载入 $\langle scheme \rangle$ 文件

```

3506 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3507 </class|heading>

```

14.8.9 标题格式的 scheme 定义

下面使用 CT_EX 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。

```

3508 <*scheme&(article|book|report|beamer)>

3509 \ctex_set:nn { part }
3510 {
3511     aftertitle = \par ,
3512     <*article|book|report>
3513     hang = false ,
3514     </article|book|report>
3515     <*plain>
3516     name = \partname \space ,
3517     <*article|book|report>
3518     number = \thepart ,
3519     </article|book|report>
3520     <*beamer>
3521     number = \insertromanpartnumber ,
3522     </beamer>
3523     </plain>
3524     <*chinese>
3525     number = \chinese { part } ,
3526     </chinese>
3527     <*article>
3528     beforeskip = 4ex ,
3529     afterskip = 3ex ,
3530     <*plain>
3531     format = \raggedright ,
3532     nameformat = \Large \bfseries ,
3533     aftername = \par \nobreak ,
3534     titleformat = \huge \bfseries ,
3535     afterindent = false
3536     </plain>
3537     <*chinese>
3538     format = \Large \bfseries \centering ,
3539     aftername = \quad ,
3540     afterindent = true
3541     </chinese>
3542     </article>
3543     <*book|report>
3544     aftername = \par \vskip 20 \p@ ,
3545     beforeskip = 0pt \@plus 1fil ,
3546     afterskip = 0pt \@plus 1fil ,
3547     pagestyle = plain ,
3548     break = \if@openright \cleardoublepage \else \clearpage \fi ,
3549     <*plain>
3550     format = \centering ,
3551     nameformat = \huge \bfseries ,
3552     titleformat = \Huge \bfseries
3553     </plain>
3554     <*chinese>
3555     format = \huge \bfseries \centering
3556     </chinese>

```

```

3557 </book|report>
3558 <*beamer>
3559     format      = \centering ,
3560     nameformat   = \usebeamerfont { part ~ name }
3561                 \usebeamerfont { fg } { part ~ name } ,
3562     aftername    = \vskip 1em \par ,
3563     titleformat  = \usebeamerfont { part ~ title }
3564 </beamer>
3565 }

3566 <*book|report>
3567 \ctex_set:nn { chapter }
3568 {
3569     pagestyle    = plain ,
3570     aftertitle   = \par ,
3571     hang         = false ,
3572     beforekip    = 50 \p@ ,
3573     afterkip     = 40 \p@ ,
3574     lofskip      = 10 \p@ ,
3575     lotskip      = 10 \p@ ,
3576     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3577 <*plain>
3578     name         = \chaptername \space ,
3579     number       = \thechapter ,
3580     format       = \raggedright ,
3581     nameformat   = \huge \bfseries ,
3582     aftername    = \par \nobreak \vskip 20 \p@ ,
3583     titleformat  = \Huge \bfseries ,
3584     afterindent  = false ,
3585     tocline      = \CTEXnumberline {#1} #2
3586 </plain>
3587 <*chinese>
3588     number       = \chinese { chapter } ,
3589     format       = \huge \bfseries \centering ,
3590     aftername    = \quad ,
3591     afterindent  = true
3592 </chinese>
3593 }
3594 </book|report>

3595 \ctex_set:nn { section }
3596 {
3597 <*article|book|report>
3598     number       = \thesection ,
3599     aftername     = \quad ,
3600     aftertitle    = \@@par ,
3601     beforekip     = 3.5ex \@plus 1ex \@minus .2ex ,
3602     afterkip      = 2.3ex \@plus .2ex ,
3603     runin         = false ,
3604     break        = \addpenalty \@secpenalty ,
3605 <*plain>
3606     format       = \Large \bfseries ,
3607     afterindent  = false
3608 </plain>
3609 <*chinese>
3610     format       = \Large \bfseries \centering ,
3611     afterindent  = true
3612 </chinese>
3613 </article|book|report>
3614 <*beamer>
3615 <*plain>
3616     name         = \sectionname \space ,
3617 </plain>
3618     format       = \centering ,
3619     number       = \insertsectionnumber ,
3620     nameformat   = \usebeamerfont { section ~ name }
3621                 \usebeamerfont { fg } { section ~ name } ,
3622     aftername    = \vskip 1em \par ,

```

```

3623     titleformat = \usebeamerfont { section ~ title } ,
3624     aftertitle   = \par
3625 </beamer>
3626 }

3627 \ctex_set:nn { subsection }
3628 {
3629 <*article|book|report>
3630     number       = \thesubsection ,
3631     format        = \large \bfseries ,
3632     aftername     = \quad ,
3633     aftertitle    = \@@par ,
3634     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3635     afterskip     = 1.5ex \@plus .2ex ,
3636     runin         = false ,
3637     break         = \addpenalty \@secpenalty ,
3638 <*plain>
3639     afterindent   = false
3640 </plain>
3641 <*chinese>
3642     afterindent   = true
3643 </chinese>
3644 </article|book|report>
3645 <*beamer>
3646 <*plain>
3647     name          = \subsectionname \space ,
3648     number        = \insertsubsectionnumber ,
3649 </plain>
3650 <*chinese>
3651     number        = \arabic { section } . \arabic { subsection } ,
3652 </chinese>
3653     format        = \centering ,
3654     nameformat    = \usebeamerfont { subsection ~ name }
3655                   \usebeamercolor [fg] { subsection ~ name } ,
3656     aftername     = \vskip 1em \par ,
3657     titleformat   = \usebeamerfont { subsection ~ title } ,
3658     aftertitle    = \par
3659 </beamer>
3660 }

3661 <*article|book|report>

3662 \ctex_set:nn { subsubsection }
3663 {
3664     number       = \thesubsubsection ,
3665     format        = \normalsize \bfseries ,
3666     aftername     = \quad ,
3667     aftertitle    = \@@par ,
3668     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3669     afterskip     = 1.5ex \@plus .2ex ,
3670     runin         = false ,
3671     break         = \addpenalty \@secpenalty ,
3672 <*plain>
3673     afterindent   = false
3674 </plain>
3675 <*chinese>
3676     afterindent   = true
3677 </chinese>
3678 }

3679 \ctex_set:nn { paragraph }
3680 {
3681     number       = \theparagraph ,
3682     format        = \normalsize \bfseries ,
3683     aftername     = \quad ,
3684     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3685     break         = \addpenalty \@secpenalty ,
3686 <*plain>
3687     afterindent   = false

```

```

3688 </plain>
3689 <*chinese>
3690     afterindent = true
3691 </chinese>
3692 }

3693 \ctex_set:nn { subparagraph }
3694 {
3695     number      = \thesubparagraph ,
3696     format      = \normalsize \bfseries ,
3697     aftername    = \quad ,
3698     beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
3699     break       = \addpenalty \@secpenalty ,
3700 <*plain>
3701     afterindent = false
3702 </plain>
3703 <*chinese>
3704     afterindent = true
3705 </chinese>
3706 }

```

处理 sub3section 与 sub4section 的格式。

```

3707 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3708 {
3709     \ctex_set:nn { paragraph }
3710     {
3711         aftertitle = \@@par ,
3712         afterskip  = 1ex \@plus .2ex ,
3713         runin      = false
3714     }
3715 }
3716 {
3717     \ctex_set:nn { paragraph }
3718     {
3719         afterskip  = 1em ,
3720         runin      = true
3721     }
3722 }
3723 \int_compare:nNnTF \g__ctex_section_depth_int > 3
3724 {
3725     \ctex_set:nn { subparagraph }
3726     {
3727         aftertitle = \@@par ,
3728         afterskip  = 1ex \@plus .2ex ,
3729         runin      = false
3730     }
3731 }
3732 {
3733     \ctex_set:nn { subparagraph }
3734     {
3735         afterskip  = 1em ,
3736         runin      = true
3737     }
3738 }
3739 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3740 { \ctex_set:nn { subparagraph } { indent = \c_zero_dim } }
3741 { \ctex_set:nn { subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

3742 \ctex_set:nn { appendix }
3743 <*article>
3744 { number      = \@Alph \c@section }
3745 </article>
3746 <*book|report>
3747 {
3748     name       = \appendixname \space ,

```

```

3749     number      = \@Alph \c@chapter
3750   }
3751 </book|report>

3752 </article|book|report>

3753 </scheme&(article|book|report|beamer)>

```

14.8.10 ctex.sty 的 heading 选项

```
3754 <*ctex|ctexheading>
```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3755 \clist_map_inline:nn { article , book , report , beamer }
3756 {
3757   \ifclassloaded {#1}
3758     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3759     { }
3760 }

```

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

```

3761 \msg_new:nnn { ctex } { not-standard-class }
3762 {
3763   None~of~the~standard~document~classes~was~loaded.\\
3764   Heading~`#1'~is~selected.\\
3765   ctex~may~not~work~as~expected.
3766 }
<ctex> 3767 \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 3768 \use:n
3769 {
3770   \tl_if_exist:NTF \c__ctex_std_class_tl
3771     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3772     {
3773       \cs_if_exist:NTF \chapter
3774       {
3775         \cs_if_exist:NF \if@mainmatter
3776         { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3777         \tl_const:Nn \c__ctex_class_tl { book }
3778       }
3779       { \tl_const:Nn \c__ctex_class_tl { article } }
3780       \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3781     }
3782     \ctex_file_input:n { ctex-heading- \c__ctex_class_tl .def }
3783   }
<ctex> 3784 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }

3785 </ctex|ctexheading>

```

14.8.11 标题配置文件

```

3786 <*name>
3787 \keys_set_known:nn { ctex }
3788 {
3789   contentsname = 目录 ,
3790   listfigurename = 插图 ,
3791   listtablename = 表格 ,
3792   figurename = 图 ,
3793   tablename = 表 ,
3794   abstractname = 摘要 ,
3795   indexname = 索引 ,
3796   bibname = 参考文献 ,
3797   appendixname = 附录 ,
3798   proofname = 证明 ,

```

```

3799     algorithmname = 算法 ,
3800     refname       = 参考文献 ,
3801     continuation  = (续) ,
3802     part / name = { 第 , 部分 } ,
3803     chapter / name = { 第 , 章 }
3804 }
3805 </name>

```

14.9 chinese 方案的其他设置

```
3806 <*scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
<article|book|report> 3807 \pagestyle { headings }
```

日期格式。

```
3808 \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。

```

3809 \ctex_if_autoindent_touched:F
3810 { \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。

```

3811 <*!generic>
3812 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3813 <*beamer>
3814 {
3815   \uselanguage { ChineseGBK }
3816   \languagealias { chinese } { ChineseGBK }
3817   \ctex_file_input:n { ctex-name-gbk.cfg }
3818 }
3819 {
3820   \uselanguage { ChineseUTF8 }
3821   \languagealias { chinese } { ChineseUTF8 }
3822   \ctex_file_input:n { ctex-name-utf8.cfg }
3823 }

```

让 translator 包优先查找中文翻译。

```

3824 \clist_put_left:Nn \trans@languagepath { chinese }
3825 </beamer>
3826 <*!beamer>
3827 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3828 { \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。

```

3829 \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3830 { \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。

```

3831 \fp_if_nan:nT { \l__ctex_line_spread_fp }
3832 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3833 </!beamer>
3834 </!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。

```

3835 <*generic>
3836 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3837 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3838 {
3839   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3840   {

```

```

3841     \uselanguage { ChineseGBK }
3842     \languagealias { chinese } { ChineseGBK }
3843     \ctex_file_input:n { ctex-name-gbk.cfg }
3844   }
3845   {
3846     \uselanguage { ChineseUTF8 }
3847     \languagealias { chinese } { ChineseUTF8 }
3848     \ctex_file_input:n { ctex-name-utf8.cfg }
3849   }
3850   \clist_put_left:Nn \trans@languagepath { chinese }
3851 }
3852 {
3853   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3854   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3855   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3856   \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3857   { \int_gset:Nn \g__ctex_font_size_int { 0 } }
3858   \fp_if_nan:nT { \l__ctex_line_spread_fp }
3859   { \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。

```

3860   \tl_if_exist:NT \c__ctex_std_class_tl
3861   { \RequirePackage { indentfirst } }
3862 }
3863 </generic>
3864 </scheme&chinese>

```

14.10 中文字号

```

3865 <*class|ctex|ctexsize>

```

```

\zihao 3866 \NewDocumentCommand \zihao { m }
3867   { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

\ctex_zihao:n 3868 \cs_new_protected:Npn \ctex_zihao:n #1
3869   {
3870     \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3871     { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3872     { \msg_error:nnn { ctex } { fontsize } {#1} }
3873   }
3874   \msg_new:nnnn { ctex } { fontsize }
3875   { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3876   {
3877     The~old~font~size~is~used~if~you~continue.\\
3878     The~available~font~sizes~are~listed~as~follow.\\
3879     \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3880   }

```

14.10.1 定义中文字号

基础行距是字号的 1.2 倍, 采用 ε -TeX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确²⁰。

```

3881 \prop_new:N \c__ctex_font_size_prop
3882 \seq_new:N \c__ctex_font_size_seq
3883 \cs_new_protected:Npn \__ctex_save_font_size:nn #1#2
3884   {
3885     \use:x
3886     {
3887       \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3888       {
3889         { \dim_to_decimal:n {#2} }
3890         { \dim_to_decimal:n { (#2) * 6 / 5 } }

```

²⁰<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

3891     }
3892   }
3893   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3894 }
3895 \clist_map_inline:nn
3896 {
3897   { 8 } { 5 bp } ,
3898   { 7 } { 5.5 bp } ,
3899   { -6 } { 6.5 bp } ,
3900   { 6 } { 7.5 bp } ,
3901   { -5 } { 9 bp } ,
3902   { 5 } { 10.5 bp } ,
3903   { -4 } { 12 bp } ,
3904   { 4 } { 14 bp } ,
3905   { -3 } { 15 bp } ,
3906   { 3 } { 16 bp } ,
3907   { -2 } { 18 bp } ,
3908   { 2 } { 22 bp } ,
3909   { -1 } { 24 bp } ,
3910   { 1 } { 26 bp } ,
3911   { -0 } { 36 bp } ,
3912   { 0 } { 42 bp }
3913 }
3914 { \__ctex_save_font_size:nn #1 }

\ctex_declare_math_sizes:nmmn 3915 \cs_new_protected:Npn \ctex_declare_math_sizes:nmmn #1#2#3#4
3916 {
3917   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3918   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3919 }

\__ctex_get_font_sizes:Nn 3920 \cs_new_protected:Npn \__ctex_get_font_sizes:Nn #1#2
3921 {
3922   \tl_clear:N #1
3923   \tl_map_inline:nn {#2}
3924   {
3925     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3926     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3927     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3928   }
3929 }

3930 \clist_map_inline:nn
3931 {
3932   { 8 } { 8 } { 5pt } { 5pt } ,
3933   { 7 } { 7 } { 5pt } { 5pt } ,
3934   { -6 } { -6 } { 5pt } { 5pt } ,
3935   { 6 } { 6 } { 5pt } { 5pt } ,
3936   { -5 } { -5 } { 6pt } { 5pt } ,
3937   { 5 } { 5 } { 7pt } { 5pt } ,
3938   { -4 } { -4 } { 8pt } { 6pt } ,
3939   { 4 } { 4 } { 5 } { 6 } ,
3940   { -3 } { -3 } { -4 } { -5 } ,
3941   { 3 } { 3 } { 4 } { 5 } ,
3942   { -2 } { -2 } { -3 } { -4 } ,
3943   { 2 } { 2 } { 3 } { 4 } ,
3944   { -1 } { -1 } { -2 } { -3 } ,
3945   { 1 } { 1 } { 2 } { 3 } ,
3946   { -0 } { -0 } { -1 } { -2 } ,
3947   { 0 } { 0 } { 1 } { 2 }
3948 }
3949 { \ctex_declare_math_sizes:nmmn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3950 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3

```



```

3951 {
3952   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3953   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3954   { \msg_error:nnn { ctex } { fontsize } {#2} }
3955 }
3956 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3957 { \cs_set_protected:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

3958 \int_case:nn { \g__ctex_font_size_int }
3959 {
3960   { 0 } { \ctex_file_input:n { ctex-c5size.clo } }
3961   { 1 } { \ctex_file_input:n { ctex-cs4size.clo } }
3962 }

3963 </class>ctex<ctexsize>

3964 <*c5size>
3965 \ctex_set_font_size:Nnn \normalsize { 5 }
3966 {
3967   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3968   \abovedisplayshortskip \z@ \@plus3\p@
3969   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3970   \belowdisplayskip \abovedisplayskip
3971   \let\@listi\@listI
3972 }
3973 \ctex_set_font_size:Nnn \small { -5 }
3974 {
3975   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3976   \abovedisplayshortskip \z@ \@plus2\p@
3977   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3978   \def\@listi{\leftmargin\leftmargini
3979             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3980             \parsep 2\p@ \@plus\p@ \@minus\p@
3981             \itemsep \parsep}
3982   \belowdisplayskip \abovedisplayskip
3983 }
3984 \ctex_set_font_size:Nnn \footnotesize { 6 }
3985 {
3986   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3987   \abovedisplayshortskip \z@ \@plus\p@
3988   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3989   \def\@listi{\leftmargin\leftmargini
3990             \topsep 3\p@ \@plus\p@ \@minus\p@
3991             \parsep 2\p@ \@plus\p@ \@minus\p@
3992             \itemsep \parsep}
3993   \belowdisplayskip \abovedisplayskip
3994 }
3995 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3996 \ctex_set_font_size:Nnn \tiny { 7 } { }
3997 \ctex_set_font_size:Nnn \large { -4 } { }
3998 \ctex_set_font_size:Nnn \Large { -3 } { }
3999 \ctex_set_font_size:Nnn \LARGE { -2 } { }
4000 \ctex_set_font_size:Nnn \huge { 2 } { }
4001 \ctex_set_font_size:Nnn \Huge { 1 } { }
4002 </c5size>
4003 <*cs4size>
4004 \ctex_set_font_size:Nnn \normalsize { -4 }
4005 {
4006   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
4007   \abovedisplayshortskip \z@ \@plus3\p@
4008   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
4009   \belowdisplayskip \abovedisplayskip
4010   \let\@listi\@listI
4011 }
4012 \ctex_set_font_size:Nnn \small { 5 }
4013 {
4014   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
4015   \abovedisplayshortskip \z@ \@plus3\p@
4016   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@

```

```

4017 \def\@listi{\leftmargin\leftmargin
4018 \topsep 9\p@ \@plus3\p@ \@minus5\p@
4019 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
4020 \itemsep \parsep}
4021 \belowdisplayskip \abovedisplayskip
4022 }
4023 \ctex_set_font_size:Nnn \footnotesize { -5 }
4024 {
4025 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
4026 \abovedisplayshortskip \z@ \@plus3\p@
4027 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
4028 \def\@listi{\leftmargin\leftmargin
4029 \topsep 6\p@ \@plus2\p@ \@minus2\p@
4030 \parsep 3\p@ \@plus2\p@ \@minus\p@
4031 \itemsep \parsep}
4032 \belowdisplayskip \abovedisplayskip
4033 }
4034 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
4035 \ctex_set_font_size:Nnn \tiny { -6 } { }
4036 \ctex_set_font_size:Nnn \large { -3 } { }
4037 \ctex_set_font_size:Nnn \Large { -2 } { }
4038 \ctex_set_font_size:Nnn \LARGE { 2 } { }
4039 \ctex_set_font_size:Nnn \huge { -1 } { }
4040 \ctex_set_font_size:Nnn \Huge { 1 } { }
4041 </cs4size>
<ctexsize> 4042 \normalsize
4043 <*class|ctex>

```

14.11 更新行距

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

```

4044 \fp_if_nan:nF { \l__ctex_line_spread_fp }
4045 {
4046 \exp_args:Nx \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

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

```

4047 \group_begin: \footnotesize \exp_args:NNNo \group_end:
4048 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
4049 }

```

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

```

4050 \normalsize

```

14.12 其他功能

CTeX `ctex-faq.sty` 中的定义是

```
\DeclareRobustCommand\CTeX{\mathbb{C}}\kern-.05em\TeX}
```

然而 `\mathbb` 未必有定义，这里就不采用它了，只定义最简单的形式。**CTeX** 可以直接用在 PDF 书签中。

```

4051 \NewDocumentCommand \CTeX { }
4052 { C \TeX }
4053 \ctex_at_end_package:nn { hyperref }
4054 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

`captiondelimit` 过时选项。

```

4055 \ctex_define:n
4056 {
4057     captiondelimit .code:n =
4058     {
4059         \ctex_deprecated_option:n
4060         { You~can~load~the~package~`caption'~to~get~its~functionality. }
4061     }
4062 }
4063 </class|ctex>

```

14.12.1 列表环境的缩进

`\verse` 和 `\quotation` 只在使用文档类的时候修改诗歌和引用环境的缩进。

```

4064 <*scheme&chinese&(article|book|report)>
4065 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
4066 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
4067 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
4068 </scheme&chinese&(article|book|report)>
4069 <*class|ctex>

```

14.13 载入中文字库

`\ctex_fontset_error:n` 字库不可用时给出紧急错误信息, 停止读取定义文件。

```

4070 \cs_new_protected:Npn \ctex_fontset_error:n #1
4071 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
4072 \msg_new:nnn { ctex } { fontset-unavailable }
4073 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }

```

`\ctex_fontset_case:nnn` 3 个参数依次为 pdfTeX、upTeX 和 XeTeX/LuaTeX。

```

4074 \cs_new:Npx \ctex_fontset_case:nnn #1#2#3
4075 {
4076     \sys_if_engine_pdftex:TF
4077     {#1}
4078     { \sys_if_engine_uptex:TF {#2} {#3} }
4079 }

```

`\ctex_fontset_case:nnnn` 4 个参数依次为 pdfTeX(生成 PDF)、pdfTeX(生成 DVI)、upTeX 和 XeTeX/LuaTeX。

```

4080 \cs_new:Npx \ctex_fontset_case:nnnn #1#2#3#4
4081 {
4082     \sys_if_engine_pdftex:TF
4083     { \sys_if_output_pdf:TF {#1} {#2} }
4084     { \sys_if_engine_uptex:TF {#3} {#4} }
4085 }

```

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

```

4086 \cs_new_protected:Npn \ctex_detect_platform:
4087 {
4088     \sys_if_platform_windows:TF
4089     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
4090     {
4091         \ctex_if_platform_macos:TF
4092         { \tl_gset:Nn \g__ctex_fontset_tl { mac } }
4093         { \tl_gset:Nn \g__ctex_fontset_tl { fandol } }
4094     }
4095 }

```

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

```
4096 \cs_new_protected:Npn \ctex_if_platform_macos:TF
4097 { \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。

```
4098 \cs_new_protected:Npn \ctex_load_fontset:
4099 {
4100   \tl_if_empty:NTF \g__ctex_fontset_tl
4101   { \ctex_detect_platform: }
4102   {
4103     \bool_lazy_or:nnTF
4104     { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsnew } }
4105     { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsold } }
4106     {
4107       \msg_warning:nxxx { ctex } { deprecated-fontset }
4108       { \g__ctex_fontset_tl } { windows }
4109       \tl_gset:Nn \g__ctex_fontset_tl { windows }
4110     }
4111     {
4112       \file_if_exist:nF { ctex-fontset- \g__ctex_fontset_tl .def }
4113       {
4114         \use:x
4115         {
4116           \ctex_detect_platform:
4117           \msg_error:nxxx { ctex } { fontset-not-found }
4118           { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
4119         }
4120       }
4121     }
4122   }
4123   \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
4124 }
4125 \msg_new:nnn { ctex } { deprecated-fontset }
4126 { CTeX~fontset~`#1'~is~deprecated.\\ Fontset~`#2'~will~be~used~instead. }
4127 \msg_new:nnnn { ctex } { fontset-not-found }
4128 { CTeX~fontset~`#1'~could~not~be~found.\\ Fontset~`#2'~will~be~used~instead. }
4129 { You~may~run~\mktexlsr'~firstly. }
4130 \@onlypreamble \ctex_load_fontset:
```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```
4131 \ctex_define:n
4132 {
4133   fontset .code:n =
4134   {
4135     \ctex_if_preamble:TF
4136     {
4137       \str_if_eq:eeTF {#1} { none }
4138       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
4139       {
4140         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
4141         {
4142           \tl_gset:Nx \g__ctex_fontset_tl {#1}
4143           \ctex_load_fontset:
4144         }
4145         {
4146           \msg_error:nxxx { ctex } { fontset-loaded }
4147           { \g__ctex_fontset_tl } {#1}
4148         }
4149       }
4150     }
4151     { \msg_error:nn { ctex } { fontset-only-preamble } }
4152   }
4153 }
4154 \msg_new:nnnn { ctex } { fontset-loaded }
```

```

4155 {
4156   CTeX~fontset~`#1'~has~been~loaded.
4157   \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
4158 }
4159 { Only~one~fontset~can~be~loaded~in~the~preamble. }
4160 \msg_new:nnn { ctex } { fontset-only-preamble }
4161 { The~fontset'~option~can~be~used~only~in~preamble. }

```

载入中文字库。

```

4162 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
4163 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

4164 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
4165 </class>ctex>
4166 <*config>
4167 %%
4168 </config>

