

CT_EX 宏集手册

ctex.org

2016/05/16 v2.4.2*

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、XeL^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 和中文文档类 ctexart、ctexrep、ctexbook 和 ctexbeamer 实现。

目录

第 1 节 介绍	2	第 8 节 实用命令	23
第 2 节 简明教程	3	8.1 字号与间距	23
2.1 CT _E X 宏集的组成	3	8.2 中文数字转换	24
2.2 CT _E X 宏集的安装和更新	3	8.3 杂项	24
2.3 使用 CT _E X 文档类	4	第 9 节 LuaL ^A T _E X 下的中文支持方式	24
2.4 使用 ctex 宏包	4	9.1 LuaL ^A T _E X 下替代字体的设置	25
第 3 节 宏包选项与 \ctexset 命令	5	第 10 节 CT _E X 宏集的配置文件	26
第 4 节 编译方式、编码与中文字库	5	10.1 修改宏包默认选项	26
4.1 编译方式	5	10.2 宏包载入后的配置	27
4.2 中文编码	6	10.3 配置标题中文翻译	27
4.3 中文字库	6	10.4 自定义字体集	27
第 5 节 排版格式设定	8	第 11 节 对旧版本的兼容性	27
5.1 文档默认字号	8	11.1 CT _E X 0.8a 及以前的版本	27
5.2 章节标题风格	9	11.2 CT _E X 0.9–CT _E X 1.0d	28
5.3 排版方案选项	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	29
第 6 节 文档汉化	11	11.4 CT _E X 2.2 之前的版本	30
6.1 日期汉化	11	第 12 节 宏集依赖情况与手工安装方法	30
6.2 文档标题汉化	11	第 13 节 开发人员	31
6.3 页面格式设置与汉化	12	第 14 节 代码实现	32
第 7 节 章节标题格式设置	14	版本历史	137
		代码索引	138

*ctex-kit rev. f12457f.

第 1 节 介绍

历史

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

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

2009 年 5 月，我们在 Google Code 建立了 `ctex-kit` 项目¹，对 `ctex` 宏包及相关脚本进行了整合，并加入了对 X_YT_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_EX3 的逐渐成熟，李清用 L^AT_EX3 重构了整个宏包的代码，并重新使用 `doc` 和 `DocStrip` 工具进行代码的管理，升级版本号为 2.0，并改称 CT_EX 宏集。

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 [GitHub](#)²。

最初，Knuth 在设计开发 T_EX 的时候没有考虑到多国语言支持，特别是对多字节的中日韩语言的支持。这使得 T_EX 以至后来的 L^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。由于中文和西文语言习惯的差异，用户很难使用标准文档类中的代码结构来表达中文标题。于是，用户不得不对标准文档类做较大的修改。除此之外，日期格式、首行缩进、中文字号和字距等细节问题，也需要精细的调校。我们设计 CT_EX 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 CT_EX 宏集的另一个主要目的就是尽可能消除这些差异带来的影响，使用户能够以一个统一的接口来使用不同的中文支持方式，使得同一份文档能够在不同环境下交换使用。

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

关于宏集名字の説明

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

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。

¹<http://code.google.com/p/ctex-kit/>

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

³比如：pdf_TE_X 引擎下的 CJK、zhmCJK 宏包，X_YT_EX 引擎下的 xeCJK 宏包和 Lua_TE_X 引擎下的 LuaTeX-j_a 宏包

第 2 节 简明教程

2.1 C_TE_X 宏集组成

为了适应用户不同的需求,我们将 C_TE_X 宏集的主要功能分散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 C_TE_X 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 <code>heading</code> 选项来开启
	ctexsize.sty	定义和调整中文字号,在 <code>ctex</code> 宏包或 C _T 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),在 <code>ctex</code> 宏包或 C _T E _X 中文文档类之外单独调用

2.2 C_TE_X 宏集的安装和更新

C_TE_X 宏集依赖的宏包和宏集已被最常见的 T_EX 发行版 T_EX Live 和 MiK_TE_X 所收录。如果本地安装 T_EX Live 或 MiK_TE_X 不是完全版本,就可能需要通过这两个发行版提供的宏包管理器来安装宏包。

T_EX Live 的宏包管理器是 `tlmgr`。用户可以在系统命令行中⁴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 2015 → TeX Live Manager 打开)。连接上远程仓库之后,搜索 `ctex` 安装即可。`tlmgr` 的图形界面使用 Perl 编写,容易造成系统假死。遇到这种问题的用户,也可以直接在系统命令行执行

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

MiK_TE_X 的宏包管理器是 `mpm` (MiK_TE_X Package Manager)。用户可以在开始菜单找到 MiK_TE_X → Maintenance (Admin) → Package Manager (Admin), 打开管理器,连接上远程仓库之后,搜索 `ctex` 安装即可。