```

14.14.2 ctexopts.cfg

这里仅为配置文件示例：使用 Windows 下的字体设置。

```

4169 <*ctexopts>
4170 %%
4171 %% \ctex_set:nn { option } { fontset = windows }
4172 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

4173 <*c19|c70>
4174 %%
4175 %% Chinese characters
4176 %%
<c19> 4177 %% character set: GBK (extension of GB 2312)
<c70> 4178 %% character set: Unicode
4179 %% font encoding: Unicode
4180 %%
4181 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm&c19> 4182 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm&c70> 4183 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf&c19> 4184 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 4185 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 4186 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt&c70> 4187 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

up \LaTeX 使用的字体族。up \LaTeX 在 NFSS 下使用字体编码 JY2 和 JT2 来分别表示横排与直排的日文。

```

<rm&jy2> 4188 \DeclareKanjiFamily{JY2}{zhrm}{ }
<rm&jt2> 4189 \DeclareKanjiFamily{JT2}{zhrm}{ }
<sf&jy2> 4190 \DeclareKanjiFamily{JY2}{zhsf}{ }
<sf&jt2> 4191 \DeclareKanjiFamily{JT2}{zhsf}{ }
<tt&jy2> 4192 \DeclareKanjiFamily{JY2}{zhtt}{ }
<tt&jt2> 4193 \DeclareKanjiFamily{JT2}{zhtt}{ }

```

```

4194 <*rm>
4195 <*c19>
4196 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
4197 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
4198 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
4199 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
4200 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4201 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4202 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
4203 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
4204 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
4205 </c19>
4206 <*c70>
4207 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
4208 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
4209 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
4210 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
4211 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4212 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4213 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
4214 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
4215 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
4216 </c70>
4217 <*jy2>
4218 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{\}
4219 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{\}
4220 \DeclareFontShape{JY2}{zhrm}{b}{n}{<-> upzhserifb-h}{\}
4221 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{\}
4222 </jy2>
4223 <*jt2>
4224 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{\}
4225 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{\}
4226 \DeclareFontShape{JT2}{zhrm}{b}{n}{<-> upzhserifb-v}{\}
4227 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{\}
4228 </jt2>
4229 </rm>
4230 <*sf>
4231 <*c19>
4232 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4233 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4234 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4235 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
4236 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4237 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4238 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4239 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4240 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4241 </c19>
4242 <*c70>
4243 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyousl}{\CJKnormal}
4244 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyousl}{\CJKbold}
4245 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyousl}{\CJKbold}
4246 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4247 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4248 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4249 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyousl}{\CJKnormal}
4250 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyousl}{\CJKbold}
4251 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyousl}{\CJKbold}
4252 </c70>
4253 <*jy2>
4254 \DeclareFontShape{JY2}{zhfs}{m}{n}{<-> upzhsans-h}{\}
4255 \DeclareFontShape{JY2}{zhfs}{b}{n}{<-> upzhsansb-h}{\}
4256 \DeclareFontShape{JY2}{zhfs}{bx}{n}{<-> upzhsansb-h}{\}
4257 </jy2>
4258 <*jt2>
4259 \DeclareFontShape{JT2}{zhfs}{m}{n}{<-> upzhsans-v}{\}
4260 \DeclareFontShape{JT2}{zhfs}{b}{n}{<-> upzhsansb-v}{\}

```

```

4261 \DeclareFontShape{JT2}{zhsf}{bx}{n}{<-> upzhsansb-v}{}
4262 </jt2>
4263 </sf>

4264 <*tt>
4265 <*c19>
4266 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4267 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4268 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4269 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4270 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4271 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4272 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4273 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4274 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4275 </c19>
4276 <*c70>
4277 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4278 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4279 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4280 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4281 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4282 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4283 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4284 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4285 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4286 </c70>
4287 <*jy2>
4288 \DeclareFontShape{JY2}{zhht}{m}{n}{<-> upzhmono-h}{}
4289 </jy2>
4290 <*jt2>
4291 \DeclareFontShape{JT2}{zhht}{m}{n}{<-> upzhmono-v}{}
4292 </jt2>
4293 </tt>

```

14.15.2 预定义字库

```
4294 <*fontset>
```

14.15.2.1 adobe

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

```

4295 <*adobe>
4296 \ctex_fontset_case:nnnn
4297 { \ctex_fontset_error:n { adobe } }
4298 {
4299   \ctex_zhmap_case:nnn
4300   {
4301     \setCJKmainfont { AdobeSongStd-Light.otf }
4302     [
4303       cmap          = UniGB-UTF16-H,
4304       BoldFont      = AdobeHeitiStd-Regular.otf,
4305       ItalicFont    = AdobeKaitiStd-Regular.otf
4306     ]
4307     \setCJKsansfont { AdobeHeitiStd-Regular.otf }
4308     [ cmap = UniGB-UTF16-H ]
4309     \setCJKmonofont { AdobeFangsongStd-Regular.otf }
4310     [ cmap = UniGB-UTF16-H ]
4311     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light.otf }
4312     [ cmap = UniGB-UTF16-H ]
4313     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular.otf }
4314     [ cmap = UniGB-UTF16-H ]
4315     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular.otf }
4316     [ cmap = UniGB-UTF16-H ]
4317     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular.otf }
4318     [ cmap = UniGB-UTF16-H ]

```

```

4319     \ctex_punct_set:n { adobe }
4320     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4321     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4322     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4323     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4324     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4325   }
4326   {
4327     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { adobe }
4328     \ctex_punct_set:n { adobe }
4329     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4330     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4331     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4332   }
4333   { \ctex_fontset_error:n { adobe } }
4334 }
4335 {
4336   \ctex_set_upfonts:nnnnnn
4337   { AdobeSongStd-Light.otf }
4338   { AdobeHeitiStd-Regular.otf }
4339   { AdobeKaitiStd-Regular.otf }
4340   { AdobeHeitiStd-Regular.otf }
4341   { AdobeHeitiStd-Regular.otf }
4342   { AdobeFangsongStd-Regular.otf }
4343   \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4344   \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4345   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4346   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4347 }
4348 {
4349   \setCJKmainfont { AdobeSongStd-Light }
4350   [ BoldFont = AdobeHeitiStd-Regular, ItalicFont = AdobeKaitiStd-Regular ]
4351   \setCJKsansfont { AdobeHeitiStd-Regular }
4352   \setCJKmonofont { AdobeFangsongStd-Regular }
4353   \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4354   \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4355   \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4356   \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4357 }
4358 </adobe>

```

14.15.2.2 fandol

```

4359 <*fandol>
4360 \ctex_fontset_case:nnnn
4361 { \ctex_fontset_error:n { fandol } }
4362 {
4363   \ctex_zhmap_case:nnn
4364   {
4365     \setCJKmainfont { FandolSong-Regular.otf }
4366     [
4367       cmap = UniGB-UTF16-H,
4368       BoldFont = FandolSong-Bold.otf,
4369       ItalicFont = FandolKai-Regular.otf
4370     ]
4371     \setCJKsansfont { FandolHei-Regular.otf }
4372     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4373     \setCJKmonofont { FandolFang-Regular.otf }
4374     [ cmap = UniGB-UTF16-H ]
4375     \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf }
4376     [ cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf ]
4377     \setCJKfamilyfont { zhhei } { FandolHei-Regular.otf }
4378     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4379     \setCJKfamilyfont { zhfs } { FandolFang-Regular.otf }
4380     [ cmap = UniGB-UTF16-H ]
4381     \setCJKfamilyfont { zhkai } { FandolKai-Regular.otf }

```



```

4382     [ cmap = UniGB-UTF16-H ]
4383     \ctex_punct_set:n { fandol }
4384     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4385     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4386     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4387     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4388     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4389     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4390   }
4391   {
4392     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { fandol }
4393     \ctex_punct_set:n { fandol }
4394     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4395     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4396     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4397   }
4398   { \ctex_fontset_error:n { fandol } }
4399 }
4400 {
4401   \ctex_set_upfonts:nnnnnn
4402   { FandolSong-Regular.otf }
4403   { FandolSong-Bold.otf }
4404   { FandolKai-Regular.otf }
4405   { FandolHei-Regular.otf }
4406   { FandolHei-Bold.otf }
4407   { FandolFang-Regular.otf }
4408   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4409   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4410   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4411   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4412 }
4413 {
4414   \setCJKmainfont { FandolSong-Regular }
4415   [
4416     Extension = .otf,
4417     BoldFont = FandolSong-Bold,
4418     ItalicFont = FandolKai-Regular
4419   ]
4420   \setCJKsansfont { FandolHei-Regular }
4421   [ Extension = .otf, BoldFont = FandolHei-Bold ]
4422   \setCJKmonofont { FandolFang-Regular }
4423   [ Extension = .otf ]
4424   \setCJKfamilyfont { zhsong } { FandolSong-Regular }
4425   [ Extension = .otf, BoldFont = FandolSong-Bold ]
4426   \setCJKfamilyfont { zhhei } { FandolHei-Regular }
4427   [ Extension = .otf, BoldFont = FandolHei-Bold ]
4428   \setCJKfamilyfont { zhfs } { FandolFang-Regular }
4429   [ Extension = .otf ]
4430   \setCJKfamilyfont { zhkai } { FandolKai-Regular }
4431   [ Extension = .otf ]
4432 }
4433 </fandol>

```

14.15.2.3 founder

```

4434 <*founder>
4435 \ctex_fontset_case:nnn
4436 {
4437   \ctex_zhmap_case:nnn
4438   {
4439     \setCJKmainfont { FZSSK.TTF }
4440     [ BoldFont = FZXBSK.TTF, ItalicFont = FZKTK.TTF ]
4441     \setCJKsansfont { FZXH1K.TTF } [ BoldFont = FZHTK.TTF ]
4442     \setCJKmonofont { FZFSK.TTF }
4443     \setCJKfamilyfont { zhsong } { FZSSK.TTF } [ BoldFont = FZXBSK.TTF ]
4444     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4445     \setCJKfamilyfont { zhkai } { FZKTK.TTF }

```

```

4446 \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4447 \setCJKfamilyfont { zhli } { FZLSK.TTF }
4448 \setCJKfamilyfont { zhyou } { FZY1K.TTF } [ BoldFont = FZY3K.TTF ]
4449 \ctex_punct_set:n { founder }
4450 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4451 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4452 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4453 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4454 \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4455 \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4456 \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4457 }
4458 {
4459 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { founder }
4460 \ctex_punct_set:n { founder }
4461 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4462 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4463 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4464 }
4465 { \ctex_fontset_error:n { founder } }
4466 }
4467 {
4468 \ctex_set_upfonts:nnnnnn
4469 { FZSSK.TTF }
4470 { FZXBSK.TTF }
4471 { FZKTK.TTF }
4472 { FZXH1K.TTF }
4473 { FZHTK.TTF }
4474 { FZFSK.TTF }
4475 \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4476 \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4477 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4478 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4479 \ctex_set_upfamily:nnn { zhli } { upschrm } {}
4480 \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4481 \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4482 \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4483 }
4484 {
4485 \setCJKmainfont { FZShuSong-Z01 }
4486 [ BoldFont = FZXiaoBiaoSong-B05, ItalicFont = FZKai-Z03 ]
4487 \setCJKsansfont { FZXiHeiI-Z08 } [ BoldFont = FZHei-B01 ]
4488 \setCJKmonofont { FZFangSong-Z02 }
4489 \setCJKfamilyfont { zhsong } { FZShuSong-Z01 }
4490 [ BoldFont = FZXiaoBiaoSong-B05 ]
4491 \setCJKfamilyfont { zhhei } { FZHei-B01 }
4492 \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4493 \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4494 \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4495 \setCJKfamilyfont { zhyou } { FZXiYuan-M01 }
4496 [ BoldFont = FZZhunYuan-M02 ]
4497 }
4498 </founder>

```

14.15.2.4 mac 相关

按 [Issue 351](#) 的讨论，以 El Capitan 为分界，分别设置 macold (El Capitan 之前) 和 macnew (El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。

```

4499 <*mac>
4500 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
4501 { \ctex_file_input:n { ctex-fontset-macnew.def } }
4502 { \ctex_file_input:n { ctex-fontset-macold.def } }
4503 </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。

```

4504 <*macold|macnew>
4505 \ctex_fontset_case:nnnn
4506 { \ctex_fontset_error:n { mac } }
4507 <*macold>
4508 { \ctex_fontset_error:n { macold } }
4509 { \ctex_fontset_error:n { macold } }
4510 </macold>
4511 <*macnew>
4512 {
4513   \ctex_zhmap_case:nnn
4514   {
4515     \setCJKmainfont { :3:Songti.ttc }
4516     [
4517       BoldFont      = :1:Songti.ttc,
4518       ItalicFont    = :0:Kaiti.ttc,
4519       BoldItalicFont = :3:Kaiti.ttc,
4520     ]
4521     \setCJKsansfont { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
4522     \setCJKmonofont { STFANGSO.ttf }
4523     \setCJKfamilyfont { zhsong } { :3:Songti.ttc } [ BoldFont = :1:Songti.ttc ]
4524     \setCJKfamilyfont { zhhei } { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
4525     \setCJKfamilyfont { zhkai } { :0:Kaiti.ttc } [ BoldFont = :3:Kaiti.ttc ]
4526     \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
4527     \setCJKfamilyfont { zhli } { :0:Baoli.ttc }
4528     \setCJKfamilyfont { zhyou } { :4:Yuanti.ttc } [ BoldFont = :0:Yuanti.ttc ]
4529     \ctex_punct_set:n { mac }
4530     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4531     \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
4532     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4533     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4534     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4535     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhpfb }
4536     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4537   }
4538   {
4539     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { mac }
4540     \ctex_punct_set:n { mac }
4541     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4542     \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
4543     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhpfb }
4544     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4545   }
4546   { \ctex_fontset_error:n { macnew } }
4547 }
4548 {
4549   \ctex_set_upmap:nnn { upserif } { :3:Songti.ttc } { :1:Songti.ttc }
4550   \ctex_set_upmap:nnn { upserifit } { :0:Kaiti.ttc } { }
4551   \ctex_set_upmap:nnn { upstsl } { :0:Baoli.ttc } { }
4552   \ctex_set_upmap:nnn { upstht } { :4:Yuanti.ttc } { }
4553   \ctex_set_upmap_unicode:nnn { upsans } { :2:PingFang.ttc } { :8:PingFang.ttc }
4554   \ctex_set_upmap_unicode:nnn { upmono } { STFANGSO.ttf } { }
4555   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4556   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4557   \ctex_set_upfamily:nnn { zhfs } { upzhmono } { }
4558   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
4559   \ctex_set_upfamily:nnn { zhli } { upschrm } { }
4560   \ctex_set_upfamily:nnn { zhyou } { upschgt } { }
4561 }
4562 </macnew>
4563 {
4564 <*macold>

```

```

4565 \setCJKmainfont { STSong }
4566 [ BoldFont = STHeiti, ItalicFont = STKaiti ]
4567 \setCJKsansfont { STXihei } [ BoldFont = STHeiti ]
4568 \setCJKmonofont { STFangsong }
4569 \setCJKfamilyfont { zhsong } { STSong }
4570 \setCJKfamilyfont { zhhei } { STHeiti }
4571 \setCJKfamilyfont { zhfs } { STFangsong }
4572 \setCJKfamilyfont { zhkai } { STKaiti }
4573 </macold>
4574 <*macnew>
4575 \setCJKmainfont { Songti~SC~Light }
4576 [
4577     BoldFont = Songti~SC~Bold,
4578     ItalicFont = Kaiti~SC,
4579     BoldItalicFont = Kaiti~SC~Bold
4580 ]
4581 \setCJKsansfont { PingFang~SC }
4582 \setCJKmonofont { STFangsong }
4583 \setCJKfamilyfont { zhsong } { Songti~SC~Light } [ BoldFont = Songti~SC~Bold ]
4584 \setCJKfamilyfont { zhhei } { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ]
4585 \setCJKfamilyfont { zhpfs } { PingFang~SC }
4586 \setCJKfamilyfont { zhfs } { STFangsong }
4587 \setCJKfamilyfont { zhkai } { Kaiti~SC } [ BoldFont = Kaiti~SC~Bold ]
4588 \setCJKfamilyfont { zhli } { Baoli~SC }
4589 \setCJKfamilyfont { zhyou } { Yuanti~SC~Light } [ BoldFont = Yuanti~SC~Regular ]
4590 </macnew>
4591 }
4592 </macold|macnew>

```

14.15.2.5 ubuntu

```

4593 <*ubuntu>
4594 \ctex_fontset_case:nnnn
4595 { \ctex_fontset_error:n { ubuntu } }
4596 {
4597     \ctex_zhmap_case:nnn
4598     {
4599         \setCJKmainfont { :2:NotoSerifCJK-Regular.ttc }
4600         [ BoldFont = :2:NotoSerifCJK-Bold.ttc, ItalicFont = gkai00mp.ttf ]
4601         \setCJKsansfont { :2:NotoSansCJK-Regular.ttc }
4602         [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
4603         \setCJKmonofont { :2:NotoSerifCJK-Regular.ttc }
4604         [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
4605         \setCJKfamilyfont { zhsong } { :2:NotoSerifCJK-Regular.ttc }
4606         [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
4607         \setCJKfamilyfont { zhhei } { :2:NotoSansCJK-Regular.ttc }
4608         [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
4609         \setCJKfamilyfont { zhkai } { gkai00mp.ttf }
4610         \ctex_punct_set:n { ubuntu }
4611         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4612         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4613         \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4614         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4615         \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4616         \ctex_punct_map_bfseries:nn
4617         { \CJKrmdefault, \CJKttdefault, zhsong }
4618         { zhsongb }
4619     }
4620     {
4621         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { ubuntu }
4622         \ctex_punct_set:n { ubuntu }
4623         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4624         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4625         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4626     }
4627     { \ctex_fontset_error:n { ubuntu } }

```

```

4628 }
4629 {
4630   \ctex_set_upmap_unicode:nnn { upserif }
4631   { :2:NotoSerifCJK-Regular.ttc } { :2:NotoSerifCJK-Bold.ttc }
4632   \ctex_set_upmap_unicode:nnn { upsans }
4633   { :2:NotoSansCJK-Regular.ttc } { :2:NotoSansCJK-Bold.ttc }
4634   \ctex_set_upmap_unicode:nnn { upmono }
4635   { :2:NotoSerifCJK-Regular.ttc } { }
4636   \ctex_set_upmap:nnn { upserifit } { gkai00mp.ttf } { }
4637   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4638   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4639   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
4640 }
4641 {
4642   \setCJKmainfont { Noto~Serif~CJK~SC } [ ItalicFont = AR~PL~KaitiM~GB ]
4643   \setCJKsansfont { Noto~Sans~CJK~SC }
4644   \setCJKmonofont { Noto~Serif~CJK~SC }
4645   \setCJKfamilyfont { zhsong } { Noto~Serif~CJK~SC }
4646   \setCJKfamilyfont { zhhei } { Noto~Sans~CJK~SC }
4647   \setCJKfamilyfont { zhkai } { AR~PL~KaitiM~GB }
4648 }
4649 </ubuntu>

```

14.15.2.6 windows

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

```

4650 <*windows>
4651 \file_if_exist:nTF { \c_dollar_str WINDIR/Fonts/msyh.ttc }
4652 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
4653 {
4654   \file_if_exist:nTF { msyh.ttc }
4655   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
4656   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttf } }
4657 }
4658 \ctex_fontset_case:nnn
4659 {
4660   \ctex_zhmap_case:nnn
4661   {
4662     \ctex_punct_set:n { windows }
4663     \setCJKmainfont { simsun.ttc }
4664     [ BoldFont = simhei.ttf, ItalicFont = simkai.ttf ]
4665     \setCJKsansfont { msyh.\c__ctex_msyh_suffix_tl }
4666     [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
4667     \setCJKmonofont { simfang.ttf }
4668     \setCJKfamilyfont { zhsong } { simsun.ttc }
4669     \setCJKfamilyfont { zhhei } { simhei.ttf }
4670     \setCJKfamilyfont { zhfs } { simfang.ttf }
4671     \setCJKfamilyfont { zhkai } { simkai.ttf }
4672     \setCJKfamilyfont { zhyahei } { msyh.\c__ctex_msyh_suffix_tl }
4673     [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
4674     \setCJKfamilyfont { zhli } { simli.ttf }
4675     \setCJKfamilyfont { zhyou } { simyou.ttf }
4676     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4677     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4678     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4679     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4680     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhyahei } { zhyaheib }
4681     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4682   }
4683   {
4684     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { windows }
4685     \ctex_punct_set:n { windows }
4686     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4687     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4688     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4689   }

```

```

4690     { }
4691 }
4692 {
4693     \ctex_set_upfonts:nnnnnn
4694     { simsun.ttc }
4695     { simhei.ttf }
4696     { simkai.ttf }
4697     { msyh.\c__ctex_msyh_suffix_tl }
4698     { msyhbd.\c__ctex_msyh_suffix_tl }
4699     { simfang.ttf }
4700     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4701     \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4702     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4703     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4704     \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4705     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4706     \ctex_set_upfamily:nnn { zhyou } { upscht } {}
4707     \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4708     \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4709 }
4710 {
4711     \setCJKmainfont { SimSun } [ BoldFont = SimHei , ItalicFont = KaiTi ]
4712     \setCJKsansfont { MicrosoftYaHei } [ BoldFont = *~Bold ]
4713     \setCJKmonofont { FangSong }
4714     \setCJKfamilyfont { zhsong } { SimSun }
4715     \setCJKfamilyfont { zhhei } { SimHei }
4716     \setCJKfamilyfont { zhfs } { FangSong }
4717     \setCJKfamilyfont { zhkai } { KaiTi }
4718     \setCJKfamilyfont { zhyahei } { MicrosoftYaHei } [ BoldFont = *~Bold ]
4719     \setCJKfamilyfont { zhli } { LiSu }
4720     \setCJKfamilyfont { zhyou } { YouYuan }
4721 }
4722 </windows>

```

14.15.3 中文字体命令

\songti 使用 up^LA^TE_X 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。
 \heiti 同时 \yahei、\pingfang 命令被设置为与 \heiti 相同。

```

\fangsong
\kaishu 4723 <!*mac>
\lishu 4724 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
\youyuan 4725 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
\yahei 4726 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
\pingfang 4727 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4728 <*windows|founder|macnew>
4729 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4730 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4731 </windows|founder|macnew>
<windows> 4732 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4733 <*macnew>
4734 \bool_lazy_or:nnTF
4735 { \sys_if_engine_pdftex_p: }
4736 { \sys_if_engine_uptex_p: }
4737 {
4738     \cs_new_eq:NN \yahei \heiti
4739     \cs_new_eq:NN \pingfang \heiti
4740 }
4741 {
4742     \NewDocumentCommand \yahei { } { \CJKfamily { zhpj } }
4743     \NewDocumentCommand \pingfang { } { \CJKfamily { zhpj } }
4744 }
4745 </macnew>
4746 </!mac>
4747 </fontset>

```

14.15.4 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4748 <zhmap>
4749 \begingroup\catcode61\catcode48\catcode32=10\relax%
4750 \catcode 35=6 % #
4751 \catcode 45=12 % -
4752 \catcode123=1 % {
4753 \catcode125=2 % }
4754 \toks0{\endlinechar=\the\endlinechar\relax}%
4755 \toks2{\endlinechar=-1}%
4756 \def\x#1 #2 {%
4757   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4758   \toks2\expandafter{\the\toks2 \catcode#1=#2}%
4759   \x 13 5 % carriage return
4760   \x 32 10 % space
4761   \x 35 6 % #
4762   \x 40 12 % (
4763   \x 41 12 % )
4764   \x 45 12 % -
4765   \x 46 12 % .
4766   \x 47 12 % /
4767   \x 58 12 % :
4768   \x 60 12 % <
4769   \x 61 12 % =
4770   \x 64 11 % @
4771   \x 91 12 % [
4772   \x 93 12 % ]
4773   \x 123 1 % {
4774   \x 125 2 % }
4775 \edef\x#1{\endgroup%
4776   \edef\noexpand#1{%
4777     \the\toks0 %
4778     \let\noexpand\noexpand\noexpand#1%
4779     \noexpand\noexpand\noexpand\undefined%
4780     \noexpand\noexpand\noexpand\endinput}%
4781     \the\toks2}%
4782 \expandafter\x\csname ctex@zhmap@endinput\endcsname

\ifzhmappdf 4783 \begingroup\expandafter\endgroup
4784 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4785 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4786 \expandafter\ifx\csname pdfoutput\endcsname\relax
4787   iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4788 \else ifctexpdf\fi
4789 \endcsname

\ProvidesFile 提供非 LATEX 格式下的 \ProvidesFile。
4790 \begingroup
4791 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4792 \long\def\x#1\ProvidesFile#2[#3]{%
4793   #1%
4794   \immediate\write-1{File: #2 #3}%
4795   \expandafter\xdef\csname ver@#2\endcsname{#3}}
4796 \expandafter\x%
4797 \fi
4798 \endgroup

```

文件标识信息。

```

<adobe> 4799 \ProvidesFile{ctex-zhmap-adobe.tex}%
<adobe> 4800 [2020/08/23 v2.5.4 Adobe font map loader for DVIPDFMx (CTEX)]
<fandol> 4801 \ProvidesFile{ctex-zhmap-fandol.tex}%
<fandol> 4802 [2020/08/23 v2.5.4 Fandol font map loader for DVIPDFMx (CTEX)]
<founder> 4803 \ProvidesFile{ctex-zhmap-founder.tex}%

```



```

<founder> 4804 [2020/08/23 v2.5.4 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
<mac> 4805 \ProvidesFile{ctex-zhmap-mac.tex}%
<mac> 4806 [2020/08/23 v2.5.4 Mac font map loader for DVIPDFMx (CTEX)]
<ubuntu> 4807 \ProvidesFile{ctex-zhmap-ubuntu.tex}%
<ubuntu> 4808 [2020/08/23 v2.5.4 Ubuntu font map loader for DVIPDFMx (CTEX)]
<windows> 4809 \ProvidesFile{ctex-zhmap-windows.tex}%
<windows> 4810 [2020/08/23 v2.5.4 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]

```

14.15.4.1 ctex-zhmap-adobe.tex

```

4811 <*adobe>
4812 \ifzhmappdf
4813 %% pdfTeX does not support OTF fonts
4814 \else
4815 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4816 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4817 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4818 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4819 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4820 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4821 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4822 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4823 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4824 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4825 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4826 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4827 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4828 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4829 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4830 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4831 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4832 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4833 \fi
4834 </adobe>

```

14.15.4.2 ctex-zhmap-fandol.tex

```

4835 <*fandol>
4836 \ifzhmappdf
4837 %% pdfTeX does not support OTF fonts
4838 \else
4839 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4840 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4841 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4842 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4843 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4844 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4845 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4846 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4847 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4848 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4849 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4850 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4851 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4852 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4853 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4854 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4855 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4856 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4857 \fi
4858 </fandol>

```

14.15.4.3 ctex-zhmap-founder.tex

```

4859 <*founder>
4860 \ifzhmappdf
4861 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF>}
4862 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF>}

```



```

4863 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4864 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4865 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4866 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4867 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4868 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4869 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4870 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4871 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4872 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4873 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4874 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4875 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4876 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4877 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4878 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4879 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4880 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4881 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4882 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4883 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4884 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4885 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4886 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4887 \else
4888 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4889 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4890 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4891 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4892 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4893 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4894 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4895 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4896 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4897 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4898 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4899 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4900 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4901 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4902 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4903 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4904 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4905 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4906 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4907 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4908 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4909 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4910 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4911 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4912 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4913 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4914 \fi
4915 </founder>

```

14.15.4.4 ctex-zhmap-mac.tex

```

4916 <*mac>
4917 \ifzhmappdf
4918 %% pdfTeX does not support OTF fonts
4919 \else
4920 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
4921 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
4922 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc}
4923 \special{pdf:mapline gbkhei@UGBK@ unicode :2:PingFang.ttc}
4924 \special{pdf:mapline gbkfs@UGBK@ unicode STFANGSO.ttf}
4925 \special{pdf:mapline gbkli@UGBK@ UniGB-UTF16-H :0:Baoli.ttc}
4926 \special{pdf:mapline gbkyou@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc}

```

```

4927 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
4928 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
4929 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc}
4930 \special{pdf:mapline unihei@Unicode@ unicode :2:PingFang.ttc}
4931 \special{pdf:mapline unifs@Unicode@ unicode STFANGSO.ttf}
4932 \special{pdf:mapline unili@Unicode@ UniGB-UTF16-H :0:Baoli.ttc}
4933 \special{pdf:mapline uniyoud@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc}
4934 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H :3:Songti.ttc -s .167}
4935 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
4936 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:PingFang.ttc -s .167}
4937 \special{pdf:mapline gbkfssl@UGBK@ unicode STFANGSO.ttf -s .167}
4938 \special{pdf:mapline gbklisl@UGBK@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
4939 \special{pdf:mapline gbkyousl@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
4940 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H :3:Songti.ttc -s .167}
4941 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
4942 \special{pdf:mapline uniheisl@Unicode@ unicode :2:PingFang.ttc -s .167}
4943 \special{pdf:mapline unifssl@Unicode@ unicode STFANGSO.ttf -s .167}
4944 \special{pdf:mapline unilisl@Unicode@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
4945 \special{pdf:mapline uniyousl@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
4946 \fi
4947 </mac>

```

14.15.4.5 ctex-zhmap-ubuntu.tex

```

4948 <*ubuntu>
4949 \ifzhmappdf
4950 %% pdfTeX does not support OTF fonts
4951 \else
4952 \special{pdf:mapline gbk@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
4953 \special{pdf:mapline gbksong@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
4954 \special{pdf:mapline gbkkai@UGBK@ unicode gkai00mp.ttf}
4955 \special{pdf:mapline gbkhei@UGBK@ unicode :2:NotoSansCJK-Regular.ttc}
4956 \special{pdf:mapline gbkfs@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
4957 \special{pdf:mapline cyberb@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
4958 \special{pdf:mapline unisong@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
4959 \special{pdf:mapline unikai@Unicode@ unicode gkai00mp.ttf}
4960 \special{pdf:mapline unihei@Unicode@ unicode :2:NotoSansCJK-Regular.ttc}
4961 \special{pdf:mapline unifs@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
4962 \special{pdf:mapline gbksongsl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4963 \special{pdf:mapline gbkkaisl@UGBK@ unicode gkai00mp.ttf -s .167}
4964 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
4965 \special{pdf:mapline gbkfssl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4966 \special{pdf:mapline unisongsl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4967 \special{pdf:mapline unikaisl@Unicode@ unicode gkai00mp.ttf -s .167}
4968 \special{pdf:mapline uniheisl@Unicode@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
4969 \special{pdf:mapline unifssl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4970 \fi
4971 </ubuntu>

```

14.15.4.6 ctex-zhmap-windows.tex

```

4972 <*windows>
4973 \ifzhmappdf
4974 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4975 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4976 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4977 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4978 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4979 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4980 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4981 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4982 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4983 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4984 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4985 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4986 \pdfmapline{=unili@Unicode@ <simli.ttf}
4987 \pdfmapline{=uniyoud@Unicode@ <simyou.ttf}
4988 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}

```

```

4989 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4990 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4991 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4992 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4993 \pdfmapline{=gbkyouisl@UGBK@ <simyou.ttf}
4994 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4995 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4996 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4997 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4998 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4999 \pdfmapline{=uniyouisl@Unicode@ <simyou.ttf}
5000 \else
5001 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
5002 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
5003 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
5004 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
5005 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
5006 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
5007 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
5008 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
5009 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
5010 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
5011 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
5012 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
5013 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
5014 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
5015 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -v 50 -s .167}
5016 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -v 70 -s .167}
5017 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -v 150 -s .167}
5018 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -v 50 -s .167}
5019 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -v 150 -s .167}
5020 \special{pdf:mapline gbkyouisl@UGBK@ unicode simyou.ttf -v 60 -s .167}
5021 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -v 50 -s .167}
5022 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -v 70 -s .167}
5023 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -v 150 -s .167}
5024 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -v 50 -s .167}
5025 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -v 150 -s .167}
5026 \special{pdf:mapline uniyouisl@Unicode@ unicode simyou.ttf -v 60 -s .167}
5027 \fi
5028 </windows>
5029 \ctex@zhmap@endinput
5030 </zhmap>

```

14.15.5 制作 spa 文件

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

```

5031 <*spa>
5032 <*macro>
5033 \input expl3-generic %
5034 \ExplSyntaxOn
5035 \sys_if_engine_xetex:F
5036 {
5037     \msg_new:nnn { ctex } { xetex }
5038     { XeTeX~is~required~to~compile~this~document! }
5039     \msg_fatal:nn { ctex } { xetex }
5040 }

```

CJKpunct 定义的标点符号是:

“「『（（〔{〈《【【
—…>。：：！？%〕〕}〕〕】’”|】

注意顺序不能改变。

5041 \seq_const_from_clist:Nn \c_ctex_punct_seq

```

5042 {
5043   "2018", "201C", "300C", "300E", "3014", "FF08", "FF3B", "FF5B",
5044   "3008", "300A", "3016", "3010",
5045   "2014", "2026", "3001", "3002", "FF0C", "FF0E", "FF1A", "FF1B",
5046   "FF01", "FF1F", "FF05", "3015", "FF09", "FF3D", "FF5D", "3009",
5047   "300B", "3017", "3011", "2019", "201D", "300D", "300F"
5048 }

```

`\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。

```

5049 \cs_new_protected:Npn \ctex_make_spa:nn #1#2
5050 {
5051   \iow_open:Nn \g__ctex_spa_iow {#1}
5052   \clist_map_inline:nn {#2}
5053     { \__ctex_write_family:nn ##1 }
5054   \iow_close:N \g__ctex_spa_iow
5055 }
5056 \iow_new:N \g__ctex_spa_iow
5057 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

5058 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
5059 {
5060   \group_begin:
5061   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
5062   \l__ctex_punct_font
5063   \clist_clear:N \l__ctex_punct_bounds_clist
5064   \seq_map_inline:Nn \c__ctex_punct_seq
5065     { \exp_args:No \__ctex_save_bounds:n { \int_use:N \tex_XeTeXcharglyph:D ##1 } }
5066   \iow_now:Nx \g__ctex_spa_iow
5067   {
5068     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 `CJKpunct` 来说是必要的。

```

5069       { \l__ctex_punct_bounds_clist , , , }
5070     }
5071   \group_end:
5072 }
5073 \cs_new_protected:Npn \__ctex_save_bounds:n #1
5074 {
5075   \clist_put_right:Nx \l__ctex_punct_bounds_clist
5076     {
5077       \__ctex_calc_bounds:nn { 1 } {#1} ,
5078       \__ctex_calc_bounds:nn { 3 } {#1}
5079     }
5080 }
5081 \clist_new:N \l__ctex_punct_bounds_clist

```

`CJKpunct` 要求的格式是边界空白宽度与 1 em 的比值的一百倍。

```

5082 \cs_new:Npn \__ctex_calc_bounds:nn #1#2
5083 {
5084   \fp_eval:n
5085     {
5086       round
5087       (
5088         \dim_to_decimal_in_unit:nn
5089           { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
5090           { 1 em }
5091       )
5092     }
5093 }
5094 \ExplSyntaxOff
5095 </macro>

```

下面是 \TeX 定义的一些字体。

```

5096 <*make>

```

```

5097 \input ctex-spa-macro %
5098 \MAKESPA {ctexpunct.spa}
5099 {
5100   {adobezhsong}      {AdobeSongStd-Light} ,
5101   {adobezhhei}      {AdobeHeitiStd-Regular} ,
5102   {adobezhkai}      {AdobeKaitiStd-Regular} ,
5103   {adobezhfs}      {AdobeFangsongStd-Regular} ,
5104   %
5105   {fandolzhsong}    {FandolSong} ,
5106   {fandolzhsongb}   {FandolSong-Bold} ,
5107   {fandolzhhei}     {FandolHei} ,
5108   {fandolzhheib}    {FandolHei-Bold} ,
5109   {fandolzhkai}     {FandolKai} ,
5110   {fandolzhfs}      {FandolFang} ,
5111   %
5112   {founderzhsong}   {FZShuSong-Z01} ,
5113   {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
5114   {founderzhhei}    {FZHei-B01} ,
5115   {founderzhheil}   {FZXiHeiI-Z08} ,
5116   {founderzhkai}    {FZKai-Z03} ,
5117   {founderzhfs}     {FZFangSong-Z02} ,
5118   {founderzhli}     {FZLiShu-S01} ,
5119   {founderzhyou}    {FZXiYuan-M01} ,
5120   {founderzhyoub}   {FZZhunYuan-M02} ,
5121   %
5122   {maczhsong}       {Songti SC Light} ,
5123   {maczhsongb}      {Songti SC Bold} ,
5124   {maczhhei}        {Heiti SC Medium} ,
5125   {maczhheil}       {Heiti SC Light} ,
5126   {maczhkai}        {Kaiti SC} ,
5127   {maczhkaib}       {Kaiti SC Bold} ,
5128   {maczhfs}         {STFangsong} ,
5129   {maczhli}         {Baoli SC} ,
5130   {maczhyou}        {Yuanti SC Light} ,
5131   {maczhyoub}       {Yuanti SC Regular} ,
5132   {maczhpf}         {PingFang SC} ,
5133   {maczhpfb}        {PingFang SC Semibold} ,
5134   %
5135   {ubuntuzhsong}    {Noto Serif CJK SC} ,
5136   {ubuntuzhsongb}   {Noto Serif CJK SC Bold} ,
5137   {ubuntuzhhei}     {Noto Sans CJK SC} ,
5138   {ubuntuzhheib}    {Noto Sans CJK SC Bold} ,
5139   {ubuntuzhkai}     {AR PL KaitiM GB} ,
5140   %
5141   {windowszhsong}   {SimSun} ,
5142   {windowszhhei}    {SimHei} ,
5143   {windowszhkai}    {KaiTi} ,
5144   {windowszhfs}     {FangSong} ,
5145   {windowszhli}     {LiSu} ,
5146   {windowszhyou}    {YouYuan} ,
5147   {windowszhahei}   {Microsoft YaHei} ,
5148   {windowszhaheib}  {Microsoft YaHei Bold}
5149 }
5150 \primitive\end
5151 </make>
5152 </spa>

```

14.16 translator 宏包的中文字典

```
5153 <*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。

```

5154 <*theorem>
5155 \providetranslation{Comments}{评论}

```

```

5156 \providetranslation{comments}{评论}
5157 \providetranslation{Comment}{评论}
5158 \providetranslation{comment}{评论}
5159 \providetranslation{Corollaries}{推论}
5160 \providetranslation{corollaries}{推论}
5161 \providetranslation{Corollary}{推论}
5162 \providetranslation{corollary}{推论}
5163 \providetranslation{Definitions}{定义}
5164 \providetranslation{definitions}{定义}
5165 \providetranslation{Definition}{定义}
5166 \providetranslation{definition}{定义}
5167 \providetranslation{Examples}{例}
5168 \providetranslation{examples}{例}
5169 \providetranslation{Example}{例}
5170 \providetranslation{example}{例}
5171 \providetranslation{Exercises}{练习}
5172 \providetranslation{exercises}{练习}
5173 \providetranslation{Exercise}{练习}
5174 \providetranslation{exercise}{练习}
5175 \providetranslation{Facts}{事实}
5176 \providetranslation{facts}{事实}
5177 \providetranslation{Fact}{事实}
5178 \providetranslation{fact}{事实}
5179 \providetranslation{Key Lemmas}{关键引理}
5180 \providetranslation{key lemmas}{关键引理}
5181 \providetranslation{Key Lemma}{关键引理}
5182 \providetranslation{key lemma}{关键引理}
5183 \providetranslation{Key Observations}{关键观察}
5184 \providetranslation{key observations}{关键观察}
5185 \providetranslation{Key Observation}{关键观察}
5186 \providetranslation{key observation}{关键观察}
5187 \providetranslation{Lemmas}{引理}
5188 \providetranslation{lemmas}{引理}
5189 \providetranslation{Lemma}{引理}
5190 \providetranslation{lemma}{引理}
5191 \providetranslation{Main Theorems}{主要定理}
5192 \providetranslation{main theorems}{主要定理}
5193 \providetranslation{Main Theorem}{主要定理}
5194 \providetranslation{main theorem}{主要定理}
5195 \providetranslation{Observations}{观察}
5196 \providetranslation{observations}{观察}
5197 \providetranslation{Observation}{观察}
5198 \providetranslation{observation}{观察}
5199 \providetranslation{Problems}{问题}
5200 \providetranslation{problems}{问题}
5201 \providetranslation{Problem}{问题}
5202 \providetranslation{problem}{问题}
5203 \providetranslation{Proofs}{证明}
5204 \providetranslation{proofs}{证明}
5205 \providetranslation{Proof}{证明}
5206 \providetranslation{proof}{证明}
5207 \providetranslation{Proof Sketch}{证明提要}
5208 \providetranslation{Proof sketch}{证明提要}
5209 \providetranslation{proof sketch}{证明提要}
5210 \providetranslation{Proof Sketches}{证明提要}
5211 \providetranslation{Proof sketches}{证明提要}
5212 \providetranslation{proof sketches}{证明提要}
5213 \providetranslation{Sketch of Proof}{证明提要}
5214 \providetranslation{Sketch of Proofs}{证明提要}
5215 \providetranslation{Sketch of proof}{证明提要}
5216 \providetranslation{Sketch of proofs}{证明提要}
5217 \providetranslation{sketch of proof}{证明提要}
5218 \providetranslation{sketch of proofs}{证明提要}
5219 \providetranslation{Propositions}{命题}
5220 \providetranslation{propositions}{命题}
5221 \providetranslation{Proposition}{命题}
5222 \providetranslation{proposition}{命题}

```

```

5223 \providetranslation{Remarks}{注}
5224 \providetranslation{remarks}{注}
5225 \providetranslation{Remark}{注}
5226 \providetranslation{remark}{注}
5227 \providetranslation{Solutions}{解}
5228 \providetranslation{solutions}{解}
5229 \providetranslation{Solution}{解}
5230 \providetranslation{solution}{解}
5231 \providetranslation{Theorems}{定理}
5232 \providetranslation{theorems}{定理}
5233 \providetranslation{Theorem}{定理}
5234 \providetranslation{theorem}{定理}
5235 </theorem>
5236 </dict>

```

14.17 ctexcap 宏包

```
5237 <*ctexcap>
```

ctexcap 是过时宏包。

```

5238 \clist_new:N \l__ctex_ctexcap_options_clist
5239 \clist_set:Nx \l__ctex_ctexcap_options_clist
5240 { \exp_not:v { opt@ \@currname . \@currentext }, heading }
5241 \msg_new:nnn { ctexcap } { deprecated }
5242 {
5243   Package~`ctexcap'~is~deprecated.\\
5244   Please~use~package~`ctex'~with~option~`#1'~instead: \\
5245   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5246 }
5247 \msg_warning:nnx { ctexcap } { deprecated }
5248 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

5249 \PassOptionsToPackage { heading = true } { ctexcap }
5250 \RequirePackageWithOptions { ctex }
5251 </ctexcap>

```

14.18 ctexhook 宏包

```
5252 <*ctexhook>
```

如果宏包已经被载入,则钩子无效,给出警告。

```

5253 \cs_new_protected:Npn \ctex_at_begin_package:nn #1
5254 {
5255   \@ifpackageloaded {#1}
5256   { \__ctex_package_loaded_warning:nn {#1} }
5257   { \ctex_gadd_package_hook:nnn { before } {#1} }
5258 }
5259 \cs_new_protected:Npn \__ctex_package_loaded_warning:nn #1#2
5260 { \msg_warning:nnx { ctex } { invalid-hook } {#1} }
5261 \msg_new:nnn { ctex } { invalid-hook }
5262 {
5263   Package~`#1'~is~loaded. \\
5264   \token_to_str:N \ctex_at_begin_package:nn \{#1\}\{...\}~is~invalid.
5265 }

```

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。

```

5266 \cs_new_protected:Npn \ctex_at_end_package:nn #1
5267 {
5268   \@ifpackageloaded {#1}
5269   { \use:n }
5270   { \ctex_gadd_package_hook:nnn { after } {#1} }
5271 }

```

```
\ctex_at_end_package:nn
```


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。

```

5272 \@ifl@t@r \fmtversion { 2020/10/01 }
5273 {
5274   \cs_new_protected:Npx \ctex_at_end_preamble:n
5275     { \hook_gput_code:nnn { env/document/before } { \c_novalue_tl } }
5276   \cs_new_protected:Npx \ctex_after_end_preamble:n
5277     { \hook_gput_code:nnn { begindocument/end } { \c_novalue_tl } }
5278   \cs_new_protected:Npx \ctex_gadd_package_hook:nnn #1#2
5279     { \hook_gput_code:nnn { package/#1/#2 } { \c_novalue_tl } }
5280   \file_input_stop:
5281 }
5282 { }

```

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。

```

5283 \cs_new_protected:Npn \ctex_at_end_preamble:n
5284 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl }
5285 \cs_new_protected:Npn \ctex_after_end_preamble:n
5286 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl }
5287 \cs_new_protected:Npn \CTEX@document@left@hook
5288 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5289 \cs_new_protected:Npn \CTEX@document@right@hook
5290 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5291 \cs_set_nopar:Npx \document
5292 {
5293   \CTEX@document@left@hook
5294   \exp_not:o { \document }
5295   \CTEX@document@right@hook
5296 }
5297 \tl_new:N \g__ctex_end_preamble_hook_tl
5298 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

给钩子附加内容。

```

5299 \cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
5300 { \ctex_gadd_hook:cn { g__ctex_at_ #1 _ #2 _hook_tl } }
5301 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1
5302 {
5303   \tl_if_exist:NF #1 { \tl_new:N #1 }
5304   \tl_gput_right:Nn #1
5305 }
5306 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

宏包钩子,只执行一次,用后清除。

```

5307 \cs_new_protected:Npn \ctex_use_package_hook:nn #1#2
5308 {
5309   \group_begin: \exp_args:Nnc \group_end:
5310   \__ctex_use_package_hook_aux:N { g__ctex_at_ #1 _ #2 _hook_tl }
5311 }
5312 \cs_new_protected:Npn \__ctex_use_package_hook_aux:N #1
5313 { \cs_if_exist_use:NT #1 { \cs_undefine:N #1 } }

```

\@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。

```

5314 \tl_put_right:Nn \@resetoptions { \CTEX@reset@options@hook }
5315 \cs_new_protected:Npn \CTEX@reset@options@hook
5316 {
5317   \cs_if_eq:NNT \@currentx \@pkgextension
5318   { \ctex_use_package_hook:nn { before } { \@currname } }
5319 }

```

\@resetoptions
\CTEX@reset@options@hook


```

\@popfilename 对 \@popfilename 做补丁来实现 \ctex_at_end_package:nn 的功能。
\CTEX@popfilename@hook
5320 \tl_put_left:Nn \@popfilename { \CTEX@popfilename@hook }
5321 \cs_new_protected:Npn \CTEX@popfilename@hook
5322 {
5323   \cs_if_eq:NNT \@currentx \pkgextension
5324   { \ctex_use_package_hook:nn { after } { \@currname } }
5325 }
5326 </ctexhook>

```

14.19 ctexpatch 宏包

```
5327 <*ctexpatch>
```

\ctex_patch_cmd_once:NnnTF 只进行第一次匹配进行替换。参数 #2 是宏重建时的 \catcode 设置。

```

5328 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnTF #1#2
5329 {
5330   \ctex_patch_boot:NnnTF \__ctex_patch_cmd:Nnnnw #1
5331   { once } {#2} { \use_i:nn } { \use_ii:nn }
5332 }

```

\ctex_patch_cmd_all:NnnTF 替换所有匹配到的文本。

```

5333 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnTF #1#2
5334 {
5335   \ctex_patch_boot:NnnTF \__ctex_patch_cmd:Nnnnw #1
5336   { all } {#2} { \use_i:nn } { \use_ii:nn }
5337 }

```

\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。

```

5338 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5339 {
5340   \ctex_patch_boot:NnnTF \__ctex_patch_cmd:Nnnnw #1
5341   { once }
5342   {
5343     \ExplSyntaxOff
5344     \char_set_catcode_letter:n { 64 }
5345   }
5346   { }
5347   { \ctex_patch_failure:N #1 }
5348 }
5349 \cs_new_protected:Npn \ctex_patch_failure:N #1
5350 { \msg_warning:nxx { ctex } { patch-failure } { \token_to_str:N #1 } }
5351 \msg_new:nnn { ctex } { patch-failure }
5352 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

\ctex_preto_cmd:NnnTF 在宏的原本定义前面增加钩子。

```

5353 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5354 {
5355   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnnw #1
5356   { left } {#2} { \use_i:nn } { \use_ii:nn }
5357 }

```

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```

5358 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5359 {
5360   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnnw #1
5361   { right } {#2} { \use_i:nn } { \use_ii:nn }
5362 }

```

\ctex_patch_boot:NNnnTF

参数记号 # 作为宏的参数被读入时,总是会双写,会影响随后的字符串替换。需要先将它转换为普通符号。

```

5363 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
5364 {
5365   \cs_set_protected:Npx \__ctex_patch_true:w { \exp_not:n {#5} }
5366   \cs_set_protected:Npx \__ctex_patch_false:w { \exp_not:n {#6} }
5367   \group_begin:
5368     \char_set_catcode_other:n { 35 }
5369     \ctex_parse_name:NN #1 #2 {#3} {#4}
5370 }
5371 \cs_new_eq:NN \__ctex_patch_true:w \use_i:nn
5372 \cs_new_eq:NN \__ctex_patch_false:w \use_ii:nn

```

\ctex_parse_name:NN

用 \DeclareRobustCommand 定义的宏或者由 \newcommand 或 \newrobustcmd 定义的带一个可选参数的宏第一次展开的结果都不是其实际定义,实际定义被保存在另外的宏中。由这些命令定义的宏的第一次展开结果可以有下面的形式(细节可查阅 xpatch 的文档):