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

当宏包发布新版本,并被发行版在远程仓库安装之后,在本地就可以通过宏包管理器来取得新版本。

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

⁴Windows 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。

⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

对于 MiKTeX, 在开始菜单找到 MiKTeX → Maintenance (Admin) → Update (Admin), 按照界面说明更新即可。

2.3 使用 CT_EX 文档类

如果用户需要在标准文档类的基础上添加中文支持和中文版式支持, 我们建议用户使用 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⁶。

例 1

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

你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译, 也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 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[UTF8]{ctexbeamer}
\begin{document}
\begin{frame}{中文演示文档}
\begin{itemize}
\item 你需要将所有源文件保存为 UTF-8 编码
\item 你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译
\item 也可以使用 (pdf)LaTeX 编译
\item 推荐使用 XeLaTeX 或 LuaLaTeX 编译
\end{itemize}
\end{frame}
\end{document}
```

2.4 使用 ctex 宏包

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

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

例 3

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

⁶使用 (pdf)L^AT_EX 时也能够使用 GBK 编码, 但不推荐。(见 4.2 节)

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。具体来说,这些选项里,有的以传统的方式提供,也有的以 $\langle key \rangle = \langle value \rangle$ 的形式提供。对于以键值对形式提供的选项,在下面的说明中使用**粗体**来表示 CT_EX 的默认设置。

另一方面,这些选项可以分为以下三类:

- 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 { \langle 键值列表 \rangle }

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 L^AT_EX3 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

CT_EX 宏集会根据用户使用的编译方式⁷,在底层选择不同的中文支持方式(见表 2)。

表 2 CT_EX 宏集的中文支持方式

编译方式	(pdf)L ^A T _E X	X _Y L ^A T _E X	LuaL ^A T _E X	upL ^A T _E X [*]
支持宏包	CJK	xeCJK	LuaTeX-jan	原生

^{*} pL^AT_EX-ng(或称 ApL^AT_EX)与 upL^AT_EX 兼容。使用 pL^AT_EX-ng 编译时,ctex 采用与 upL^AT_EX 相同的设置。

⁷ L^AT_EX、pdfL^AT_EX、X_YL^AT_EX、LuaL^AT_EX 及 upL^AT_EX。

不同的编译方式和中文支持方式会在一定程度上影响 $\text{CT}_{\text{E}}\text{X}$ 宏集的行为, 比如宏包对编码的处理。在用户使用 $\text{X}_{\text{L}}\text{A}_{\text{T}}_{\text{E}}\text{X}$ 、 $\text{Lua}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 及 $\text{up}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 编译时, $\text{CT}_{\text{E}}\text{X}$ 宏集使用 (且仅能使用) UTF-8 编码; 而因为历史原因, 在用户使用 $\text{L}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 及 $\text{pdf}_{\text{L}}\text{_{T}}_{\text{E}}\text{X}$ 编译时, 宏包默认使用 GBK 编码。用户需要保证编译方式、源文件编码、宏包编码选项三者一致。关于宏包编码选项, 可以参考 4.2 节。

除去文档编码之外, 选择不同的编译方式还可能影响 $\text{CT}_{\text{E}}\text{X}$ 宏集对字体选择、空格处理、标点处理的处理。具体的影响将在本文档后续内容中进行阐述。

4.2 中文编码

GBK ☆
UTF8 ☆

指明编写文档时使用的编码格式。 $\text{CT}_{\text{E}}\text{X}$ 宏集无法检测用户编写文档时使用的编码格式, 因此需要用户自行指定编码。我们建议用户总是使用 UTF-8 编码, 并显式指定 UTF8 选项, 并使用 $\text{X}_{\text{L}}\text{A}_{\text{T}}_{\text{E}}\text{X}$ 、 $\text{Lua}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 或 $\text{up}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 编译。

使用 $\text{X}_{\text{L}}\text{A}_{\text{T}}_{\text{E}}\text{X}$ 、 $\text{Lua}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 或 $\text{up}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 编译时, $\text{CT}_{\text{E}}\text{X}$ 宏集强制使用 UTF-8 编码, 此时 GBK 选项无效。使用 $(\text{pdf})\text{L}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 编译时, $\text{CT}_{\text{E}}\text{X}$ 宏集默认启用 GBK 选项; 不过, 用户也可以显式声明 UTF8 选项, 使 $\text{CT}_{\text{E}}\text{X}$ 宏集工作在 UTF-8 编码下。

4.3 中文字库

以往, 为 $\text{L}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 文档配置中文支持是一件相当繁琐的事情。默认情况下, $\text{CT}_{\text{E}}\text{X}$ 宏集能自动检测用户使用的编译方式 (参见 4.1 节) 和操作系统⁸, 选择合适的底层支持和字库, 从而简化配置过程。自动配置的情况参见表 3。