```

1 \protect_\xaa_\_          % \DeclareRobustCommand\xaa[1]{...}
2 \protect_\xab_\_          % \DeclareRobustCommand\xab[1] [] {...}
3 \@protected@testopt_\xac_\_ \xac_\_ {} % \newcommand\xac[1] [] {...}
4 \@testopt_\_ \xad_\_ {} % \newrobustcmd\xad[1] [] {...}
5 \x@protect_\_1\protect_\_1_\_ % \DeclareRobustCommand\1[1]{...}
6 \x@protect_\_2\protect_\_2_\_ % \DeclareRobustCommand\2[1] [] {...}
7 \@protected@testopt_\_3\_\_3_\_ {} % \newcommand\3[1] [] {...}
8 \@testopt_\_ \4_\_ {} % \newrobustcmd\4[1] [] {...}

```

ctexpatch 的主要原理是先对宏的 \meaning 作字符串替换,然后再用 \scantokens 来重建它。我们希望对宏的实际定义打补丁,为此需要先得到对应的名字。letltxmacro、show2e 和 xpatch 宏包中都有类似的工作。

```

5373 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5374 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5375 \group_begin:
5376 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5377 {
5378   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5379   {
5380     \bool_lazy_or:nnTF
5381     { \cs_if_exist_p:c { ##3 ~ } }
5382     { \cs_if_exist_p:c { #1##3 } }
5383     {
5384       \group_begin:
5385       \use:x
5386       {
5387         \group_end:
5388         \__ctex_parse_name:nNNNnN
5389         { \cs_replacement_spec:N ##2 }
5390         \exp_not:N ##2
5391         \exp_not:c { ##3 ~ }
5392         \exp_not:c { #1##3 }
5393       } { ##3 } ##1
5394     }
5395     { ##1##2 }
5396   }
5397   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5398   {
5399     \exp_args:Nc ##6
5400     {
5401       \str_case:nnTF {##1}
5402       {
5403         { \protect ##3 } { }
5404         { \x@protect ##2 \protect ##3 } { }
5405       }
5406       {
5407         \str_if_eq:eeTF

```

```

5408         { \exp_not:n { #1@protected@ ##3 #1##3 } }
5409         {
5410             \exp_last_unbraced:Ne \__ctex_parse_name:w
5411             { \cs_replacement_spec:N ##3 } #3 ~ #2 \q_stop
5412         }
5413         { #1##5 ~ } { ##5 ~ }
5414     }
5415     {
5416         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5417         {
5418             { #1@protected@ ##2 ##4 } { }
5419             { #1@ ##4 } { }
5420         }
5421         { #1##5 } { ##5 }
5422     }
5423 }
5424 }
5425 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5426 }
5427 \use:x
5428 {
5429     \__ctex_tmp:w
5430     { \c_backslash_str }
5431     { \c_left_brace_str }
5432     { \tl_to_str:n { testopt } }
5433 }
5434 \group_end:
5435 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

```

分别保存宏的 `\meaning` 中的前缀、参数文本和替换文本。

```

5436 \str_new:N \l__ctex_prefix_str
5437 \str_new:N \l__ctex_parameter_str
5438 \str_new:N \l__ctex_replacement_str

```

```

\ctex_get_macro_meaning:NTF
\__ctex_get_macro_meaning:w

```

解构待补丁宏的 `\meaning`。若命令不是宏,则走向 `false` 分支。

```

5439 \group_begin:
5440     \cs_set_protected:Npn \__ctex_tmp:w #1
5441     {
5442         \prg_new_protected_conditional:Npnn
5443         \ctex_get_macro_meaning:N ##1 { TF }
5444         {
5445             \exp_after:wN \__ctex_get_macro_meaning:w
5446             \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5447         }
5448         \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5449         ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5450         {
5451             \tl_if_empty:nTF { ##4 }
5452             { \prg_return_false: }
5453             {
5454                 \str_set:Nn \l__ctex_prefix_str { ##1 }
5455                 \str_set:Nn \l__ctex_parameter_str { ##2 }
5456                 \str_set:Nn \l__ctex_replacement_str { ##3 }
5457                 \prg_return_true:
5458             }
5459         }
5460     }
5461     \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5462 \group_end:

```

```

\ctex_if_rescanable:NnTF

```

检查宏是否可以重建。

```

5463 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5464 {
5465     \ctex_get_macro_meaning:NTF #1

```

```

5466     {
5467         \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5468         \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5469     }
5470     {#4}
5471 }
5472 \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。

```

5473 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5474 {
5475     \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
5476     \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
5477     \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5478     \use:x
5479     {
5480         \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5481         \exp_not:o { \l__ctex_parameter_tl }
5482         { \exp_not:o { \l__ctex_replacement_tl } }
5483     }
5484 }
5485 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5486 {
5487     \str_if_empty:NTF #2
5488     { \tl_clear:N #1 }
5489     { \tl_set_rescan:Nno #1 {#3} {#2} }
5490 }
5491 \tl_new:N \l__ctex_prefix_tl
5492 \tl_new:N \l__ctex_parameter_tl
5493 \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。

```

5494 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5495 {
5496     \group_end:
5497     \ctex_if_rescanable:NnTF #1 {#3}
5498     {
5499         \use:x
5500         {
5501             \__ctex_patch_replace:nnnTF {#2}
5502             { \tl_to_str:n {#4} }
5503             { \tl_to_str:n {#5} }
5504         }
5505         {
5506             \__ctex_patch_rebuild:Nn #1 {#3}
5507             \__ctex_patch_true:w
5508         }
5509         { \__ctex_patch_false:w }
5510     }
5511     { \__ctex_patch_false:w }
5512 }

```

__ctex_patch_replace:nnnTF 替换前先检查原文本是否存在。

```

5513 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5514 {
5515     \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5516     { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5517 }

```

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

5518 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5519 {
5520     \group_end:

```

```

5521 \ctex_get_macro_meaning:NTF #1
5522 {
5523   \str_if_empty:NTF \l__ctex_parameter_str
5524   { \__ctex_hookto_cmd_parameterless:Nnnnw }
5525   { \__ctex_hookto_cmd_parameter:Nnnnw }
5526   #1 {#2} {#3} {#4}
5527 }
5528 { \__ctex_patch_false:w }
5529 }

```

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。

```

5530 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5531 {
5532   \str_if_empty:NF \l__ctex_prefix_str
5533   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5534   \tex_edef:D #1
5535   {
5536     \use:c { __ctex_ #2 _hook_aux:nn }
5537     { \exp_not:o {#1} }
5538     { \exp_not:n {#4} }
5539   }
5540   \__ctex_patch_true:w
5541 }
5542 \cs_generate_variant:Nn \tl_rescan:nn { no }
5543 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5544 \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。

```

5545 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5546 {
5547   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5548   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5549   {
5550     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5551     \__ctex_patch_rebuild:Nn #1 {#3}
5552     \__ctex_patch_true:w
5553   }
5554   { \__ctex_patch_false:w }
5555 }
5556 </ctexpatch>

```

版本历史

v2.0 (2014/03/06 – 2015/05/06)

General: c5size, cs4size 是过时选项。	41
captiondelimiter 是过时选项。	118
fancyhdr 成为过时选项, 原选项功能总是打开。	44
fnctef 成为过时选项, 原选项功能总是打开。	44
hyperref 成为过时选项, 原选项功能总是打开。	44
indent, noindent 是过时选项。	42
nofonts, adobefonts, winfonts 是过时选项。	42
nopunct 是过时选项。	43
nospace 是过时选项。	43
nozhmap 是过时选项。	43
punct 选项可以设置标点格式。	43
ctex 宏包新增 heading 选项。	43
\CTEXindent, \CTEXnoindent 是过时命令。	86
\CTEXsetup, \CTEXoptions 是过时命令。	15, 82
\CTEXunderdot, \CTEXunderline、 \CTEXunderdblline、\CTEXunderwave、\CTEXsout、 \CTEXxout 是过时命令; CTExfilltwosides 是过时环 境。	78
标题设置新增 pagestyle 选项。	89
调整 \footnotesep 的大小, 以适合行距的变化。	118
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号 选项。	44
将标题汉化功能加入 ctex.sty。	87
将中文字号功能提取到可以独立使用的 ctexsize。	115
解决 etoolbox 与 breqn 关于 \end 的冲突。	119
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	78
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jan 宏包支持 Lua _{La} TeX。	52
新增 autoindent 选项。	42
新增 fontset 选项。	42
新增 linespread 选项。	42
新增 linestretch 选项。	83
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。	44
新增 zhmCJK 支持选项。	43
新增 zihao 选项。	41
新增统一设置接口 \ctexset。	82
应用 _{La} TeX3 重新整理代码。	1
中文字号不再采用近似值。	115
自动检测操作系统, 载入对应的字体配置。	77
\CJK@surr: 解决与 \nouppercase 的冲突。	49

v2.0.1 (2015/05/15)

General: 修复 10pt、11pt 等选项无效的问题。	44
---	----

v2.0.2 (2015/05/16)

General: 修复加载 ctex 宏包后章节标题后第一段无段首 缩进的问题。	115
---	-----

v2.1 (2015/05/18 – 2015/06/19)

General: format+, nameformat+ 等带加号的选项, 加号与 前面的文字之间可以有可选的空格。	88
nameformat 可以接受章节名字为参数。	88
不依赖 ifpdf 宏包。	40
不再设置 hyperref 宏包的 colorlinks 选项。	77
给 pdf _{La} TeX 下的非 UTF-8 编码 CJK 字体族加上 CMap。	47

将章节标题设置功能提取到可以独立使用的宏包

ctexheading 中。	1
新的标题格式选项 aftertitle。	88
修复 ctexbook 和 ctexrep 类的中文 part/number 选项 初值为空的错误。	109
\ctex_if_platform_macos:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	120
hyperref: 补充定义 \hypersetup。	44

v2.2 (2015/06/21 – 2015/06/30)

General: before skip 和 after skip 选项的符号不再有特 殊意义。	88
before skip、after skip 和 indent 选项支持表达式。	91
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/before skip 和 part/after skip 选项有意义。	91
给 enumitem 宏包注册 \chinese 和 \zhnum。	86
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。	139
将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。	109
删去 etoolbox 与 breqn 的兼容补丁。	119
新的标题格式选项 afterindent。	88
新的标题格式选项 numbering。	88
新的标题格式选项 runin。	88
新增子宏包 ctexpatch 实现给宏打补丁功能。	141

v2.3 (2015/09/17 – 2016/01/05)

General: .value_required: 和 .value_forbidden: 已 过时。	41
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。	40
更新 LuaTeX-jan 支持 (20150922.0)。	52
更新 unicode-math 宏包补丁。	54
兼容 titleps 宏包。	103
修复 nameformat 作用域问题。	88
与 _{La} TeX3 (2015/12/20) 同步。	38

v2.4 (2015/02/19 – 2016/04/25)

General: 初步支持 up _{La} TeX。	73
加强 beamer 宏包支持。	1
提供 up _{La} TeX 的 NFSS 字体定义。	121
提供 translator 宏包的中文定理名称翻译。	137
正确更新 CJK 包的 \CJKfamilydefault。	76
正确设置 up _{La} TeX 下字体命令。	74
\ctex_parse_name:NN: 修复宏名解析错误。	142

v2.4.1 (2016/04/26 – 2016/05/14)

General: beamer 不调整默认行距。	114
beamer 不调整默认字体大小。	114
使用 bootfont.bin 判断 Windows XP 以避免权限问 题。	129
随字体更新 up _{La} TeX 的 \xkanjiskip。	81
新的标题格式选项 part/fixbefore skip 和 chapter/fixbefore skip。	89
正确更新 up _{La} TeX 的 \CJKfamilydefault。	76
\ccwd: 正确设置 up _{La} TeX 下的 \ccwd。	78
\chinese: 支持 \pagenumbering。	86
zihao: 不允许无参 zihao 选项。	41

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。	54	<code>\CTEX@selectfont@hook</code> : 修正导言区 <code>\selectfont</code> 钩子位置。	79
<code>\em</code> : 兼容 <code>upL^AT_EX</code> 2016/05/07u00 的定义。	74	v2.4.14	(2018/05/01)
v2.4.3	(2016/06/03 – 2016/08/26)	General: 配置 <code>macnew</code> 的默认字体设置。	126
General: 更新 <code>unicode-math</code> 补丁。	61	区分 <code>macold</code> 及 <code>macnew</code> 。	126
简化 <code>fontspec</code> 补丁。	54	为 <code>macnew</code> 配置字体命令。	130
确保 <code>\proofname</code> 非空。	87	v2.4.15	(2019/01/29 – 2019/04/05)
删除选项 <code>part/fixbeforeSKIP</code> 和 <code>chapter/fixbeforeSKIP</code> 。	89	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。	91
新的标题格式选项 <code>fixSKIP</code> 。	88	将 <code>upL^AT_EX</code> 的默认字体由 <code>mc</code> 改为 <code>zhrm</code> , 并启用 <code>\jfam</code> 。	74
v2.4.4	(2016/09/09 – 2016/09/19)	将 <code>JY2</code> 和 <code>JT2</code> 编码的字体定义提取到单独的文件中。	121
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	51	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。	106
改进 <code>hyperref</code> 宏包的标题锚点设置。	100	同步 <code>L^AT_EX 3</code> 2019/03/05。	1
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	48	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。	125
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	102	显式补丁 <code>upL^AT_EX</code> 的 <code>\rmfamily</code> 等字体命令。	73
提供 <code>\CTEXifname</code> 。	91	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。	91
提供 <code>\partmark</code> 。	91	v2.4.16	(2019/05/11 – 2019/05/29)
新的标题格式选项 <code>break</code> 。	88	General: 更好地兼容 <code>nameref</code> 宏包。	101
v2.4.5	(2016/10/01 – 2016/10/25)	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法, 在其标题内部禁用 <code>autoindent</code> 。	91
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	89	允许设置 <code>autoindent</code> 为 0。	40
新的标题格式选项 <code>hang</code> 。	88	v2.5	(2019/10/25 – 2020/04/30)
新的标题格式选项 <code>tocline</code> 。	88	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。	128
<code>\ps@headings</code> : 修复补丁失败。	105	标题选项 <code>format</code> 也可以接受参数。	91
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。	129
General: <code>\CTEXifname</code> 初始为假。	91	不再自动载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。	78
支持字体属性可选项在后的新语法。	66	操作系统检测移动至载入中文字库处, 且不再需要依赖特定引擎。	77
重新初始化 <code>\ifthechapter</code> 等。	104	处理 <code>\ctex_file_input:n</code> 在 <code>ctexsize</code> 中未定义的错误。	38
v2.4.7	(2016/12/23 – 2016/12/27)	给 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下的文档类指定驱动为 <code>DVIPDFMx</code> 。	46
General: 依赖 <code>pxeverysel</code> 宏包。	75	更新 <code>LuaTeX-j</code> 支持(20200412.0)。	52
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	105	兼容 <code>titletoc</code> 宏包。	104
v2.4.8	(2017/02/23)	删除 <code>fontspec</code> 补丁。	54
General: 解决与 <code>fontspec</code> 2017/01/24 v2.5d 的字体族匹配兼容问题。	67	使用环境变量代替绝对路径查找字体。	129
v2.4.9	(2017/02/27)	所有引擎下默认编码均设为 UTF-8。	42
General: 调整 <code>unicode-math</code> 补丁的代码顺序。	61	为 <code>macnew</code> 增加粗楷体、隶书和圆体的定义。	126
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。	78
General: 常数 <code>\c_minus_one</code> 已过时。	38	应用新内核中的 <code>\labelformat</code> 。	108
定义 <code>\cht</code> 、 <code>\cdp</code> 和 <code>\cwd</code> 。	81	允许 <code>macnew</code> 在 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下使用。	126
使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 (<code>L^AT_EX 3</code> 2017/07/19)。	38	在 <code>ctexsize</code> 也载入 <code>fix-cm</code> 。	38
v2.4.11	(2017/08/17 – 2017/11/21)	增加宏包开头钩子。	139
General: 不把 Enclosed Alphanumerics 设置为 JAchar。	52	增加字体映射文件 <code>zhmacfonts.tex</code> 。	133
不把希腊和西里尔字母设置为 JAchar。	52	重构标题选项 <code>indent</code> 和 <code>hang</code> 。	90
因上游 <code>l3keys</code> 变化, 重新定义 <code>format_U+</code> 等带空格加号的选项。	88	重构字体选项 <code>AlternateFont</code> 。	68
<code>\ps@headings</code> : 补充页眉空格。	105	<code>\CTEX@selectfont@hook</code> : 兼容 KOMA-Script 的 <code>\selectfont</code> 补丁。	79
v2.4.12	(2017/12/05 – 2018/01/27)	<code>\ctex_if_platform_macos:TF</code> : 改用 <code>/System/Library/Fonts/Menlo.ttf</code> 为特征文件。	120
General: 同步 <code>L^AT_EX 3</code> 2017/12/16。	1	<code>fntef</code> : 仅在该选项启用时会载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。	44
修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。	82		
正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	73		
<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	118		

v2.5.1	(2020/05/02)	v2.5.3	(2020/05/31 – 2020/06/06)
General: zhconv 更名为 ctex-zhconv。	1	General: 不再依赖 xunicode, 单独补丁 tuenc.def。	53
v2.5.2	(2020/05/05 – 2020/05/06)	兼容 cleveref。	108
General: ctexmakespa.tex 更名为 ctex-spa-make.tex。 . .	135	正确关闭和恢复 L ^A T _E X3 语法环境。	39
ctexspamacro.tex 更名为 ctex-spa-macro.tex。 . .	135	v2.5.4	(2020/06/07 – 2020/08/16)
zhadobefonts.tex 等字体映射文件更名为		General: 更新 LuaTeX-ja 支持(20200808.0)。	52
ctex-zhmap-*.tex。	131	兼容 L ^A T _E X 2020/10/01 的钩子机制。	139
兼容 L ^A T _E X 2020-02-02 之前的版本。	75	同时兼容 cleveref 和 hyperref。	108
新增标题选项 secnumdepth 和 tocdepth。	107	修正主要字体命令补丁。	75
修正 macnew 和 ubuntu 字库的 CJKpunct 标点信息。 .	126	应用 l3cctab。	39

代码索引

意大利体的数字表示描述对应索引项的页码; 带下划线的数字表示定义对应索引项的代码行号; 罗马字体的数字表示使用对应索引项的代码行号。

Symbols			
_	3327, 3334, 3350, 3353	\abovedisplayskip	3967, 3970, 3975, 3982, 3986, 3993, 4006, 4009, 4014, 4021, 4025, 4032
\#	481, 3223, 3282	\abstractname	2322
.../afterindent	24	abstractname	12
.../aftername	20	\addCJKfontfeature	1280, 1324
.../aftername+	20	\addCJKfontfeatures	1297
.../afterskip	23	\addcontentsline	2550, 2595, 2599, 2604, 2636, 2640, 2645, 2745, 2750, 2756, 2762, 2891, 2905
.../aftertitle	21	\AddEnumerateCounter	2277, 2279, 2280, 2281
.../aftertitle+	21	\addpenalty	2848, 3604, 3637, 3671, 3685, 3699
.../beforeskip	23	\addtocontents	2538, 2544, 2767, 2768
.../break	24	\addvspace	2539, 2545, 2556, 2559, 2767, 2768, 2848, 2852, 3366, 3377, 3385, 3396, 3404, 3415
.../break+	24	adobefonts	33
.../fixskip	24	\algorithmname	2329, 2345
.../format	18	algorithmname	13
.../format+	18	AlternateFont	30, 35, 1409
.../hang	22	\appendix	3060, 3481
.../indent	23	appendix/name	25, 3049
.../name	17	appendix/number	26, 3049
.../nameformat	18	appendix/numbering	25, 3049
.../nameformat+	18	\appendixname	2324, 3748
.../number	17	appendixname	12
.../numberformat	20	\arabic	3651
.../numberformat+	20	\AtBeginDocument	544, 1683
.../titleformat	20	\AtBeginDvi	436
.../titleformat+	20	\AtBeginShipoutFirst	440, 1866, 1881
.../tocline	25	\AtBeginUTFCommand	764
\	1693	\AtEndOfClass	133
\\	15, 33, 34, 109, 155, 157, 204, 213, 640, 1198, 1280, 2338, 3763, 3764, 3877, 3878, 4126, 4128, 4157, 5243, 5244, 5245, 5263, 5352	\AtEndOfPackage	134
\{	1209, 1210, 5245, 5264	\AtEndUTFCommand	765
\}	1209, 1210, 5245, 5264	autoindent	11, 183, 2127
10pt	8, 335	B	
11pt	8, 335	\baselinestretch	2186
12pt	8, 335	\begin	3373, 3392, 3411
\abovedisplayshortskip	3968, 3976, 3987, 4007, 4015, 4026	\begingroup	409, 2609, 2651, 2671, 2692, 2791, 2816, 2881, 2919, 2952, 3361, 3380, 3399, 4749, 4783, 4790

- \belowdisplayshortskip . 3969, 3977, 3988, 4008, 4016, 4027
- \belowdisplayskip 3970, 3982, 3993, 4009, 4021, 4032
- \bfdefault 1065, 1071, 1399, 1406
- \bfseries 2614, 2618, 2656, 2660, 2676,
2697, 2797, 2802, 2821, 3532, 3534, 3538, 3551, 3552,
3555, 3581, 3583, 3589, 3606, 3610, 3631, 3665, 3682, 3696
- \bibname 2327, 2330, 2338, 2346, 2352, 2353
- bibname 12
- bool commands:**
 - \bool_gset_false:N 246, 251
 - \bool_gset_true:N 241
 - \bool_if:NTF 1238, 2974, 3164, 3174, 3185, 3767
 - \bool_lazy_or:nnTF 4103, 4734, 5380
 - \bool_new:N 129, 1275
 - \bool_set_false:N 1250
 - \bool_set_true:N 1268
 - \c_false_bool 3008, 3010
 - \c_true_bool 3007, 3009
- box commands:**
 - \box_ht:N 4048
 - \box_new:N 62
 - \box_use_drop:N 2984
 - \box_wd:N 1914, 2983, 3278
- C**
- c5size 33
- cap 33, 294
- captiondelimiter 34, 4055
- \catcode 4749, 4750, 4751, 4752, 4753, 4757, 4758
- CCT 33
- cctab commands:**
 - \cctab_begin:N 77, 89, 513
 - \cctab_const:Nn 81, 517
 - \cctab_end: 79, 96, 515
 - \cctab_select:N 83, 519
 - \c_document_cctab 83
- CCTfont 33
- \ccwd 23, 27, 127, 187, 1910, 1955,
1956, 1957, 2133, 2171, 2195, 2196, 2203, 2229, 2230,
2231, 2234, 2236, 2237, 2248, 2259, 2261, 4065, 4066, 4067
- \cdp 2041
- \centering 2652, 2693, 3362,
3381, 3400, 3538, 3550, 3555, 3559, 3589, 3610, 3618, 3653
- \chapter 2722, 3773
- chapter/beforeskip 35
- chapter/fixbeforeskip 35
- chapter/lofskip 25, 2465
- chapter/lotskip 25, 2465
- chapter/numbering 16
- chapter/pagestyle 21, 2465
- \chaptermark 2766
- \chaptername 3578
- char commands:**
 - \char_set_catcode_comment:n 652
 - \char_set_catcode_letter:n . 84, 1686, 3298, 3488, 5344
 - \char_set_catcode_other:N 481, 977, 3223, 3282
 - \char_set_catcode_other:n 5368
 - \char_set_catcode_space:n 770, 2046
- CharRange 30, 1409
- \Chinese 34, 2273
- \chinese 29, 2271, 2281, 3525, 3588
- \cht 2041
- \CJKbold 4203, 4204, 4214, 4215, 4233, 4234, 4236, 4237, 4239,
4240, 4244, 4245, 4247, 4248, 4250, 4251, 4267, 4268,
4270, 4271, 4273, 4274, 4278, 4279, 4281, 4282, 4284, 4285
- \CJKfamily 1137, 1157, 1504, 1701, 1797, 1798, 1799,
1800, 4724, 4725, 4726, 4727, 4729, 4730, 4732, 4742, 4743
- \CJKfamilydefault 559, 1363, 1365, 1369, 1376,
1389, 1791, 1792, 1800, 1809, 1816, 1820, 1828, 1839, 1841
- \CJKfontspec 1297
- \CJKglue 1913, 1927, 1941, 1953
- \CJKhook 572
- \CJKnormal 4196, 4197, 4198, 4199, 4200, 4201, 4202,
4207, 4208, 4209, 4210, 4211, 4212, 4213, 4232, 4235,
4238, 4243, 4246, 4249, 4266, 4269, 4272, 4277, 4280, 4283
- \CJKpunctmapfamily 606, 607, 608, 609, 610, 611, 615, 616, 617, 618, 619, 620
- \CJKrmdefault 504, 1205, 1216,
1328, 1365, 1368, 1369, 1792, 1794, 1797, 1803, 1806,
1824, 4320, 4323, 4324, 4329, 4330, 4331, 4384, 4387,
4389, 4394, 4395, 4396, 4450, 4453, 4454, 4461, 4462,
4463, 4530, 4533, 4534, 4541, 4543, 4544, 4611, 4614,
4617, 4623, 4624, 4625, 4676, 4677, 4678, 4686, 4687, 4688
- \CJKsfdefault 505, 1206,
1217, 1335, 1795, 1798, 1804, 1807, 1825, 4321, 4385,
4388, 4451, 4455, 4531, 4535, 4542, 4612, 4615, 4679, 4680
- \CJKtilde 560
- \CJKttdefault 506, 1207, 1218, 1341, 1796, 1799,
1805, 1808, 1826, 4322, 4386, 4452, 4532, 4613, 4617, 4681
- clearalternatefont 31, 1588
- \cleardoublepage 2570, 2723, 3548, 3576
- \clearpage 2572, 2723, 3548, 3576
- clist commands:**
 - \clist_clear:N 1086, 5063
 - \clist_concat:NNN 1106
 - \clist_const:Nn 585
 - \clist_gput_right:Nn 351, 363, 384, 385
 - \clist_gset:Nn 1351
 - \clist_if_empty:NTF 1434
 - \clist_map_break:n 3143, 3758
 - \clist_map_function:nN . . 1417, 1591, 1593, 1626, 1640
 - \clist_map_inline:Nn 575, 1534
 - \clist_map_inline:nn 336, 598, 1633, 1905, 3138, 3755, 3895, 3930, 5052
 - \clist_new:N 136, 1103, 1352, 1430, 1477, 5081, 5238
 - \clist_put_left:Nn 1107, 3824, 3850
 - \clist_put_right:Nn 1267, 1523, 5075
 - \clist_set:Nn 1090, 1420, 1460, 1461, 1680, 5239
 - \clist_use:Nn 5248
- \clubpenalty 2951, 2958
- \contentsname 2317

- contentsname 12
- continuation 13
- cs commands:
 - \cs_generate_variant:Nn
 - ... 98, 457, 469, 1052, 1169, 1256, 1276, 1277, 1478, 1494, 1566, 1587, 1626, 1634, 2986, 3437, 5306, 5435, 5542
 - \cs_gset_eq:NN 242, 247, 252, 441, 489, 883, 1243, 1246, 3096, 3097, 3102, 3127
 - \cs_gset_nopar:Npx 878
 - \cs_gset_protected:Npn 3061, 3108, 3123
 - \cs_gset_protected:Npx 1134, 1498, 1584, 3037
 - \cs_gset_protected_nopar:Npx 920
 - \cs_if_eq:NNTF . 1011, 3094, 3126, 5317, 5323, 5468, 5548
 - \cs_if_exist:NTF . 44, 327, 466, 872, 888, 918, 952, 1030, 1497, 1582, 1783, 2005, 3209, 3457, 3495, 3773, 3775
 - \cs_if_exist_p:N 5381, 5382
 - \cs_if_exist_use:N 1153, 1154
 - \cs_if_exist_use:NTF 4, 720, 855, 1175, 1602, 5313
 - \cs_if_free:NTF 451, 461, 577, 593, 1240, 1398, 1972, 1975, 2146, 2160, 2277, 3157, 3202, 3248, 3253, 3257
 - \cs_new:Npn 119, 126, 937, 967, 972, 974, 978, 1082, 1148, 1156, 1201, 1212, 1655, 2271, 2447, 3011, 3019, 3027, 3046, 3182, 3246, 3251, 5082, 5425, 5543
 - \cs_new:Npx 4074, 4080
 - \cs_new_eq:NN
 - ... 6, 7, 8, 9, 20, 23, 113, 114, 115, 131, 132, 328, 458, 563, 624, 897, 922, 988, 1018, 1324, 1331, 1835, 1953, 1963, 1969, 2272, 2273, 2274, 2505, 2524, 2527, 2995, 3005, 3006, 3007, 3008, 3009, 3010, 3060, 3084, 3085, 3498, 3771, 3776, 4738, 4739, 5057, 5371, 5372, 5472, 5544
 - \cs_new_protected:Npn
 - ... 65, 68, 70, 72, 75, 87, 117, 133, 134, 142, 144, 149, 431, 433, 435, 449, 459, 464, 470, 502, 511, 524, 564, 570, 573, 591, 596, 604, 613, 622, 692, 694, 700, 702, 710, 712, 717, 730, 735, 750, 852, 862, 875, 880, 886, 939, 950, 962, 981, 990, 1006, 1020, 1045, 1083, 1104, 1113, 1132, 1149, 1159, 1184, 1222, 1234, 1236, 1258, 1260, 1283, 1359, 1384, 1393, 1416, 1418, 1425, 1432, 1438, 1445, 1447, 1452, 1458, 1479, 1488, 1495, 1501, 1512, 1520, 1526, 1532, 1550, 1559, 1567, 1579, 1597, 1614, 1632, 1635, 1637, 1650, 1669, 1677, 1691, 1694, 1696, 1720, 1734, 1747, 1760, 1775, 1780, 1814, 1846, 1850, 1888, 1896, 1900, 1910, 1924, 1954, 1958, 1970, 2020, 2145, 2147, 2172, 2180, 2192, 2211, 2221, 2226, 2380, 2410, 2458, 2488, 2495, 2500, 2508, 2515, 2522, 2528, 2530, 2533, 2549, 2963, 2969, 2971, 2979, 2987, 2996, 3077, 3087, 3119, 3130, 3150, 3155, 3160, 3170, 3180, 3224, 3272, 3290, 3431, 3453, 3462, 3468, 3473, 3483, 3868, 3883, 3915, 3920, 3950, 3956, 4070, 4086, 4096, 4098, 5049, 5058, 5073, 5253, 5259, 5266, 5283, 5285, 5287, 5289, 5299, 5301, 5307, 5312, 5315, 5321, 5328, 5333, 5338, 5349, 5353, 5358, 5363, 5373, 5378, 5397, 5448, 5463, 5473, 5485, 5494, 5513, 5518, 5530, 5545
 - \cs_new_protected:Npx
 - ... 474, 723, 1307, 3211, 5274, 5276, 5278
 - \cs_new_protected_nopar:Npn 898, 912
 - \cs_replacement_spec:N 5389, 5411
 - \cs_set:Npn 1966, 3263
 - \cs_set_eq:NN 116, 188, 193, 198, 522, 542, 567, 571, 579, 594, 704, 737, 745, 752, 754, 755, 891, 892, 989, 1019, 1037, 1080, 1137, 1497, 1504, 1514, 1515, 1516, 1582, 1583, 1693, 1819, 1984, 1987, 1988, 1991, 2288, 2293, 2296, 2301, 2529, 2531, 2998, 2999, 3000, 3001, 3002, 3003, 3091, 3134, 3144
 - \cs_set_nopar:Npn 1719
 - \cs_set_nopar:Npx 3172, 5291
 - \cs_set_protected:Npn
 - ... 566, 1023, 1038, 1927, 3204, 3261, 3957, 5376, 5440
 - \cs_set_protected:Npx 715, 1960, 5365, 5366
 - \cs_to_str:N 60, 1306, 1778, 5374
 - \cs_undefine:N . 1117, 1118, 1122, 1123, 1517, 1992, 5313
 - cs4size 33
 - \csname 410, 413, 414, 2849, 2858, 2859, 2860, 2867, 2887, 2888, 2890, 2893, 2902, 2903, 2904, 2907, 3490, 3491, 4782, 4784, 4785, 4786, 4791, 4795
 - \CTeX 29, 4051
 - ctex commands:
 - \ctex_add_cmap:n 454, 459
 - \ctex_add_cmap:Nn 462, 464, 469
 - \ctex_add_dvi_zhmap: 435, 437, 441
 - \ctex_add_to_selectfont:n 1958, 2011, 2039, 2066, 2159
 - \ctex_after_end_preamble:n ... 40, 116, 437, 5276, 5283
 - \ctex_appto_cmd:NnnTF 485, 742, 1685, 5358
 - \ctex_assign_heading_name:nn 2416, 2455, 3053
 - \ctex_at_begin_package:nn 140, 5253
 - \ctex_at_end:n 133, 265, 303, 309, 320, 325, 652, 1953, 4164
 - \ctex_at_end_package:nn 141, 438, 488, 761, 767, 1028, 1860, 1862, 1880, 1885, 1892, 2275, 3106, 3121, 3190, 3197, 3256, 3258, 3259, 3295, 3338, 3466, 3504, 4053, 5266
 - \ctex_at_end_preamble:n 330, 549, 556, 1813, 5274, 5283
 - \ctex_auto_ignorespaces: 563, 571, 2076
 - \ctex_CJK_input:n 511
 - \ctex_cleveref_hook: 3466, 3473
 - \ctex_declare_math_sizes:nnnn 3915, 3949
 - \ctex_define:n
 - ... 64, 1588, 1627, 2068, 2089, 2127, 2161, 2289, 2315, 2343, 2353, 2356, 2359, 2487, 3049, 3422, 4055, 4131
 - \ctex_define_option:n 64, 164, 357, 360
 - \ctex_deprecated_command:Nn
 - ... 142, 2110, 2112, 2120, 2122, 2258, 2265
 - \ctex_deprecated_option:n
 - ... 142, 146, 202, 211, 302, 308, 312, 319, 324, 4059
 - \ctex_deprecated_option:nn 142, 269
 - \ctex_detect_platform: 4086, 4101, 4116
 - \ctex_family_cmap:nn 449, 458
 - \ctex_file_input:n 52, 75, 366, 507, 625, 753, 2102, 3782, 3817, 3822, 3827, 3828, 3843, 3848, 3854, 3855, 3960, 3961, 4123, 4164, 4501, 4502
 - \ctex_fix_varioref_label:n 3465, 3468
 - \ctex_fontset_case:nnn 4074, 4435, 4658
 - \ctex_fontset_case:nnnn . 4080, 4296, 4360, 4505, 4594

- \ctex_fontset_error:n [4070](#), [4297](#),
[4333](#), [4361](#), [4398](#), [4465](#), [4506](#), [4508](#), [4509](#), [4546](#), [4595](#), [4627](#)
- \ctex_gadd_hook:Nn [572](#), [5299](#)
- \ctex_gadd_package_hook:nnn ... [5257](#), [5270](#), [5278](#), [5299](#)
- \ctex_get_macro_meaning:N [5443](#)
- \ctex_get_macro_meaning:NTF [5439](#), [5465](#), [5521](#)
- \ctex_hang_from:n [2971](#)
- \ctex_heading_depth:nn [3425](#), [3428](#), [3431](#)
- \ctex_heading_glue:n [2987](#)
- \ctex_heading_hang:Nnn [2965](#), [2970](#), [2971](#)
- \ctex_hypersetup:n
..... [77](#), [328](#), [1843](#), [1855](#), [1858](#), [1873](#), [1876](#), [1879](#)
- \ctex_if_autoindent_touched:TF [131](#), [188](#), [193](#), [198](#), [3809](#)
- \ctex_if_ccglue_touched:TF [1938](#), [2182](#)
- \ctex_if_ccglue_touched_p: [1938](#)