表 3 $\text{CT}_{\text{E}}\text{X}$ 宏集自动配置字体策略

	Mac OS X	Windows New ¹	Windows Old ²	其他
$\text{X}_{\text{L}}\text{A}_{\text{T}}_{\text{E}}\text{X}$	xeCJK 华文字库	xeCJK 中易字库 + 微软雅黑	xeCJK 中易字库	xeCJK Fandol 字库 ³
$\text{Lua}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ ⁴	LuaTeX-ja 华文字库	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja 中易字库	LuaTeX-ja Fandol 字库
$\text{pdf}_{\text{L}}\text{_{T}}_{\text{E}}\text{X}$	不可用 ⁵	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	不可用 ⁵
$\text{L}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ + DVIPDFMx	不可用 ⁶	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	CJK + zhmetrics Fandol 字库
$\text{up}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ + DVIPDFMx	不可用 ⁶	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex 中易字库	zhmetrics-uptex Fandol 字库

¹ Windows Vista 及以后的 Windows 操作系统。

² Windows XP 及以前的 Windows 操作系统。

³ 由马起园、苏杰、黄晨成等人开发的开源中文字体, 参见: <https://github.com/clerkma/fandol-fonts>。

⁴ $\text{Lua}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 编译时使用 LuaTeX-ja 宏包。对此, 第 9 节有特别说明。

⁵ 受 $\text{pdf}_{\text{L}}\text{_{T}}_{\text{E}}\text{X}$ 的限制, 无法嵌入 OpenType 字体。

⁶ 目前受 DVIPDFMx 的限制, Mac OS X 系统上的黑体和仿宋无法读取。

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

⁸ $\text{CT}_{\text{E}}\text{X}$ 宏集现在能够识别 Mac OS X 系统以及 Windows 系统。

zhmap	☆	zhmap = <code><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 / \LaTeX 编译时有意义。

true	使用 zhmetrics 宏包, 将 CJK 字库通过 \special 命令映射到 .ttf 文件。
false	使用传统的 CJK 字库 (Type 1) ⁹ 。
zhmCJK	载入 zhmCJK 宏包 ¹⁰ , 由 zhmCJK 宏包提供从 CJK 字库到 .ttf 的映射。

fontset	fontset = <code><none adobe fandol founder mac ubuntu windows windowsnew windowsold ...></code>
---------	---

New: 2014-03-08

指定 C \TeX 宏集加载的字库。

如果没有指定 fontset 的值, C \TeX 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)。

C \TeX 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 pdf \LaTeX 。
fandol	使用 Fandol 中文字体, 不支持 pdf \LaTeX 。
founder	使用方正公司的中文字体。
mac	使用 Mac OS X 系统下的华文字体, 不支持 (pdf) \LaTeX 。
ubuntu	使用 Ubuntu 系统下的文泉驿和文鼎字体。
windows	使用简体中文 Windows 系统下的中文字体, 自动判断 Windows 系统版本, 采用 windowsnew 或 windowsold 的设置。
windowsnew	使用简体中文 Windows Vista 或之后系统下的中易字体和微软雅黑字体。
windowsold	使用简体中文 Windows XP 或之前系统下的中易字体。

注意: 使用 (pdf) \LaTeX 编译的时候, 若设置 zhmap = false (比如需要使用 \LaTeX + Dvips 编译), 则需要按照传统方式¹¹在本地安装好 CJK 字体。

如果不想使用 C \TeX 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。C \TeX 宏集将载入名为 ctex-fontset-<name>.def 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录树下合适位置建立一个名为 ctex-fontset-<name>.def 的文件, 在这个文件里面自定义中文字体。然后通过使用 fontset=<name> 选项来调用它。字体配置文件的具体写法可以参考 C \TeX 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。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 \TeX 宏集预定义的中文字库还定义了一些字体命令。

\songti	宋体, CJK 等价命令 \CJKfamily{zhsong}。
---------	----------------------------------

⁹如果需要使用自定义的字体映射文件, 或者希望使用 Type1 字库, 请禁用本选项。

¹⁰zhmCJK 宏包基于 zhmetrics 和 CJK 宏包, 提供与 xeCJK 宏包类似的用户接口。

¹¹可以使用 zhmetrics 宏包提供的脚本 C \TeX Fonts.lua。

`\heiti` 黑体,CJK 等价命令 `\CJKfamily{zhhei}`。
`\fangsong` 仿宋,CJK 等价命令 `\CJKfamily{zhfs}`。
`\kaishu` 楷书,CJK 等价命令 `\CJKfamily{zhkai}`。

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

`\lishu` 隶书,CJK 等价命令 `\CJKfamily{zhli}`。
`\youyuan` 圆体,CJK 等价命令 `\CJKfamily{zhyou}`。

在 windowsnew 字库中,还有 `\yahei`。

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

第 5 节 排版格式设定

5.1 文档默认字号

`zihao` ☆
New: 2015-05-06

`zihao = <-4|5|false>`

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

`scheme = chinese` 时,对标准文档类默认值为 5,即设置 `\normalsize` 为五号字;对 `beamer` 则为 `false`,使用文档类原有的设置。

表 4 标准字体命令与字号的对应

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

10pt ☆
11pt ☆
12pt ☆