- \ctex_if_platform_macos:TF [4091](#), [4096](#)
- \ctex_if_preamble:TF [115](#), [4135](#)
- \ctex_if_rescanable:NnTF [5463](#), [5497](#)
- \ctex_ignorespaces_case:N [564](#), [2073](#), [2076](#), [2079](#)
- \ctex_indent_box:n [2495](#)
- \ctex_load_fontset: [4098](#), [4143](#), [4163](#)
- \ctex_load_zhmap:nnnn
..... [502](#), [4327](#), [4392](#), [4459](#), [4539](#), [4621](#), [4684](#)
- \ctex_ljt_add_font_features:n [1258](#), [1321](#)
- \ctex_ljt_add_font_features:nn [1258](#), [1259](#)
- \ctex_ljt_add_kyenc:n [779](#), [1061](#)
- \ctex_ljt_char_range_key:nn [1674](#), [1677](#)
- \ctex_ljt_clear_alt_font:n [816](#), [1545](#)
- \ctex_ljt_clear_alter_nate_font:n [1591](#), [1597](#)
- \ctex_ljt_declare_alter_nate_shape:nnnnnn [1506](#), [1526](#)
- \ctex_ljt_declare_char_range:n [1629](#), [1632](#), [1634](#)
- \ctex_ljt_declare_char_range:nn [1636](#), [1637](#)
- \ctex_ljt_def_char_range_key:n [1642](#), [1669](#)
- \ctex_ljt_ensure_default_family: [1359](#), [1832](#)
- \ctex_ljt_extract_font: [891](#), [898](#)
- \ctex_ljt_family_if_exist:nNTF
..... [1161](#), [1170](#), [1363](#), [1368](#), [1386](#), [1389](#), [1600](#)
- \l_ctex_ljt_family_tl
[1163](#), [1168](#), [1225](#), [1226](#), [1241](#), [1245](#), [1248](#), [1259](#), [1594](#), [1595](#)
- \ctex_ljt_fontspec:nn [1222](#), [1234](#), [1269](#), [1316](#)
- \ctex_ljt_get_and_define_fonts:nN [981](#)
- \ctex_ljt_get_and_define_fonts_al:nN [985](#), [988](#)
- \ctex_ljt_get_and_define_fonts_ja:nN [984](#), [990](#)
- \ctex_ljt_if_alter_nate_shape_exist:nTF
..... [901](#), [914](#), [932](#), [941](#)
- \ctex_ljt_if_jfont:nTF [61](#), [967](#), [979](#), [983](#)
- \ctex_ljt_if_jfont_math:NTF [974](#), [1013](#)
- \ctex_ljt_is_kenc:n [782](#), [969](#)
- \ctex_ljt_math_group_hook: ... [61](#), [61](#), [1012](#), [1018](#), [1038](#)
- \ctex_ljt_patch_external_font:n [785](#), [938](#)
- \ctex_ljt_pickup_alt_font:nn [822](#), [943](#)
- \ctex_ljt_pickup_font: ... [868](#), [886](#), [957](#), [993](#), [995](#), [997](#)
- \ctex_ljt_reset_alter_nate_font:n [1593](#), [1614](#)
- \ctex_ljt_save_alter_nate_family:Nnnn [1471](#), [1492](#), [1495](#)
- \ctex_ljt_save_alter_nate_seq:Nn [1442](#), [1447](#)
- \ctex_ljt_save_char_range:n [1640](#), [1647](#)
- \ctex_ljt_save_char_range:nn [1649](#), [1650](#)
- \ctex_ljt_select_alter_nate_font: [939](#), [2014](#)
- \ctex_ljt_select_font: [852](#), [2013](#)
- \ctex_ljt_set_alt_font:nnnn [807](#), [1561](#), [1564](#)
- \ctex_ljt_set_alter_nate_family:nn [1446](#), [1479](#)
- \ctex_ljt_set_alter_nate_family:nnn . [1491](#), [1499](#), [1501](#)
- \ctex_ljt_set_alter_nate_family:Nnnnn [1483](#), [1488](#), [1494](#)
- \ctex_ljt_set_alter_nate_family:nnTF [1091](#), [1432](#)
- \ctex_ljt_set_alter_nate_prop:n [1411](#), [1416](#)
- \ctex_ljt_set_alter_nate_shape:n [1552](#), [1559](#), [1566](#), [1571](#)
- \ctex_ljt_set_alter_nate_shape:nnN [1538](#), [1567](#)
- \ctex_ljt_set_alter_nate_shape:nnn [1541](#), [1547](#)
- \ctex_ljt_set_alter_nate_shape:nnnn [1549](#), [1550](#)
- \ctex_ljt_set_alter_nate_shape:Nnnnnnn . [1529](#), [1532](#)
- \ctex_ljt_set_family:nnn
..... [1083](#), [1253](#), [1300](#), [1310](#), [1328](#), [1335](#), [1341](#), [1347](#)
- \ctex_ljt_set_kanjiskip:N [1694](#), [1932](#)
- \ctex_ljt_set_math_letter:NN [1020](#), [1032](#), [1040](#)
- \ctex_ljt_set_xkanjiskip:N [1694](#), [2027](#)
- \ctex_ljt_subst_font: [892](#), [912](#)
- \ctex_ljt_swap_cs:NN [1505](#), [1512](#)
- \ctex_ljt_switch_family:n
..... [1158](#), [1159](#), [1169](#), [1226](#), [1254](#), [1308](#)
- \ctex_ljt_update_mathfont: [1381](#), [1384](#)
- \ctex_ljt_update_mathfont:n [1387](#), [1390](#), [1393](#)
- \ctex_ljt_use_jfont: [802](#), [908](#)
- \ctex_ljt_use_math_group:Nn [1006](#), [1037](#)
- \ctex_ljt_zero_globaldefs: [697](#), [845](#)
- \ctex_make_spa:nn [5049](#)
- \ctex_mono_jfm:n [1045](#), [1053](#), [2096](#)
- \ctex_parse_name:NN [1785](#), [1995](#), [5369](#), [5373](#)
- \ctex_parse_name:NNn [5374](#), [5378](#), [5435](#)
- \ctex_patch_boot:NnnnTF
..... [5330](#), [5335](#), [5340](#), [5355](#), [5360](#), [5363](#)
- \ctex_patch_cmd:Nnn
..... [76](#), [1715](#), [3226](#), [3232](#), [3238](#), [3310](#), [3313](#), [3317](#),
[3323](#), [3333](#), [3341](#), [3344](#), [3349](#), [3352](#), [4065](#), [4066](#), [4067](#), [5338](#)
- \ctex_patch_cmd_all:NnnnTF [3485](#), [5333](#)
- \ctex_patch_cmd_once:NnnnTF [1709](#), [1997](#), [5328](#)
- \ctex_patch_failure:N
.. [491](#), [746](#), [748](#), [1689](#), [2002](#), [3289](#), [3301](#), [3493](#), [5347](#), [5349](#)
- \ctex_plane_to_utfxvibe:Nn [523](#)
- \ctex_preto_cmd:NnnTF [738](#), [3286](#), [3297](#), [5353](#)
- \ctex_provide_font_hook:NNN [1775](#)
- \ctex_punct_map_bfseries:nn [596](#),
[4323](#), [4330](#), [4387](#), [4388](#), [4395](#), [4454](#), [4455](#), [4456](#), [4462](#),
[4534](#), [4535](#), [4536](#), [4543](#), [4615](#), [4616](#), [4624](#), [4677](#), [4680](#), [4687](#)
- \ctex_punct_map_family:nn
..... [591](#), [4320](#), [4321](#), [4322](#), [4329](#), [4384](#), [4385](#),
[4386](#), [4394](#), [4450](#), [4451](#), [4452](#), [4461](#), [4530](#), [4531](#), [4532](#),
[4541](#), [4542](#), [4611](#), [4612](#), [4613](#), [4623](#), [4676](#), [4679](#), [4681](#), [4686](#)
- \ctex_punct_map_itshape:nn [613](#), [4324](#), [4331](#),
[4389](#), [4396](#), [4453](#), [4463](#), [4533](#), [4544](#), [4614](#), [4625](#), [4678](#), [4688](#)
- \ctex_punct_map_series:nnn [600](#), [601](#), [604](#)
- \ctex_punct_set:n [573](#), [4319](#), [4328](#),
[4383](#), [4393](#), [4449](#), [4460](#), [4529](#), [4540](#), [4610](#), [4622](#), [4662](#), [4685](#)

- \ctex_punct_space:nn [622](#)
- \ctex_scheme_input:n [86](#), [3506](#), [3784](#)
- \ctex_select_size: .. [2134](#), [2141](#), [2144](#), [2145](#), [2166](#), [2224](#)
- \ctex_set:n [64](#), [266](#), [2104](#), [2124](#), [3808](#), [3810](#)
- \ctex_set:nn
 [32](#), [64](#), [147](#), [2114](#), [3509](#), [3567](#), [3595](#), [3627](#), [3662](#),
 [3679](#), [3693](#), [3709](#), [3717](#), [3725](#), [3733](#), [3740](#), [3741](#), [3742](#), [4171](#)
- \ctex_set_default_ccwd:Nn [117](#), [197](#), [2140](#), [2165](#)
- \ctex_set_deprecated_option:n
 [142](#), [176](#), [177](#), [231](#), [232](#), [233](#), [255](#), [260](#), [294](#), [295](#)
- \ctex_set_font_size:Nnn [3950](#), [3965](#),
 [3973](#), [3984](#), [3995](#), [3996](#), [3997](#), [3998](#), [3999](#), [4000](#), [4001](#),
 [4004](#), [4012](#), [4023](#), [4034](#), [4035](#), [4036](#), [4037](#), [4038](#), [4039](#), [4040](#)
- \ctex_set_ignorespaces: [564](#), [572](#)
- \ctex_set_upfamily:nnn ... [1720](#), [1767](#), [4343](#), [4344](#),
 [4345](#), [4346](#), [4408](#), [4409](#), [4410](#), [4411](#), [4475](#), [4476](#), [4477](#),
 [4478](#), [4479](#), [4480](#), [4555](#), [4556](#), [4557](#), [4558](#), [4559](#), [4560](#),
 [4637](#), [4638](#), [4639](#), [4700](#), [4701](#), [4702](#), [4703](#), [4704](#), [4705](#), [4706](#)
- \ctex_set_upfonts:nnnnnn
 [1760](#), [1770](#), [4336](#), [4401](#), [4468](#), [4693](#)
- \ctex_set_upmap:nnn .. [1734](#), [1762](#), [1763](#), [1764](#), [1765](#),
 [1768](#), [4481](#), [4482](#), [4549](#), [4550](#), [4551](#), [4552](#), [4636](#), [4707](#), [4708](#)
- \ctex_set_upmap_unicode:nnn
 [1747](#), [1769](#), [4553](#), [4554](#), [4630](#), [4632](#), [4634](#)
- \ctex_set_zhmap:n [430](#), [507](#), [1736](#), [1749](#)
- \ctex_titleps_hook: [3222](#), [3257](#), [3258](#)
- \ctex_titlesec_hook: [3130](#), [3194](#)
- \ctex_update_ccglue: [1924](#), [2189](#), [2209](#), [2252](#)
- \ctex_update_ccwd: [1910](#), [2183](#), [2260](#)
- \ctex_update_default_family: [50](#), [549](#), [1813](#), [1814](#)
- \ctex_update_em_unit: [1954](#), [2174](#), [2228](#)
- \ctex_update_kanjisize: [2041](#)
- \ctex_update_parindent: [2152](#), [2211](#), [2253](#)
- \ctex_update_size: [2146](#), [2147](#)
- \ctex_update_stretch: [2151](#), [2172](#)
- \ctex_update_xkanjiskip: [2019](#), [2039](#)
- \ctex_update_ziju: [2154](#), [2226](#)
- \ctex_use_package_hook:nn [5307](#), [5318](#), [5324](#)
- \ctex_use_zhmap: [433](#), [436](#), [440](#)
- \ctex_varioref_hook: [3462](#), [3496](#), [3504](#)
- \ctex_zhmap_case:nnn
 .. [132](#), [242](#), [247](#), [252](#), [4299](#), [4363](#), [4437](#), [4513](#), [4597](#), [4660](#)
- \ctex_zihao:n [3867](#), [3868](#)
- \ctex_ziju:n [2220](#), [2221](#)
- ctex internal commands:
- \g_ctex_after_end_preamble_hook_tl [5286](#), [5290](#), [5298](#)
- _ctex_assign_heading_name:nnn [2455](#)
- \l_ctex_autoindent_tl
 .. [130](#), [187](#), [192](#), [197](#), [2133](#), [2137](#), [2140](#), [2213](#), [2216](#), [2491](#)
- _ctex_calc_bounds:nn [5077](#), [5078](#), [5082](#)
- _ctex_ccglue: [1941](#), [1953](#)
- \l_ctex_ccglue_skip [1928](#),
 [1932](#), [1935](#), [1937](#), [1945](#), [1949](#), [2187](#), [2199](#), [2208](#), [2243](#), [2251](#)
- \c_ctex_CJK_cctab [513](#), [517](#)
- \c_ctex_class_tl
 .. [90](#), [92](#), [388](#), [393](#), [398](#), [403](#), [3771](#), [3777](#), [3779](#), [3780](#), [3782](#)
- _ctex_clear_fntef_color:n ... [1888](#), [1896](#), [1900](#), [1907](#)
- _ctex_cleveref_hook_aux:N [3473](#)
- \c_ctex_cmap_encoding_seq [447](#), [453](#)
- \l_ctex_ctexcap_options_clist [5238](#), [5239](#), [5248](#)
- _ctex_def_heading_keys:n [2410](#), [2485](#)
- _ctex_default_ccwd_aux:n [118](#), [119](#)
- _ctex_default_ccwd_aux:w [122](#), [126](#)
- \g_ctex_encoding_tl [128](#), [221](#), [224](#), [227](#),
 [374](#), [378](#), [495](#), [498](#), [551](#), [559](#), [1856](#), [2269](#), [3812](#), [3839](#), [3853](#)
- \g_ctex_end_preamble_hook_tl [5284](#), [5288](#), [5297](#)
- \c_ctex_engine_file_str [43](#), [2102](#)
- \c_ctex_engine_str [43](#)
- \l_ctex_family_default_init_tl [1816](#), [1834](#)
- _ctex_family_default_wrap:n [1819](#), [1835](#), [1838](#)
- \g_ctex_font_size_int [159](#),
 [170](#), [171](#), [172](#), [346](#), [350](#), [382](#), [3829](#), [3830](#), [3856](#), [3857](#), [3958](#)
- \c_ctex_font_size_prop [3870](#), [3881](#), [3925](#), [3952](#)
- \c_ctex_font_size_seq [3879](#), [3882](#), [3893](#)
- \l_ctex_font_size_tl [3870](#), [3871](#), [3917](#), [3918](#), [3952](#), [3953](#)
- \g_ctex_fontset_tl [230](#), [4089](#), [4092](#), [4093](#), [4100](#), [4104](#),
 [4105](#), [4108](#), [4109](#), [4112](#), [4118](#), [4123](#), [4140](#), [4142](#), [4147](#), [4162](#)
- _ctex_get_font_sizes:Nn [3917](#), [3920](#)
- _ctex_get_macro_meaning:w [5439](#)
- \l_ctex_heading_bool [271](#), [3767](#)
- \c_ctex_heading_level_prop [3433](#), [3438](#)
- \l_ctex_heading_skip [2512](#), [2519](#), [2521](#), [2523](#),
 [2524](#), [2535](#), [2536](#), [2539](#), [2541](#), [2542](#), [2545](#), [2990](#), [2991](#), [2992](#)
- \c_ctex_headings_cs_seq [3079](#), [3092](#), [3101](#), [3124](#)
- \c_ctex_headings_seq ... [2370](#), [2482](#), [3207](#), [3270](#), [3464](#)
- _ctex_hookto_cmd:Nnnnw [5355](#), [5360](#), [5518](#)
- _ctex_hookto_cmd_parameter:Nnnnw [5525](#), [5545](#)
- _ctex_hookto_cmd_parameterless:Nnnnw . [5524](#), [5530](#)
- _ctex_initial_heading:n [2380](#), [2484](#)
- _ctex_insert_indent: [2498](#), [2500](#), [2977](#)
- _ctex_left_hook_aux:nn [5543](#)
- \l_ctex_line_spread_fp
 [118](#), [180](#), [3831](#), [3832](#), [3858](#), [3859](#), [4044](#), [4046](#)
- \l_ctex_line_stretch_tl [2165](#), [2170](#), [2171](#), [2175](#)
- \l_ctex_ltj_add_alternate_bool [1238](#), [1250](#), [1268](#), [1275](#)
- _ctex_ltj_alternate_cs:n
 [1118](#), [1121](#), [1122](#), [1123](#), [1153](#), [1154](#), [1156](#), [1241](#),
 [1244](#), [1245](#), [1247](#), [1248](#), [1442](#), [1484](#), [1544](#), [1563](#), [1603](#), [1620](#)
- \l_ctex_ltj_alternate_family_tl [1508](#), [1511](#)
- \l_ctex_ltj_alternate_options_clist
 [1461](#), [1462](#), [1473](#), [1477](#)
- \l_ctex_ltj_alternate_prop
 [1087](#), [1427](#), [1428](#), [1431](#), [1441](#), [1449](#)
- \l_ctex_ltj_base_CJKfamily_tl
 [1088](#), [1102](#), [1151](#), [1544](#), [1563](#)
- \l_ctex_ltj_base_family_tl [71](#),
 [1119](#), [1146](#), [1152](#), [1490](#), [1530](#), [1583](#), [1600](#), [1606](#), [1607](#), [1619](#)
- \l_ctex_ltj_char_range_clist
 ... [1086](#), [1413](#), [1434](#), [1460](#), [1472](#), [1485](#), [1507](#), [1529](#), [1680](#)
- \g_ctex_ltj_char_range_prop [1536](#), [1637](#)
- \l_ctex_ltj_char_range_tl
 [1536](#), [1539](#), [1639](#), [1641](#), [1643](#), [1645](#), [1652](#)

`_ctex_ljt_chardef_text_cmd`: 706, 710
`_ctex_ljt_chardef_text_cmd_aux:N` 711, 712
`_ctex_ljt_check_family:n` 1113, 1440
`\l_ctex_ljt_cmd_tl` 705, 709, 711
`\l_ctex_ljt_current_font_tl`
..... 57, 855, 859, 860, 872, 918, 923
`\l_ctex_ljt_current_shape_tl` 941, 944, 947, 948
`_ctex_ljt_declare_char_range:nn` 1633, 1635
`\g_ctex_ljt_default_features_clist` 1106, 1351, 1352
`_ctex_ljt_family_csname:n` ... 1117, 1134, 1148, 1175
`\g_ctex_ljt_family_font_name_prop`
..... 1093, 1110, 1115, 1186, 1262, 1361, 1373
`\g_ctex_ljt_family_font_options_prop`
..... 1094, 1110, 1265
`\g_ctex_ljt_family_int` . 1228, 1230, 1522, 1523, 1525
`\g_ctex_ljt_family_name_prop`
..... 1110, 1119, 1140, 1172, 1490
`_ctex_ljt_family_unknown_warning:n` 1166, 1184, 1611
`\l_ctex_ljt_font_options_clist` 1090, 1095,
1096, 1097, 1103, 1139, 1266, 1267, 1270, 1481, 1482, 1486
`\g_ctex_ljt_fontname_seq` 877, 882, 885
`_ctex_ljt_fontspec:nnn` 1229, 1236, 1256
`\g_ctex_ljt_fontspec_family_tl`
..... 1138, 1141, 1147, 1152, 1177
`\g_ctex_ljt_fontspec_prop` 1224, 1252, 1257
`_ctex_ljt_gset_family_cs:nn` 1098, 1132
`_ctex_ljt_if_alt_set:nTF` 837, 934
`_ctex_ljt_if_jfont_math:w` 975, 978
`\l_ctex_ljt_jfm_tl` 1045, 1064, 1066, 1108
`_ctex_ljt_ljt_if_jfont:w` 969, 972
`\c_ctex_ljt_math_fam_int` 1072, 1073
`\c_ctex_ljt_math_family_tl`
..... 1395, 1397, 1399, 1402, 1406
`\c_ctex_ljt_math_tl`
... 1067, 1068, 1070, 1072, 1347, 1386, 1396, 1401, 1405
`_ctex_ljt_msg_def_family_map:n` 1199, 1201
`_ctex_ljt_msg_family_map:n` 1131, 1198, 1212
`_ctex_ljt_noembed_wrap:n` 1080, 1082
`_ctex_ljt_pass_args:nnnn`
..... 1283, 1299, 1309, 1315, 1327, 1334, 1340, 1346
`_ctex_ljt_patch_external_font:w` 904, 937
`_ctex_ljt_patch_text_cmd:NN` 732, 735
`_ctex_ljt_pickup_alt_font:Nn` 830, 965
`_ctex_ljt_pop_fontname:` 871, 880, 959
`_ctex_ljt_post_arg:w` 1286, 1292
`_ctex_ljt_push_alternate_prop:n` 1417, 1418
`_ctex_ljt_push_alternate_prop:nn` . 1422, 1425, 1681
`_ctex_ljt_push_fontname:n` 867, 875, 956
`\c_ctex_ljt_range_max_int` 1664, 1668
`\c_ctex_ljt_range_min_int` 1663, 1667
`_ctex_ljt_range_normalization:nn` . 1554, 1653, 1655
`\g_ctex_ljt_reset_alternate_prop`
..... 1124, 1605, 1617, 1625
`_ctex_ljt_save_alternate_auxi:w` 1450, 1452
`_ctex_ljt_save_alternate_auxii:w` . 1455, 1456, 1458
`_ctex_ljt_save_alternate_shape:Nn` 1543, 1562, 1579
`_ctex_ljt_select_font_aux:` 856, 862
`_ctex_ljt_set_alternate_family:n` 1142, 1149
`_ctex_ljt_set_alternate_family_aux:nnn` 1436, 1445
`_ctex_ljt_set_family_aux:nnn` 1435, 1438
`_ctex_ljt_tmp:w` 1514, 1516, 1517
`\l_ctex_ljt_tmp_clist` 1420, 1421, 1423, 1430
`\l_ctex_ljt_tmp_tl` 882, 883, 1089,
1090, 1115, 1126, 1129, 1263, 1271, 1305, 1308, 1310,
1363, 1368, 1386, 1387, 1389, 1390, 1464, 1466, 1467, 1474
`\g_ctex_ljt_unknown_family_seq` ... 1188, 1190, 1195
`_ctex_ljt_update_family_uid:N` ... 1096, 1481, 1519
`_ctex_ljt_use_global_options:N`
..... 1097, 1104, 1462, 1482
`\c_ctex_msyh_suffix_tl`
..... 4650, 4665, 4666, 4672, 4673, 4697, 4698
`\c_ctex_package_cctab` 77, 81, 89, 519
`_ctex_package_loaded_warning:nn` 5256, 5259
`\l_ctex_parameter_str` 5436, 5455, 5476, 5523
`\l_ctex_parameter_tl` 5476, 5481, 5492
`_ctex_parse_name:nNNNnN` 5388, 5397
`_ctex_parse_name:w` 5410, 5416, 5425
`_ctex_patch_cmd:Nnnnw` 5330, 5335, 5340, 5494
`_ctex_patch_false:w` 5366, 5372, 5509, 5511, 5528, 5554
`_ctex_patch_rebuild:Nn` 5467, 5473, 5506, 5547, 5551
`_ctex_patch_replace:nnnTF` 5501, 5513
`_ctex_patch_rescan:NNn` 5475, 5476, 5477, 5485
`_ctex_patch_toc_width:n` 3281
`_ctex_patch_true:w` 5365, 5371, 5507, 5540, 5552
`_ctex_plus_key_aux:nn`
..... 2437, 2439, 2440, 2441, 2442, 2443, 2444, 2447
`\l_ctex_prefix_str` 5436, 5454, 5475, 5532, 5533
`\l_ctex_prefix_tl` 5475, 5480, 5491
`_ctex_provide_font_hook_aux:NNNN` 1777, 1780
`\l_ctex_punct_bounds_clist` ... 5063, 5069, 5075, 5081
`\c_ctex_punct_family_clist` 575, 585
`\l_ctex_punct_font` 5061, 5062
`\c_ctex_punct_seq` 5041, 5064
`\l_ctex_punct_tl`
..... 257, 546, 547, 632, 1053, 2093, 2094, 2095, 2096
`_ctex_rebuild_cmd:w` ... 5467, 5468, 5472, 5547, 5548
`\l_ctex_replacement_str`
..... 5436, 5456, 5477, 5515, 5516, 5550
`\l_ctex_replacement_tl` 5477, 5482, 5493
`_ctex_restore_selectfont:` 1970, 1992, 2001
`_ctex_right_hook_aux:nn` 5544
`_ctex_save_bounds:n` 5065, 5073
`_ctex_save_cmap:Nn` 466, 470
`_ctex_save_font_size:nn` 3881
`\l_ctex_scheme_tl` 282, 3506, 3784
`\g_ctex_section_depth_int`
..... 99, 276, 278, 3707, 3723, 3739
`\c_ctex_section_headings_seq` . 2366, 2372, 3034, 3148
`_ctex_set_font_size:nnNn` 3953, 3956
`\g_ctex_spa_iow` 5051, 5054, 5056, 5066
`\c_ctex_std_class_tl` 114, 3755, 3770, 3771, 3837, 3860

<code>\g__ctex_std_options_clist</code>	135, 351, 363, 384, 385, 389, 394, 399, 404
<code>__ctex_titlesec_format:n</code>	3144, 3152, 3155
<code>__ctex_titlesec_format:Nn</code>	3134, 3158, 3160
<code>__ctex_titlesec_format_explicit:Nn</code>	3135, 3170
<code>__ctex_titlesec_hook:n</code>	3144, 3148, 3150
<code>__ctex_titlesec_spacing:Nn</code>	3153, 3180
<code>__ctex_titlesec_spacing:nnnnn</code>	3181, 3182
<code>__ctex_tmp:N</code>	1966, 1995
<code>__ctex_tmp:w</code>	5376, 5429, 5440, 5461
<code>\l__ctex_tmp_box</code>	60, 1913, 1914, 2982, 2983, 2984, 3274, 3278
<code>\l__ctex_tmp_dim</code>	60, 2175, 2176, 2185, 2188, 2195, 2233, 2235, 2237, 2246, 2248
<code>\l__ctex_tmp_int</code>	60, 527, 529, 530, 532, 536, 538, 2194, 2196, 2197, 2203, 2204, 2235, 2236, 2237, 2240, 2241, 2246, 2248, 3033, 3036, 3040
<code>\l__ctex_tmp_tl</code>	60, 335, 342, 357, 358, 472, 473, 526, 528, 1968, 1997, 2412, 2467, 2480, 2487, 3433, 3434, 3836, 3837, 3925, 3926
<code>__ctex_update_stretch_auxi:</code>	2177, 2180
<code>__ctex_update_stretch_auxii:</code>	2178, 2192
<code>__ctex_use_package_hook_aux:N</code>	5310, 5312
<code>\c__ctex_version_tl</code>	3, 6, 7, 8, 9, 20, 23
<code>__ctex_write_family:nn</code>	5053, 5058
<code>\l__ctex_xkanjiskip_skip</code>	2019
<code>\l__ctex_xkanjiskip_tl</code>	2026, 2031, 2032
<code>\c__ctex_zero_tl</code>	2149, 2156, 2158
<code>\g__ctex_zhmap_tl</code>	432, 434, 443
<code>\g__ctex_zhmCJK_bool</code> .	129, 241, 246, 251, 494, 550, 1774
<code>\l__ctex_ziju_dim</code> 2229, 2230, 2234, 2245, 2247, 2251, 2255	
<code>\l__ctex_ziju_tl</code>	2149, 2157, 2158, 2223, 2229
<code>\ctexbibname</code>	2339, 2359
<code>\CTEXcounter</code>	2274
<code>\CTEXdigits</code>	29, 2284
<code>CTEXfilltwosides</code>	34, 35
<code>\CTEXifname</code> 26, 2527, 2617, 2659, 2801, 3013, 3021, 3029, 3265, 3312, 3315, 3319, 3331, 3335, 3343, 3346, 3351, 3354	
<code>\CTEXindent</code>	34, 2256
<code>\CTEXnoindent</code>	34, 2256
<code>\CTEXnumber</code>	29, 2284
<code>\CTEXnumberline</code>	3027, 3047, 3585
<code>\CTEXoptions</code>	34, 2106
<code>\ctexset</code>	5, 2103, 2113, 2123
<code>\CTEXsetfont</code>	34, 2144
<code>\CTEXsetup</code>	34, 2106
<code>\CTEXsout</code>	34, 35
<code>\ctexspadef</code>	622, 5068
<code>\CTEXthechapter</code>	26, 2744, 3022, 3200, 3331, 3351
<code>\CTEXtheparagraph</code>	26
<code>\CTEXthepart</code>	26, 3014, 3205
<code>\CTEXthesection</code>	26, 3312, 3319, 3335, 3343, 3354
<code>\CTEXthesubparagraph</code>	26
<code>\CTEXthesubsection</code>	26, 3315, 3346
<code>\CTEXthesubsubsection</code>	26
<code>\CTEXunderdblline</code>	34, 35
<code>\CTEXunderdot</code>	34, 35
<code>\CTEXunderline</code>	34, 35
<code>\CTEXunderwave</code>	34, 35
<code>\CTEXxout</code>	34, 35
<code>\CurrentOption</code>	363
<code>\cwd</code>	2041
D	
<code>declarecharrange</code>	30, 1627
<code>\DeclareErrorKanjiFont</code>	1703
<code>\DeclareFontEncoding</code>	1055
<code>\DeclareFontFamily</code>	480, 1062, 4182, 4183, 4184, 4185, 4186, 4187
<code>\DeclareFontShape</code>	1063, 1065, 1724, 1725, 1728, 1729, 1730, 1731, 4196, 4197, 4198, 4199, 4200, 4201, 4202, 4203, 4204, 4207, 4208, 4209, 4210, 4211, 4212, 4213, 4214, 4215, 4218, 4219, 4220, 4221, 4224, 4225, 4226, 4227, 4232, 4233, 4234, 4235, 4236, 4237, 4238, 4239, 4240, 4243, 4244, 4245, 4246, 4247, 4248, 4249, 4250, 4251, 4254, 4255, 4256, 4259, 4260, 4261, 4266, 4267, 4268, 4269, 4270, 4271, 4272, 4273, 4274, 4277, 4278, 4279, 4280, 4281, 4282, 4283, 4284, 4285, 4288, 4291
<code>\DeclareFontSubstitution</code>	1058
<code>\DeclareKanjiFamily</code>	1722, 1723, 4188, 4189, 4190, 4191, 4192, 4193
<code>\DeclareKanjiSubstitution</code>	1704, 1705
<code>\DeclareMathSizes</code>	3918
<code>\DeclareSymbolFont</code>	1068, 1396, 1706
<code>\def</code>	416, 2590, 2631, 2667, 2688, 2704, 2735, 2776, 2786, 2811, 2831, 2857, 2861, 2898, 2913, 2929, 2930, 2934, 3978, 3989, 4017, 4028, 4756, 4792
<code>\defaultCJKfontfeatures</code>	1325
<code>\defbeamertemplate</code>	3360, 3379, 3398
<code>\defbeamertemplatealias</code>	3417, 3418, 3419
<code>\detokenize</code>	417
dim commands:	
<code>\dim_abs:n</code>	2247
<code>\dim_add:Nn</code>	2230
<code>\dim_compare:nNnTF</code>	2176, 2215, 2231, 2238, 2502, 2511, 2991
<code>\dim_eval:n</code>	2201, 2246
<code>\dim_max:nn</code>	3277
<code>\dim_min:nn</code>	2247
<code>\dim_new:N</code>	63, 1923, 2042, 2043, 2044, 2255
<code>\dim_set:Nn</code>	1914, 1955, 1956, 1957, 2175, 2185, 2216, 2229, 2233, 2261, 2497, 2518, 2973, 3275, 4048
<code>\dim_sub:Nn</code>	2237
<code>\dim_to_decimal:n</code>	3889, 3890, 3927
<code>\dim_to_decimal_in_unit:nn</code>	5088
<code>\dim_use:N</code>	123, 4048
<code>\dim_zero:N</code>	2267
<code>\c_max_dim</code>	2176, 2511
<code>\c_zero_dim</code>	2188, 2201, 2215, 2231, 2238, 2425, 2502, 2991, 3740
<code>\directlua</code>	675, 679
<code>\do</code>	421

\document 5291, 5294

E

\edef 4775, 4776
 \else 411, 414, 415, 419, 1711,
 1712, 1716, 1717, 2562, 2571, 2580, 2596, 2601, 2637,
 2642, 2723, 2731, 2747, 2753, 2759, 2772, 2782, 2842,
 2847, 2866, 2871, 2892, 2897, 2906, 2928, 2944, 2957,
 3548, 3576, 4787, 4788, 4814, 4838, 4887, 4919, 4951, 5000
 else commands:
 \else: 497, 970, 1942, 3100, 3316
 \em 1709
 \emminershape 1711, 1712, 1717, 1719
 \end 3376, 3395, 3414, 5150
 \endcsname 410, 413, 414, 2849, 2858, 2859,
 2860, 2867, 2887, 2888, 2890, 2893, 2902, 2903, 2904,
 2907, 3490, 3491, 4782, 4784, 4785, 4786, 4789, 4791, 4795
 \endgroup 411, 428, 2622, 2663, 2679, 2700, 2805,
 2824, 2889, 2927, 2952, 3378, 3397, 3416, 4775, 4783, 4798
 \endinput 411, 4780
 \endlinechar 4754, 4755
 \everypar 2846, 2947, 2959
 \EverySelectfont 2005, 2008
 exp commands:
 \exp_after:wN 122,
 711, 888, 904, 906, 970, 975, 3181, 3871, 3918, 3953, 5445
 \exp_args:cc 3458
 \exp_args:Nc 1777, 3153, 3158, 5399
 \exp_args:Nco 3263
 \exp_args:NNc 732, 854, 5309
 \exp_args:Nnc 108
 \exp_args:NNNo 4047
 \exp_args:NNo 2487
 \exp_args:Nnx 3471
 \exp_args:No 357, 528, 983, 1422, 3177, 5065, 5461
 \exp_args:Nx 265, 556, 629, 2220, 3867, 4046
 \exp_last_unbraced:Ne 5410
 \exp_last_unbraced:NNNo 125
 \exp_last_unbraced:NNo 1996
 \exp_not:N 476, 558, 559,
 560, 726, 727, 923, 955, 1058, 1140, 1141, 1471, 1824,
 1825, 1826, 1838, 2386, 2387, 2388, 2393, 2395, 2396,
 2397, 2398, 2401, 2402, 2405, 2417, 2418, 2419, 2420,
 2421, 2430, 2431, 2432, 2434, 2435, 2449, 2451, 2453,
 3039, 3041, 3042, 3043, 3044, 3165, 3166, 3175, 3176,
 3213, 3215, 3216, 3471, 3887, 4118, 5390, 5391, 5392, 5480
 \exp_not:n 121, 152,
 1137, 1138, 1139, 1270, 1271, 1472, 1473, 1474, 1499,
 1555, 1556, 1584, 1819, 1839, 1961, 2436, 3184, 3459,
 5240, 5294, 5365, 5366, 5408, 5480, 5481, 5482, 5537, 5538
 \expandafter 410, 411, 413, 414, 422, 426, 3491,
 4757, 4758, 4782, 4783, 4784, 4785, 4786, 4791, 4795, 4796
 \ExplSyntaxOff 485,
 653, 739, 743, 1710, 1998, 3286, 3359, 3487, 5094, 5343
 \ExplSyntaxOn 690, 3420, 5034

F

\familydefault 1822
 fancyhdr 33, 317
 \fangsong 4723
 \fi 411, 414,
 415, 423, 424, 425, 427, 2553, 2558, 2564, 2573, 2582,
 2585, 2600, 2605, 2615, 2626, 2641, 2646, 2657, 2683,
 2707, 2715, 2716, 2719, 2723, 2733, 2751, 2757, 2763,
 2775, 2785, 2789, 2798, 2799, 2809, 2814, 2828, 2832,
 2838, 2844, 2851, 2853, 2875, 2876, 2894, 2908, 2911,
 2932, 2941, 2960, 2961, 3311, 3314, 3318, 3328, 3329,
 3334, 3342, 3345, 3350, 3353, 3372, 3391, 3410, 3548,
 3576, 4787, 4788, 4797, 4833, 4857, 4914, 4946, 4970, 5027
 fi commands:
 \fi: . 510, 543, 562, 970, 1811, 1942, 2003, 3103, 3320, 3336
 \figurename 2320
 figurename 12
 file commands:
 \g_file_curr_name_str 644
 \file_if_exist:nTF 48, 4097, 4112, 4500, 4651, 4654
 \file_if_exist_input:nTF 92
 \file_input:n 78, 93, 95, 514
 \file_input_stop: 5280
 \fmtversion 5272
 fntef 34, 35, 298
 \font 4182, 4183, 4184, 4185, 4186, 4187
 fontset 7, 230, 4131
 \fontsize 3871
 fontspec commands:
 \fontspec_gset_family:Nnn 1138
 \fontspec_set_family:Nnn 1508
 fontspec internal commands:
 _fontspec_fontname_wrap:n 62, 1080
 \l_fontspec_rmfamily_encoding_tl 76
 \footnotesep 4047
 \footnotesize 3984, 4023, 4047
 fp commands:
 \fp_eval:n 2223, 5084
 \fp_if_nan:nTF 3831, 3858, 4044
 \fp_set:Nn 3832, 3859
 \fp_use:N 2156, 4046
 \c_nan_fp 181
 \c_zero_fp 2156

G

GBK 6, 35, 218
 \gdef . 418, 424, 3065, 3066, 3067, 3068, 3071, 3072, 3073, 3074
 \global 2727, 2946, 2949
 \globaljfont 906
 \glueexpr 3366, 3377, 3385, 3396, 3404, 3415
 group commands:
 \group_begin: 480, 696, 769, 854, 864,
 890, 916, 954, 976, 1022, 1085, 1136, 1503, 1581, 1599,
 1616, 1818, 1965, 2045, 2392, 2400, 2465, 2989, 3089,
 3222, 3281, 4047, 5060, 5288, 5309, 5367, 5375, 5384, 5439

\group_end: 484, 493, 701, 851, 854, 869, 894,
929, 958, 980, 1026, 1100, 1143, 1509, 1585, 1612, 1623,
1830, 1996, 2065, 2403, 2407, 2487, 2993, 3104, 3245,
3285, 4047, 5071, 5288, 5309, 5387, 5434, 5462, 5496, 5520
\group_insert_after:N 926, 927
\gtfamily 1711, 1716

H

hbox commands:

\hbox_set:Nn 1913, 2982, 3274
heading 9, 271
\heiti 4723

hook commands:

\hook_gput_code:nnn 5275, 5277, 5279
\hskip 2884, 2900, 2922, 2929, 2955, 3342, 3345
\hspace 2595, 2636, 3014, 3022, 3205
\Huge 2660, 2697, 2802, 2821, 3552, 3583, 4001, 4040
\huge 2618,
2656, 2676, 2797, 3534, 3551, 3555, 3581, 3589, 4000, 4039
hyperref 34, 322
\hypersetup 327, 328, 1847
\hyphenchar 4182, 4183, 4184, 4185, 4186, 4187

I

\if 417

if commands:

\if_bool:N 494, 550, 1774
\if_cs_exist:N 523, 1964
\if_false: 114
\if_meaning:w 1941
\if_true: 113
\ifctexpdf 112
\ifdim 2836, 2879, 2917, 2936
\ifnum 415, 2591,
2613, 2632, 2655, 2736, 2795, 2862, 2892, 2906, 3311,
3314, 3318, 3325, 3326, 3334, 3342, 3345, 3350, 3353, 4787
\ifodd 412, 424, 2558, 2560, 2585, 2592, 2626,
2633, 2683, 2707, 2729, 2740, 2789, 2809, 2814, 2828,
2840, 2851, 2867, 2880, 2918, 2937, 2941, 3370, 3389, 3408
\ifx 410, 413, 414, 4785, 4786, 4791
\ifzhmappdf 4783, 4812, 4836, 4860, 4917, 4949, 4973
\ignorespaces 2962
\immediate 4794
indent 33, 200
\indexname 2323
indexname 12
\input 5033, 5097
\insertcontinuationtext 2332, 2341, 2348
\insertpart 3374, 3375
\insertromanpartnumber 3364, 3521
\insertsection 3393, 3394
\insertsectionnumber 3383, 3619
\insertsubsection 3412, 3413
\insertsubsectionnumber 3402, 3648

int commands:

\int_case:nn 382, 3958

\int_compare:nNnTF
..... 529, 714, 2197, 2240, 3707, 3723, 3739, 3829, 3856
\int_const:Nn 1072, 1667, 1668
\int_div_truncate:nn 84, 536
\int_eval:n 1659, 1660, 1663, 1664, 3435
\int_from_hex:n 528
\int_gincr:N 1228, 1522
\int_gset:Nn
.. 101, 161, 170, 171, 172, 276, 278, 346, 350, 3830, 3857
\int_incr:N 3036
\int_mod:nn 538
\int_new:N 61, 100, 160, 1525
\int_set:Nn 520, 527, 2194, 2235, 2492
\int_sub:Nn 532, 2241
\int_to_Hex:n 530, 535, 537
\int_use:N 477, 1230, 1523, 3040, 5065
\int_zero:N 3033
\c_zero_int 2197

int internal commands:

__ctexpar 2923
\interlinepenalty . 2610, 2652, 2672, 2693, 2817, 2885, 2923

low commands:

\iow_close:N 5054
\iow_indent:n 5245
\iow_new:N 5056
\iow_now:Nn 5066
\iow_open:Nn 5051
\itemsep 3981, 3992, 4020, 4031
\itshape 1711, 1712, 1716, 1717

J

\jfam 1014, 1073, 1708

K

\kaishu 4723
\kanjifamily 1702, 1806, 1807, 1808
\kanjifamilydefault 1809

keys commands:

\l_keys_choice_tl 266
\keys_define:nn 66, 69, 1077, 1409, 1519, 1673
\keys_if_exist:nnTF 1671
\l_keys_key_tl 139, 141, 155
\keys_set:nn 71, 73
\keys_set_known:nn 3787
\keys_set_known:nnN 1089

L

\labelformat 3471, 3495
\languagealias 3816, 3821, 3842, 3847
\LARGE 3999, 4038
\Large 2614, 3532, 3538, 3606, 3610, 3998, 4037
\large 3631, 3997, 4036
\lastbox 2950
\leavevmode 2553, 2832

left commands:

\c_left_brace_str 5431
\leftmargin 3978, 3989, 4017, 4028

- `\leftmargini` 3978, 3989, 4017, 4028
`\let` 420, 423, 2865, 2874, 3971, 4010, 4778, 4784
`\linespread` 4046
`linespread` 11, 180
`linestretch` 11, 2161
`\linewidth` 2195, 2203, 2234
`\lishu` 4723
`\listfigurename` 2318
`listfigurename` 12
`\listtablename` 2319
`listtablename` 12
`\LoadClass` 390, 395, 400, 405
`\long` 4792
`\ltjdefcharrange` . 654, 655, 656, 658, 663, 664, 668, 671, 672
`LTJFONTUID` 1519
`\ltjgetparameter` 1918, 1945, 2023, 2037
`\ltjlineendcomment` 652
`\ltjsetkanjiskip` 1695
`\ltjsetmathletter` 1024
`\ltjsetparameter` 674, 677, 683, 1692
`\ltjsetxkanjiskip` 1697
lua commands:
`\lua_now:n` 771, 2047
`\LuaTeXjversion` 838
- M**
- `\makeatletter` 739, 743
`\MAKESPA` 5057, 5098
`\markboth` 2526, 2620, 2649
`\mathgroup` 1014
max commands:
`\c_max_char_int` 1668
`\mcfamily` 1716
`\mdefault` 1059, 1063, 1069, 1397, 1402
mode commands:
`\mode_if_math:TF` 1008
msg commands:
`\msg_critical:nnn` 53, 644, 4071
`\msg_error:nn` 4151
`\msg_error:nnn` 19, 22, 38, 41, 635, 648, 651, 2305, 3872, 3954
`\msg_error:nnnn` 4117, 4146
`\msg_fatal:nn` 5039
`\msg_new:nnn` . 138, 140, 154, 156, 372, 638, 1130, 1196,
1278, 2336, 3761, 4072, 4125, 4160, 5037, 5241, 5261, 5351
`\msg_new:nnnn` 12, 30, 45, 106, 2307, 3874, 4127, 4154
`\msg_warning:nn` 223, 377, 1273, 2086, 2097, 2358
`\msg_warning:nnn` 143, 1191, 3780, 4138, 5247, 5260, 5350
`\msg_warning:nnnn` 151, 1126, 4107
- N**
- `\newCJKfontfamily` 1297
`\NewDocumentCommand`
1157, 1292, 1297, 1303, 1313, 1319, 1325,
1332, 1338, 1344, 1350, 1547, 1647, 1701, 2104, 2107,
2117, 2144, 2219, 2256, 2263, 2284, 2286, 2455, 3866,
4051, 4724, 4725, 4726, 4727, 4729, 4730, 4732, 4742, 4743
`\newfontfeature` 1074, 1075, 1076
`\newpage` 2709, 2714
`\ngostype` 44
`\nobreak` 2614,
2623, 2680, 2797, 2802, 2806, 2821, 2825, 2938, 3533, 3582
`\nobreakspace` 2614, 2656
`nocap` 33, 294
`\noexpand` 4776, 4778, 4779, 4780
`nofonts` 33
`noindent` 33
`nopunct` 33
`\normalem` 1887
`\normalfont` 1329, 1336, 1342, 1380,
1790, 1812, 2490, 2610, 2652, 2672, 2693, 2792, 2817, 3044
`\normalsize` 3665, 3682, 3696, 3965, 4004, 4042, 4050
`nospace` 33
`nozhmap` 33
`\null` 2583, 2712
`\numberline` 2746, 2893, 2907, 3022, 3030, 3216, 3281
- O**
- `\onecolumn` 2578
- P**
- `\pagestyle` 3807
`\par` 2517, 2554, 2614, 2620, 2621, 2656,
2660, 2662, 2676, 2678, 2697, 2699, 2797, 2802, 2804,
2821, 2823, 2833, 2888, 2926, 2938, 3365, 3366, 3374,
3377, 3384, 3385, 3393, 3396, 3403, 3404, 3412, 3415,
3511, 3533, 3544, 3562, 3570, 3582, 3622, 3624, 3656, 3658
`paragraph/afterskip` 35
`paragraph/beforeskip` 35
`paragraph/numbering` 16
`paragraph/runin` 22
`\parindent` 2215, 2216,
2238, 2259, 2261, 2266, 2267, 2610, 2672, 2792, 2817, 3741
`\parsep` 3980, 3981, 3991, 3992, 4019, 4020, 4030, 4031
`\part` 2551
`part/beforeskip` 35
`part/fixbeforeskip` 35
`part/numbering` 16
`part/pagestyle` 21, 2465
`\partmark` 2525, 2608, 2650
`\partname` 2614, 2656, 3364, 3516
`\PassOptionsToClass` 389, 394, 399, 404
`\PassOptionsToPackage` 495, 1851, 2269, 5249
`\pdfmapline` 4861, 4862, 4863, 4864, 4865, 4866,
4867, 4868, 4869, 4870, 4871, 4872, 4873, 4874, 4875,
4876, 4877, 4878, 4879, 4880, 4881, 4882, 4883, 4884,
4885, 4886, 4974, 4975, 4976, 4977, 4978, 4979, 4980,
4981, 4982, 4983, 4984, 4985, 4986, 4987, 4988, 4989,
4990, 4991, 4992, 4993, 4994, 4995, 4996, 4997, 4998, 4999
`\pdfoutput` 415, 4787
`\pdfstringdefDisableCommands` 4054
`\pingfang` 4723
prg commands:
`\prg_do_nothing:`
441, 1018, 1497, 1582, 1963, 2073, 3006, 5472

- `\prg_generate_conditional_variant:Nnn` 1183
`\prg_new_conditional:Npnn` 932, 1938
`\prg_new_protected_conditional:Npnn` 1170, 5442
`\prg_return_false:` ... 935, 1180, 1942, 1946, 1950, 5452
`\prg_return_true:` 934, 1173, 1178, 1942, 1946, 1950, 5457
`\primitive` 5150
`\ProcessKeysOptions` 369
`\proofname` 2310, 2325
`proofname` 13
prop commands:
`\prop_clear:N` 1087
`\prop_const_from_keyval:Nn` 3438
`\prop_get:NnN` 1265
`\prop_get:NnNTF`
... 1172, 1224, 1262, 1490, 1536, 3433, 3870, 3925, 3952
`\prop_gpop:NnNTF` 1115, 1119, 1617
`\prop_gput:Nnn` . 1093, 1094, 1140, 1252, 1605, 1641, 3887
`\prop_gremove:Nn` 1124
`\prop_if_empty:NTF` 1186, 1361, 1441
`\prop_map_break:n` 1375
`\prop_map_inline:Nn` 1373, 1449
`\prop_new:N` 1110, 1111, 1112, 1257, 1431, 1625, 1646, 3881
`\prop_put:Nnn` 1428
`\prop_remove:Nn` 1427
`\protect` 2539, 2545, 2746,
2767, 2768, 2893, 2907, 3022, 3030, 3215, 3227, 3229,
3233, 3235, 3239, 3241, 3249, 3254, 3266, 3267, 5403, 5404
`\ProvideDocumentCommand` 2525
`\ProvidesFile` 4790, 4799, 4801, 4803, 4805, 4807, 4809
`\providetranslation`
..... 5155, 5156, 5157, 5158, 5159, 5160, 5161,
5162, 5163, 5164, 5165, 5166, 5167, 5168, 5169, 5170,
5171, 5172, 5173, 5174, 5175, 5176, 5177, 5178, 5179,
5180, 5181, 5182, 5183, 5184, 5185, 5186, 5187, 5188,
5189, 5190, 5191, 5192, 5193, 5194, 5195, 5196, 5197,
5198, 5199, 5200, 5201, 5202, 5203, 5204, 5205, 5206,
5207, 5208, 5209, 5210, 5211, 5212, 5213, 5214, 5215,
5216, 5217, 5218, 5219, 5220, 5221, 5222, 5223, 5224,
5225, 5226, 5227, 5228, 5229, 5230, 5231, 5232, 5233, 5234
`punct` 10, 33, 257, 2089
`\punctstyle` 547, 2094

Q

`\quad` 2858,
3311, 3312, 3314, 3315, 3318, 3319, 3331, 3335, 3343,
3346, 3351, 3354, 3539, 3590, 3599, 3632, 3666, 3683, 3697
quark commands:
`\q_mark` 1450, 1458, 5446, 5449
`\q_stop`
123, 126, 969, 972, 975, 978, 5411, 5416, 5425, 5446, 5449
`\quotation` 4064

R

`\raggedright` 2610, 2672, 2792, 2817, 3531, 3580
`\refname` 2326, 2331, 2338, 2347, 2355, 2356
`refname` 13
`\refstepcounter` ... 2594, 2635, 2742, 2869, 3453, 3498, 3499

`\relax` 410, 413, 414, 417, 2632, 2655, 2834, 2870, 2878, 2884,
2900, 2916, 2929, 2935, 2954, 3342, 3345, 3366, 3377,
3385, 3396, 3404, 3415, 4749, 4754, 4757, 4785, 4786, 4791
`\renewcommand` 2552, 2568, 2722
`\RenewDocumentCommand` 3499
`\RequirePackage`
... 27, 56, 57, 59, 303, 309, 320, 330, 496, 499, 500,
501, 628, 646, 649, 763, 768, 1771, 1860, 1869, 2270, 3861
`\RequirePackageWithOptions` 5250
`resetalternatefont` 31, 1588
reverse commands:
`\reverse_if:N` 550, 1774
`\rmdefault` 1824
`\rmfamily` 1787

S

scan commands:
`\scan_stop:`
... 71, 123, 727, 906, 1014, 1583, 2186, 2195, 5061, 5290
`scheme` 10, 282
`\scriptsize` 3995, 4034
`\secdef` 2565, 2587, 2734
`secnumdepth` 16, 3422
`section/afterskip` 35
`section/beforeskip` 35
`section/numbering` 16
`section/runin` 22
`\sectionname` 3383, 3616
`\selectfont` 1164, 1608,
1621, 1702, 1787, 1788, 1789, 1995, 2002, 2016, 2146, 3871
seq commands:
`\seq_const_from_clist:Nn` . 447, 2367, 2377, 3079, 5041
`\seq_gpop:NNTF` 882
`\seq_gpush:Nn` 877
`\seq_gput_left:Nn` 2373, 2374
`\seq_gput_right:Nn` 1190, 3893
`\seq_gset_eq:NN` 2372
`\seq_if_in:NnTF` 453, 1188
`\seq_map_function:NN` 3148, 3270
`\seq_map_inline:Nn`
... 2482, 3034, 3082, 3092, 3101, 3124, 3207, 3464, 5064
`\seq_new:N` 885, 1195, 2371, 3882
`\seq_use:Nnnn` 3879
`\setbox` 2950
`\setCJKfamilyfont` 1209, 1297, 4311,
4313, 4315, 4317, 4353, 4354, 4355, 4356, 4375, 4377,
4379, 4381, 4424, 4426, 4428, 4430, 4443, 4444, 4445,
4446, 4447, 4448, 4489, 4491, 4492, 4493, 4494, 4495,
4523, 4524, 4525, 4526, 4527, 4528, 4569, 4570, 4571,
4572, 4583, 4584, 4585, 4586, 4587, 4588, 4589, 4605,
4607, 4609, 4645, 4646, 4647, 4668, 4669, 4670, 4671,
4672, 4674, 4675, 4714, 4715, 4716, 4717, 4718, 4719, 4720
`\setCJKmainfont` 1205, 1325, 4301, 4349, 4365,
4414, 4439, 4485, 4515, 4565, 4575, 4599, 4642, 4663, 4711
`\setCJKmathfont` 1325

`\setCJKmonofont` 1207, 1325, 4309, 4352, 4373,
 4422, 4442, 4488, 4522, 4568, 4582, 4603, 4644, 4667, 4713
`\setCJKromanfont` 1331, 1357
`\setCJKsansfont` 1206, 1325, 4307, 4351, 4371,
 4420, 4441, 4487, 4521, 4567, 4581, 4601, 4643, 4665, 4712
`\setcounter` 3434, 3435
`\SetSymbolFont` 1070, 1401, 1405, 1707
`\sfdefault` 1825
`\sffamily` 1788
`\shapedefault`
 ... 1059, 1063, 1065, 1069, 1071, 1397, 1399, 1402, 1406

skip commands:

`\skip_horizontal:N` 1928, 2992
`\skip_if_eq:nnTF` 1945, 1949, 2022, 2536, 2542
`\skip_new:N` 1937, 2035, 2521
`\skip_set:Nn` 1918, 1921,
 2026, 2036, 2187, 2199, 2243, 2251, 2523, 2535, 2541, 2990
`\skip_set_eq:NN` 1935, 2028
`\skip_sub:Nn` 2512, 2519
`\skip_use:N` 2539, 2545
`\skip_zero:N` 2208
`\c_zero_skip` 2423, 2424, 2473, 2474, 2536, 2542

`\small` 3973, 4012
`\songti` 4723
`\space` 415, 2743, 2797, 3516, 3578, 3616, 3647, 3748
`space` 10, 33, 262, 2068

`\special` 1738, 1739, 1742, 1743, 1751, 1752,
 1755, 1756, 1867, 1881, 4815, 4816, 4817, 4818, 4819,
 4820, 4821, 4822, 4823, 4824, 4825, 4826, 4827, 4828,
 4829, 4830, 4831, 4832, 4839, 4840, 4841, 4842, 4843,
 4844, 4845, 4846, 4847, 4848, 4849, 4850, 4851, 4852,
 4853, 4854, 4855, 4856, 4888, 4889, 4890, 4891, 4892,
 4893, 4894, 4895, 4896, 4897, 4898, 4899, 4900, 4901,
 4902, 4903, 4904, 4905, 4906, 4907, 4908, 4909, 4910,
 4911, 4912, 4913, 4920, 4921, 4922, 4923, 4924, 4925,
 4926, 4927, 4928, 4929, 4930, 4931, 4932, 4933, 4934,
 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943,
 4944, 4945, 4952, 4953, 4954, 4955, 4956, 4957, 4958,
 4959, 4960, 4961, 4962, 4963, 4964, 4965, 4966, 4967,
 4968, 4969, 5001, 5002, 5003, 5004, 5005, 5006, 5007,
 5008, 5009, 5010, 5011, 5012, 5013, 5014, 5015, 5016,
 5017, 5018, 5019, 5020, 5021, 5022, 5023, 5024, 5025, 5026

`\SplitArgument` 1548, 1648, 2456

str commands:

`\c_backslash_str` 5430
`\c_colon_str` 1001, 1002, 1003
`\c_dollar_str` 4651
`\str_case:nnTF` 1822, 5401, 5416
`\str_case_e:nnTF` 1203, 1214
`\str_const:Nn` 43, 50
`\str_if_empty:NTF` 5487, 5523, 5532
`\str_if_eq:nnTF` 498, 546, 551, 1047, 1365,
 1856, 1864, 3812, 3839, 3853, 4137, 4140, 4157, 4162, 5407
`\str_if_eq_p:nn` 4104, 4105
`\str_lowercase:n` 472
`\str_new:N` 5436, 5437, 5438

`\str_set:Nn` 5454, 5455, 5456
`\strutbox` 4048
`sub3section` 9, 274
`sub4section` 9, 274
`subparagraph/afterskip` 35
`subparagraph/beforeskip` 35
`subparagraph/numbering` 16
`subparagraph/runin` 22
`subsection/afterskip` 35
`subsection/beforeskip` 35
`subsection/numbering` 16
`subsection/runin` 22
`\subsectionname` 3402, 3647
`subsubsection/afterskip` 35
`subsubsection/beforeskip` 35
`subsubsection/numbering` 16
`subsubsection/runin` 22
`\symmincho` 1708

sys commands:

`\c_sys_engine_str` 44
`\sys_if_engine_pdftex:TF` ... 220, 306, 376, 4076, 4082
`\sys_if_engine_pdftex_p:` 4735
`\sys_if_engine_uptex:TF` 4078, 4084
`\sys_if_engine_uptex_p:` 4736
`\sys_if_engine_xetex:TF` 300, 5035
`\sys_if_output_pdf:TF` 112, 482, 1859, 4083
`\sys_if_platform_windows:TF` 4088

T

`\tablename` 2321
`tablename` 12
`\TeX` 4052

TeX and L^AT_EX 2_ε commands:

`\@ifdefinable` 704
`\@ifitaliccorr` 1693
`\@@par` 2885, 3600, 3633, 3667, 3711, 3727
`\@afterheading` 2628, 2685, 2774, 2784, 2943
`\@afterindentfalse` 2563, 2728, 2732, 2837, 2843
`\@afterindenttrue` 2561, 2730, 2835, 2841
`\@Alph` 3744, 3749
`\@break@tfor` 423
`\@chapapp` 2743, 2797, 3327, 3350
`\@chapter` 2734, 2735
`\@chinese` 2272, 2281
`\@classoptionslist` 46, 418, 424, 426
`\@clubpenalty` 2958
`\@currentHref` 3113
`\@currentlabel` 3455, 3459
`\@currentxt` 4, 140, 5240, 5317, 5323
`\@currname` 4, 140, 5240, 5318, 5324
`\@dblarg` 2856
`\@defaultfamilyhook` 1790
`\@empty` 108, 2865, 2874
`\@endpart` 2664, 2701, 2703
`\@EverySelectfont@Init` 1964, 1978, 1990
`\@firstofone` 56

\@firstoftwo	3266	\@text@composite@x	53, 754
\@gobble	56	\@tfor	421
\@hangfrom	2884, 2922	\@topnewpage	2771, 2781
\@ifclassloaded	3757	\@topnum	2727
\@ifdefinable	704	\@ttfamilyhook	1789
\@ifl@t@r	5272	\@xsect	2912, 2933, 2934
\@ifpackagelater	37, 40, 634, 647, 650, 676, 1894	\@zhdig	2280
\@ifpackageloaded	18, 21, 643, 758, 1844, 3187, 3292, 3475, 5255, 5268	\@zhnum	2272, 2279
\@ifpackagewith	3132, 3141, 3192	\abstractname	12
\@ifstar	2854	\add@unicode@accent	54, 755
\@listI	3971, 4010	\addCJKfontfeature	71
\@listi	3971, 3978, 3989, 4010, 4017, 4028	\AfterEndPreamble	140
\@M	2610, 2652, 2672, 2693, 2817, 2885, 2923, 2951	\algorithmname	13
\@makechapterhead	2771, 2773, 2786	\appendixname	12, 25
\@makeschapterhead	2781, 2783, 2811	\AtBeginDocument	40, 49
\@minus	3601, 3634, 3668, 3684, 3698, 3967, 3969, 3975, 3977, 3979, 3980, 3986, 3988, 3990, 3991, 4006, 4008, 4014, 4016, 4018, 4019, 4025, 4027, 4029, 4030	\AtBeginDvi	47
\@namedef	3227, 3229, 3233, 3235, 3239, 3241, 3249, 3254	\AtBeginShipoutFirst	47
\@nameuse	3227, 3229, 3239, 3241, 3249	\AtEndOfClass	41
\@ne	420, 3185, 3314, 3345	\AtEndOfPackage	41
\@nil	955	\AtEndOfPackageFile*	139
\@nobreakfalse	2945	\AtEndPreamble	140
\@noskipsecfalse	2949	\baselineskip	11, 11, 81, 84
\@noskipsectrue	2946	\bfseries	51
\@onlypreamble	40, 444, 509, 1353, 1354, 1355, 1356, 1357, 1358, 1767, 1768, 1769, 1770, 4130	\bibname	12
\@part	2565, 2587, 2589	\c@chapter	3749
\@pkgextension	6, 7, 8, 9, 20, 23, 645, 5317, 5323	\c@secnumdepth	2591, 2613, 2632, 2655, 2736, 2795, 2862, 2892, 2906, 3311, 3314, 3318, 3325, 3326, 3334, 3342, 3345, 3350, 3353
\@plus	3545, 3546, 3601, 3602, 3634, 3635, 3668, 3669, 3684, 3698, 3712, 3728, 3967, 3968, 3969, 3975, 3976, 3977, 3979, 3980, 3986, 3987, 3988, 3990, 3991, 4006, 4007, 4008, 4014, 4015, 4016, 4018, 4019, 4025, 4026, 4027, 4029, 4030	\c@section	3744
\@popfilename	141, 5320	\caption	34
\@pushfilename	140	\catcode	48, 52, 131, 141
\@reset@ptions	140, 5314	\catcodetable	52
\@rmfamilyhook	75, 1787	\ccwd	11, 11, 27, 27, 34, 34, 40, 78, 79, 79, 83, 84, 84, 85
\@schapter	2734, 2776	\chapter	9, 17, 21, 26, 26, 26, 90, 98, 113
\@secCNTformat	2857, 2870	\chaptermark	13
\@secondoftwo	3267	\chaptername	17
\@secpenalty	2848, 3604, 3637, 3671, 3685, 3699	\char	54
\@sect	2856, 2861	\chardef@text@cmd	53, 752
\@setfontsize	3957	\CheckCommand	79
\@sffamilyhook	1788	\Chinese	34
\@spart	2565, 2587, 2666	\chinese	29, 34, 86
\@ssect	2855, 2913	\CJK@@ignorespaces	50, 563
\@startsection	98, 101, 109, 2831, 3039	\CJK@encoding	61, 63, 70, 861, 865, 949, 1054, 1108, 1396, 1399, 1401, 1405, 1545, 1555, 1556, 1574, 1575
\@svsec	2865, 2870, 2874, 2884, 2886, 2900, 2901	\CJK@envStart	49, 49, 50, 558
\@svsechd	2898, 2929, 2930, 2952	\CJK@family	57, 852, 866, 917, 949, 1161, 1607, 1617, 1619, 2016, 2017
\@tempdima	3275, 3277	\CJK@ignorespaces	50, 50, 567, 571
\@tempskipa	2834, 2836, 2837, 2848, 2878, 2879, 2916, 2917, 2935, 2936, 2939, 2954, 2955	\CJK@input	49, 511
\@tempswafalse	2581	\CJK@loadBinding	48, 49, 554
\@tempswatrue	2579	\CJK@loadEncoding	48
		\CJK@makeActive	49, 553
		\CJK@plane	49, 80, 461, 462, 472, 2005
		\CJK@surr	49, 523
		\CJK@upperReset	50
		\CJKfamily	63, 73

\CJKfamilydefault	49, 50, 67, 76, 76, 77, 77
\CJKglue	84, 118
\CJKhook	50, 50
\CJKrmdefault	48
\CJKsymbol	80
\CJKunderdotbasesep	34
\CJKunderline	34
\clearpage	50
\contentsname	12, 12
\CS	58
\CTEX@add	420, 423, 424
\CTEX@add@unicode@accent	723, 755
\CTEX@addloflotskip	2532, 2769
\CTEX@addtocline	2549, 2607, 2648, 2765, 2896, 2910
\CTEX@afterindent	2840, 3000, 3007
\CTEX@aftertitle	2926, 2931, 2999, 3006
\CTEX@alchar	692, 715
\CTEX@appendix@number	3054, 3066, 3072
\CTEX@appendix@numbering	3055, 3068, 3074
\CTEX@beginallalchar	692, 719, 725, 740, 764
\CTEX@chapter@afterindent	2729
\CTEX@chapter@aftername	2801
\CTEX@chapter@afterskip	2808, 2827
\CTEX@chapter@aftertitle	2804, 2823
\CTEX@chapter@beforeskip	2788, 2813
\CTEX@chapter@break	2725
\CTEX@chapter@fixskip	2789, 2809, 2814, 2828
\CTEX@chapter@format	2794, 2819
\CTEX@chapter@lofskip	2471
\CTEX@chapter@lotskip	2472
\CTEX@chapter@numbering	2740, 3074
\CTEX@chapter@pagestyle	2470, 2726
\CTEX@chapter@titleformat	2803, 2822
\CTEX@chapter@tocline	3011
\CTEX@chaptername	2801
\CTEX@chardef@text@cmd	702, 752
\CTEX@defaultfamilyhook	1800
\CTEX@document@left@hook	5287, 5293
\CTEX@document@right@hook	5289, 5295
\CTEX@endallalchar	692, 721, 728, 744, 765
\CTEX@Family@CMap	458, 486, 489
\CTEX@fixheadingskip	2510, 2515, 2558, 2626, 2683, 2707, 2809, 2828, 2851, 2941
\CTEX@fixskip	2851, 2941, 3001, 3008
\CTEX@fixtopskip	2507, 2585, 2789, 2814
\CTEX@getttitle	2606, 2647, 2670, 2691, 2764, 2779, 2877, 2915, 3119
\CTEX@hang	2970, 3002, 3009
\CTEX@heading@format@initial	2488, 2611, 2653, 2673, 2694, 2793, 2818, 2882, 2920, 3367, 3386, 3405
\CTEX@heading@glue	2956, 2987
\CTEX@headinghang	2616, 2658, 2675, 2696, 2800, 2820, 2963
\CTEX@headingskip	2524, 2559, 2586, 2627, 2684, 2708, 2790, 2810, 2815, 2829, 2852, 2942
\CTEX@hyperheadinghook	3087, 3117
\CTEX@ifnamefalse	2527, 2597, 2602, 2638, 2643, 2668, 2689, 2748, 2754, 2760, 2777, 2863, 2872, 2914
\CTEX@ifnametrue	2527, 2593, 2634, 2741, 2868
\CTEX@indentbox	2505, 2901, 2930, 3369, 3388, 3407
\CTEX@makeanchor	2598, 2603, 2639, 2644, 2749, 2873, 3077, 3085, 3097, 3102, 3108
\CTEX@makeanchor@chapter	2755, 2761
\CTEX@makeanchor@schapter	2778
\CTEX@makeanchor@sect	2864
\CTEX@makeanchor@spart	2669, 2690
\CTEX@makeanchor@ssect	2855
\CTEX@part@afterindent	2560
\CTEX@part@aftername	2617, 2659, 3371
\CTEX@part@afterskip	2625, 2682, 2706, 3377
\CTEX@part@aftertitle	2621, 2662, 2678, 2699, 3375
\CTEX@part@beforeskip	2557, 2584, 3366
\CTEX@part@break	2555, 2574
\CTEX@part@fixskip	2558, 2585, 2626, 2683, 2707
\CTEX@part@format	2612, 2654, 2674, 2695, 3368
\CTEX@part@indent	3369
\CTEX@part@numbering	2592, 2633, 3370
\CTEX@part@pagestyle	2469, 2576
\CTEX@part@titleformat	2619, 2661, 2677, 2698, 3375
\CTEX@part@tocline	3011
\CTEX@partname	2617, 2659, 3371
\CTEX@patch@text@cmd	730, 756
\CTEX@patch@tunec	750
\CTEX@popfilename@hook	5320
\CTEX@postappendix	3059, 3067, 3073
\CTEX@postchapter	3073
\CTEX@postsection	3067
\CTEX@preappendix	3058, 3065, 3071
\CTEX@prechapter	3071
\CTEX@presection	3065
\CTEX@reset@options@hook	5314
\CTEX@rmfamilyhook	75, 1775, 1797, 1806
\CTEX@runin	2880, 2918, 2937, 3003, 3010
\CTEX@save@appendix	3060, 3063
\CTEX@save@refstepcounter	3498, 3501
\CTEX@section@aftername	3390
\CTEX@section@afterskip	3396
\CTEX@section@aftertitle	3394
\CTEX@section@beforeskip	3385
\CTEX@section@format	3387
\CTEX@section@indent	3388
\CTEX@section@numbering	3068, 3389
\CTEX@section@titleformat	3394
\CTEX@sectionhang	2886, 2924, 2963
\CTEX@sectionname	3390
\CTEX@selectfont@hook	1958, 2005, 2008
\CTEX@selectfont@save	1969, 1984, 1988, 1991
\CTEX@setcurrentlabel@n	3453, 3502
\CTEX@setheadingskip	2522, 2557, 2584, 2625, 2682, 2706, 2788, 2808, 2813, 2827, 2850, 2940
\CTEX@sffamilyhook	1798, 1807
\CTEX@subsection@aftername	3409

\CTEX@subsection@afterskip	3415	\f@size	861, 867, 944, 952, 955, 956, 1914, 1955, 3278
\CTEX@subsection@aftertitle	3413	\fangsong	8
\CTEX@subsection@beforeskip	3404	\fbox	18
\CTEX@subsection@format	3406	\figurename	12
\CTEX@subsection@indent	3407	\font	55, 58
\CTEX@subsection@numbering	3408	\font@name	57, 58, 870, 877, 878, 883, 888, 906, 907, 924, 992, 993, 994, 995, 996, 1003
\CTEX@subsection@titleformat	3413	\fontcharwd	79
\CTEX@subsectionname	3409	\footnotesep	118, 118
\CTEX@text@composite@x	717, 754	\footnotesize	118
\CTEX@textcmd	737, 738, 742, 745	\get@external@font	900
\CTEX@thechapter	3072	\getanddefine@fonts	981
\CTEX@thesection	3066	\globaldefs	56
\CTEX@titleformat@n	98, 2925, 2931, 2998, 3005	\globaljfont	58
\CTEX@titlepslabel@clear	3236, 3246	\H@old@chapter	3091
\CTEX@titlepslabel@set	3230, 3242, 3246	\H@refstepcounter	3476
\CTEX@toc@width@n	3272, 3287, 3299	\hbox	22, 90
\CTEX@todayold	2288, 2293	\heiti	8, 130
\CTEX@ttfamilyhook	1799, 1808	\hrule	90
\CTEX@update@sectionformat@n	2839, 2996	\Hy@chapapp	2749, 2755, 2761, 2778
\CTEX@verbatim@font@hook	1687, 1691	\Hy@driver	1864
\ctex@zhmap@endinput	5029	\Hy@MakeCurrentHrefAuto	3110
\CTEXdigits	29	\Hy@org@chapter	3091
\CTEXifname	25, 26	\Hy@raisedlink	3111
\CTEXnumber	29, 29	\hyper@anchorend	3114
\CTEXnumberline	25	\hyper@anchorstart	3113
\CTEXoptions	34, 34	\if@mainmatter	2738, 2796, 3325, 3775, 3776
\ctexset	5, 5, 6, 8, 12, 15, 30, 31, 34, 34, 120	\if@nobreak	2845
\CTEXsetup	34	\if@noskipsec	2553, 2832, 2948
\CTEXthechapter	13, 14, 26, 102	\if@openright	2569, 2711, 2723, 3548, 3576
\CTEXthesection	13, 17, 105	\if@tempswa	2717
\CTEXunderdotbasesep	34	\if@twocolumn	2577, 2770, 2780
\CTEXunderline	34	\if@twoside	2710, 3309, 3332
\curr@fontshape	867, 901, 910, 914, 956	\iffalse	60
\DeclareAlternateKanjiFont	70	\ifHy@implicit	3090
\DeclareFontEncoding	60	\ifin@	60, 423, 970
\DeclareFontFamily	47, 58	\iftrue	60
\DeclareFontShape	58	\in@	422
\DeclareFontShape@	70, 1506	\indexname	12
\DeclareRobustCommand	142	\insertcontinuationtext	13
\DeclareTextCommand	54	\itdefault	58
\DeclareTextSymbol	53	\itshape	51
\DeclareUnicodeAccent	54	\jfam	74
\DeclareUnicodeComposite	53	\kaishu	8
\define@newfont	57, 893	\kanjifamilydefault	76
\do@subst@correction	57, 58, 892	\kanjiskip	81
\em	74	\labelformat	108, 108, 108
\endlinechar	48	\leftmark	13
\escapechar	60	\linewidth	84
\EverySelectfont	79	\lishu	8
\external@font	903, 904, 906	\listfigurename	12
\extract@font	57, 891	\listtablename	12
\f@baselineskip	2186	\ltj@@does@alt@set	59
\f@encoding	865, 909, 1011	\ltj@@getjfontnumber	59, 964
\f@family	57, 866, 909, 917	\ltj@@set@stackfont	1001, 1002, 1003
\f@series	861, 949	\ltj@allalchar	698
\f@shape	861, 949		

<code>\ltj@curjfont</code>	58, 80	<code>\ps@fancy</code>	3338
<code>\ltj@pickup@altfont@auxy</code>	56, 950	<code>\ps@headings</code>	3308
<code>\ltj@pickup@altfont@copy</code>	56, 56, 962	<code>\refname</code>	12, 13
<code>\ltj@setpar@global</code>	73, 1000, 1695, 1697	<code>\refstepcounter</code>	108
<code>\ltj@tempcntc</code>	59, 965	<code>\refstepcounter@noarg</code>	3478
<code>\ltjalchar</code>	54	<code>\refstepcounter@optarg</code>	3479
<code>\ltjlineendcomment</code>	52	<code>\relax</code>	49, 83
<code>\ltjsetkanjiskip</code>	73	<code>\rightmark</code>	13
<code>\ltjsetparameter</code>	73	<code>\rmfamily</code>	52, 75, 75, 76
<code>\ltjsetxkanjiskip</code>	73	<code>\roman</code>	29
<code>\m@ne</code>	2591, 2613, 2736, 2795, 3318, 3325, 3326, 3350, 4182, 4183, 4184, 4185, 4186, 4187	<code>\scantokens</code>	142
<code>\markboth</code>	13	<code>\scr@new@selectfont</code>	1972, 1987
<code>\markright</code>	13	<code>\scr@selectfont</code>	1984, 1988
<code>\math@bgroup</code>	1010	<code>\scriptfont@name</code>	995, 1002
<code>\math@egroup</code>	1015	<code>\section</code>	9, 15, 16, 17, 17, 22, 22, 23, 23, 24, 24, 26, 26, 35, 88, 98, 105
<code>\math@fonts</code>	998	<code>\section*</code>	16
<code>\mathbb</code>	118	<code>\sectionmark</code>	13
<code>\maxdimen</code>	11, 84	<code>\sectionname</code>	17
<code>\meaning</code>	142, 143, 143, 144	<code>\sectionpage</code>	15, 106
<code>\newcommand</code>	142	<code>\selectfont</code>	59, 75, 79, 80, 81, 83
<code>\newrobustcmd</code>	142	<code>\setCJKfamilyfont</code>	30, 30
<code>\newtitlemark</code>	104	<code>\setCJKmainfont</code>	30
<code>\nfss@catcodes</code>	62	<code>\setcounter</code>	107
<code>\noindent</code>	90	<code>\sf@size</code>	994
<code>\normalfont</code>	76	<code>\sffamily</code>	52
<code>\normalsize</code>	8, 114	<code>\size@update</code>	83, 84, 1999, 2000, 2146, 2160
<code>\nouppercase</code>	49	<code>\sldefault</code>	58
<code>\NR@gettitle</code>	3123	<code>\songti</code>	8
<code>\numberline</code>	104	<code>\special</code>	6, 47, 77
<code>\p@</code>	1914, 1955, 2656, 2767, 2768, 2787, 2797, 2807, 2812, 2826, 3278, 3544, 3572, 3573, 3574, 3575, 3582, 3967, 3968, 3969, 3975, 3976, 3977, 3979, 3980, 3986, 3987, 3988, 3990, 3991, 4006, 4007, 4008, 4014, 4015, 4016, 4018, 4019, 4025, 4026, 4027, 4029, 4030	<code>\split@name</code>	955
<code>\pagenumbering</code>	29	<code>\SplitArgument</code>	89
<code>\pagestyle</code>	10	<code>\ssf@size</code>	996
<code>\par@update</code>	80, 1975, 1979, 1983	<code>\strutbox</code>	118
<code>\paragraph</code>	9, 22, 39	<code>\subparagraph</code>	9, 22, 39
<code>\parindent</code>	11, 34, 83, 85, 90, 98, 118	<code>\subsectionname</code>	17
<code>\parskip</code>	90	<code>\subsectionpage</code>	15, 106
<code>\part</code>	15, 21, 23, 24, 90, 98	<code>\subst@correction</code>	920, 927
<code>\partmark</code>	91	<code>\tablename</code>	12
<code>\partname</code>	17	<code>\textfont@name</code>	993, 1001
<code>\partpage</code>	15, 106	<code>\tf@size</code>	992
<code>\PassOptionsToClass</code>	46	<code>\thechapter</code>	13, 26, 26
<code>\PassOptionsToPackage</code>	77	<code>\thesection</code>	13, 17, 26, 105
<code>\pdfmapline</code>	123	<code>\titleformat</code>	101
<code>\pickup@font</code>	57	<code>\titlespacing</code>	101
<code>\pickup@jfont</code>	897	<code>\today</code>	11, 11
<code>\pingfang</code>	8, 130	<code>\trans@languagepath</code>	3824, 3850
<code>\ProcessKeysOptions</code>	32, 40	<code>\ttfamily</code>	52
<code>\proofname</code>	13, 87	<code>\ttl@a</code>	3205, 3213
<code>\protected</code>	103	<code>\ttl@chapterout</code>	3200
<code>\protected@edef</code>	2285, 2287, 2870, 3455	<code>\ttl@extract</code>	101
<code>\ProvidesFile</code>	131	<code>\ttl@setifthe</code>	3259
		<code>\ttl@setsubmark</code>	103, 3232, 3238
		<code>\ttl@settopmark</code>	103, 3226, 3257
		<code>\ttl@tocpart</code>	3202, 3204
		<code>\ttlh@hang</code>	3166, 3176

<code>\ttlh@runin</code>	3165, 3175	<code>\tex_xkanjiskip:D</code>	2024, 2028, 2038
<code>\tunec.def</code>	54	<code>\textasteriskcentered</code>	756
<code>\tw@</code>	423	<code>\the</code>	4754, 4757, 4758, 4777, 4781
<code>\uppercase</code>	49	<code>\thechapter</code>	2743, 2746, 2797, 3327, 3350, 3579
<code>\use@mathgroup</code>	1006, 1037	<code>\theparagraph</code>	3681
<code>\usepackage</code>	33, 34, 35	<code>\thepart</code>	2595, 2614, 2636, 2656, 3518
<code>\verbatim@font</code>	1685, 1689	<code>\thesection</code>	3311, 3318, 3334, 3342, 3353, 3598
<code>\vskip</code>	22	<code>\thesubparagraph</code>	3695
<code>\wrong@fontshape</code>	57	<code>\thesubsection</code>	3314, 3345, 3630
<code>\x@protect</code>	5404	<code>\thesubsubsection</code>	3664
<code>\xdef</code>	103	<code>\thispagestyle</code>	2575, 2576, 2713, 2724, 2726
<code>\XeTeXglyphbounds</code>	135	<code>\tiny</code>	3996, 4035
<code>\xkanjiskip</code>	81	tl commands:	
<code>\yahei</code>	8, 130	<code>\c_novalue_tl</code>	5275, 5277, 5279
<code>\youyuan</code>	8	<code>\c_space_tl</code>	477
<code>\z@</code>	683, 686, 2610, 2672, 2727, 2792, 2817, 2836, 2879, 2917, 2936, 2950, 3185, 3311, 3334, 3342, 3353, 3968, 3976, 3987, 4007, 4015, 4026	<code>\tl_clear:N</code>	192, 358, 1639, 1643, 1889, 1901, 2017, 2137, 2462, 2480, 2491, 3922, 5488
<code>\zhdig</code>	86	<code>\tl_clear_new:N</code>	60, 335
<code>\zhdigits</code>	29	<code>\tl_const:Nn</code>	3, 388, 393, 398, 403, 623, 645, 1054, 1067, 1395, 1792, 1794, 1795, 1796, 1803, 1804, 1805, 2156, 2384, 2390, 3758, 3777, 3779, 4652, 4655, 4656
<code>\zhnum</code>	86	<code>\tl_gput_right:Nn</code>	432, 454, 1784, 1797, 1798, 1799, 1800, 1806, 1807, 1808, 5284, 5286, 5304
<code>\zhnumber</code>	29, 29	<code>\tl_gset:Nn</code>	221, 224, 227, 378, 530, 533, 992, 994, 996, 1369, 1809, 1820, 4089, 4092, 4093, 4109, 4142
<code>\zihao</code>	27	<code>\tl_gset_eq:NN</code>	1841
<code>\ziju</code>	27	<code>\tl_gset_rescan:Nnn</code>	1376
<code>\zw</code>	79	<code>\tl_head:N</code>	3926
tex commands:		<code>\tl_head:n</code>	1420
<code>\tex_afterassignment:D</code>	706	<code>\tl_if_blank:nTF</code>	726, 1454, 1463, 1663, 1664, 1679, 2109, 2119
<code>\tex_chardef:D</code>	707	<code>\tl_if_empty:NNTF</code>	374, 856, 2213, 2396, 3470, 4100
<code>\tex_def:D</code>	5480	<code>\tl_if_empty:nTF</code>	127, 1726, 1740, 1753, 5451
<code>\tex_dimexpr:D</code>	123, 2195	<code>\tl_if_eq:NNTF</code>	1816, 2149, 3837
<code>\tex_divide:D</code>	84, 2196, 2236	<code>\tl_if_exist:NNTF</code>	90, 1791, 1794, 1795, 1796, 1803, 1804, 1805, 2310, 2341, 2352, 2355, 3770, 3860, 5303
<code>\tex_edef:D</code>	5534	<code>\tl_if_in:NnTF</code>	5515
<code>\tex_endlinechar:D</code>	520	<code>\tl_if_novalue:nTF</code>	1285, 1306, 1657, 2461
<code>\tex_font:D</code>	476, 5061	<code>\tl_map_inline:Nn</code>	1569
<code>\tex_glueexpr:D</code>	2186	<code>\tl_map_inline:nn</code>	3923
<code>\tex_hangindent:D</code>	2983	<code>\tl_new:N</code>	128, 130, 443, 709, 858, 859, 947, 1051, 1102, 1129, 1146, 1147, 1168, 1511, 1645, 1782, 1834, 2031, 2157, 2170, 2312, 2382, 2383, 3058, 3059, 5297, 5298, 5303, 5491, 5492, 5493
<code>\tex_iftrue:D</code>	3776	<code>\tl_put_left:Nn</code>	1990, 5320
<code>\tex_ignorespaces:D</code>	1158, 1317, 1322, 2079, 2220, 3867, 5290	<code>\tl_put_right:Nn</code>	342, 998, 1652, 1978, 1979, 1983, 2412, 2451, 2453, 3926, 3927, 5314
<code>\tex_immediate:D</code>	473	<code>\tl_remove_all:Nn</code>	1421
<code>\tex_indent:D</code>	2503	<code>\tl_replace_all:Nnn</code>	1467
<code>\tex_interlinepenalty:D</code>	2492	<code>\tl_replace_once:Nnn</code>	1785
<code>\tex_kanjiskip:D</code>	1921, 1935, 1949	<code>\tl_rescan:nn</code>	144, 5533, 5542
<code>\tex_noindent:D</code>	2493, 2981	<code>\tl_set:Nn</code>	118, 187, 472, 504, 505, 506, 526, 705, 860, 903, 948, 1048, 1049, 1088, 1151, 1163, 1305, 1464, 1466, 1507, 1836, 1968, 2016, 2032, 2093, 2133, 2171, 2223, 2313, 2460, 2463, 2467, 3162, 3181, 3200, 3205, 3213, 3836, 4054
<code>\tex_numexpr:D</code>	727		
<code>\tex_pagegoal:D</code>	2511		
<code>\tex_parindent:D</code>	2497, 2502, 2973		
<code>\tex_parskip:D</code>	2519		
<code>\tex_pdffontattr:D</code>	476		
<code>\tex_pdflastobj:D</code>	477		
<code>\tex_pdfobj:D</code>	473		
<code>\tex_prevdepth:D</code>	2518		
<code>\tex_topskip:D</code>	2512		
<code>\tex_Uchar:D</code>	715, 726, 727		
<code>\tex_XeTeXcharglyph:D</code>	5065		
<code>\tex_XeTeXglyphbounds:D</code>	5089		

- `\tl_set_eq:NN` 865, 866, 917, 993, 995, 1152, 1177, 1607, 1619, 2158
`\tl_set_rescan:Nnn` 5489
`\tl_tail:N` 3167
`\tl_tail:n` 3177
`\tl_to_str:n` 126, 5432, 5461, 5502, 5503
`\tl_trim_spaces:n` 1466
`\tl_trim_spaces_apply:nN` 1636
`\tl_use:N` 434
`tocdepth` 25, 3422
`\today` 2288, 2293, 2296, 2301
`today` 11, 2288
token commands:
`\c_parameter_token` 104, 3290, 3292, 3295
`\token_to_meaning:N` 5446
`\token_to_str:N` 60, 152, 733, 975, 983, 992, 994, 996, 1205, 1206, 1207, 1209, 1216, 1217, 1218, 1280, 2338, 2339, 3875, 5068, 5245, 5264, 5350
`\toks` 4754, 4755, 4757, 4758, 4777, 4781
`\topsep` 3979, 3990, 4018, 4029
`\TrimSpaces` 2107
`\ttdefault` 1826
`\ttfamily` 1789
`\twocolumn` 2718
`\typeout` 2743, 2744, 3200
- U**
- `ubuntu` 35
um commands:
`\um_input_math_symbol_table:` 1030, 1034
`\um_sym:nnn` 1033
um internal commands:
`__um_input_math_symbol_table:` 1042
`__um_switchto_literal:` 1039
`__um_sym:nnn` 1041
`\undefined` 4779
`\UnicodeEncodingName` 733
`\unless` 2880, 2918, 2937
`\unskip` 2953
`\upshape` 1716, 1719
use commands:
`\use:N` 867, 909, 910, 956, 992, 994, 996, 1072, 1121, 1620, 2535, 2541, 2550, 2967, 3030, 5516, 5536, 5550
`\use:n` 131, 955, 1056, 1235, 1288, 1294, 1366, 1370, 1469, 1835, 2976, 3005, 3283, 3768, 3885, 4114, 5269, 5385, 5427, 5478, 5499
`\use:nn` 5544
`\use_i:nn` ... 115, 970, 2529, 5331, 5336, 5356, 5361, 5371
`\use_i:nnn` 242
`\use_ii:nn` 116, 970, 2527, 2531, 5331, 5336, 5356, 5361, 5372
`\use_ii:nnn` 132, 247
`\use_iii:nnn` 252
`\use_none:n` ... 188, 193, 198, 934, 1137, 1369, 1504, 2274
`\use_none:nn` 489
`\usebeamercolor` 3364, 3383, 3402, 3561, 3621, 3655
`\usebeamerfont` 3363, 3374, 3382, 3393, 3401, 3412, 3560, 3563, 3620, 3623, 3654, 3657
`\usefont` 1790
`\uselanguage` 3815, 3820, 3841, 3846
`\usepackage` 5245
`UTF8` 6, 35, 218
- V**
- `\verse` 4064
`\vfil` 2583, 2705
`\vskip` 2624, 2627, 2656, 2681, 2684, 2708, 2797, 2807, 2810, 2826, 2829, 2939, 2942, 3365, 3384, 3403, 3544, 3562, 3582, 3622, 3656
`\vspace` 2586, 2787, 2790, 2812, 2815
- W**
- `windows` 35
`windowsnew` 35
`windowsold` 35
`winfonts` 33
`\write` 4794
- X**
- `\x` 416, 422, 426, 4756, 4759, 4760, 4761, 4762, 4763, 4764, 4765, 4766, 4767, 4768, 4769, 4770, 4771, 4772, 4773, 4774, 4775, 4782, 4792, 4796
x commands:
`\x:` 421
`\xdef` 4795
`\xeCJKsetup` 629, 1897, 2074, 2077, 2080, 2095
- Y**
- `\yahei` 4723
`\youyuan` 4723
- Z**
- `\zhdig` 2280
`\zhdigits` 29, 2287
`zhmap` 6, 33, 237
`\zhnum` 2279
zhnum commands:
`\zhnum_counter:n` 2271
`\zhnumber` 29, 2285
`\zhnumsetup` 2297, 2302
`\zhtoday` 2296, 2301
`\zihao` 27, 3866, 3875
`zihao` 8, 167
`\ziju` 27, 2219
`\zw` 684, 1918, 1956, 2033