C_T_E_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_T_E_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_T_E_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。

标准文档类的其他选项在 C_T_E_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。C_T_E_X 会将这些选项传给标准文档类¹²。

¹²事实上,L^AT_EX 在文档类中的选项是全局设定的,除了对使用的文档类有影响外,也可能影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中,那么该选项将会被自动激活。

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。

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

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

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。

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

具体格式可参考 7 节中的 `runin` 和 `afterskip` 选项。

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

5.3 排版方案选项

scheme ☆ scheme = (chinese|plain)

New: 2015-04-11

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

`chinese` 对 `beamer` 以外的文档类, 调整默认字号为五号字, 并调 ‘ 整行距为 1.3; 汉化文档中的标题名字 (如 “图”、“表”、“目录” 和 “参考文献” 等, 见 6.2 节); 在 `heading = true` 的情况下¹³ (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 CT_EX 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¹⁴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;

¹³使用 CT_EX 文档类, 或者使用 `ctex` 宏包且开启该选项时。

¹⁴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。

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。

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¹⁵。
 false 总是忽略掉汉字后面的空格,不论汉字后是什么(使用 (pdf)LaTeX 编译时);等同于 auto 的效果(使用 XeLaTeX 编译时)。不建议使用该选项。
 auto 根据空格后面的情况决定是否保留:如果空格后面是汉字,则忽略该空格,否则保留。

例如,使用

例 6

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

将得到“汉字分词技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

使用 LuaLaTeX 及 upLaTeX 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。

true 等价于设置 `autoindent = 2`。

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。

¹⁵LaTeX 将单个换行视作一个空格。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化,使之以中文的方式显示今天的日期。如编译本文档的日期就是“2016 年 5 月 16 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“2016 年 5 月 16 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年五月十六日”。使用全汉字的日期格式。
`old` 效果为“May 16, 2016”。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

这里主要介绍由宏包 `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”。

<hr/> <code>tablename</code> ★ <hr/>	<code>tablename = <名字></code> 设置表格环境标题名 <code>\tablename</code> 。中文默认为“表”。
<hr/> <code>abstractname</code> ★ <hr/>	<code>abstractname = <名字></code> 设置摘要 <code>abstract</code> 环境标题名 <code>\abstractname</code> 。中文默认为“摘要”。注意 <code>book</code> 类没有摘要,该选项无效。
<hr/> <code>indexname</code> ★ <hr/>	<code>indexname = <名字></code> 设置索引标题名 <code>\indexname</code> 。中文默认为“索引”。
<hr/> <code>appendixname</code> ★ <hr/>	<code>appendixname = <名字></code> 设置附录标题名 <code>\appendixname</code> 。中文默认为“附录”。
<hr/> <code>bibname</code> ★ <hr/>	<code>bibname = <名字></code> 设置参考文献标题名 <code>\refname</code> (对 <code>article</code>)或 <code>\bibname</code> (对 <code>report</code> 、 <code>book</code> 和 <code>beamer</code>)。中文默认为“参考文献”。
<hr/> <code>proofname</code> ★ <hr/>	<code>proofname = <名字></code> 设置证明环境的名称 <code>\proofname</code> 。中文默认为“证明”。 如果使用 <code>ctexbeamer</code> 文档类或者在 <code>beamer</code> 文档类下使用 <code>ctex</code> 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。
<hr/> <code>refname</code> ★ <hr/>	<code>refname = <名字></code> 设置参考文献标题名 <code>\refname</code> 。中文默认为“参考文献”。
<hr/> <code>algorithmname</code> ★ <hr/>	<code>algorithmname = <名字></code> 设置算法环境标题名 <code>\algorithmname</code> 。中文默认为“算法”。
<hr/> <code>continuation</code> ★ <hr/>	<code>continuation = <名字></code> 设置 <code>beamer</code> 可断页的帧在续页标题中的延续标识 <code>\insertcontinuationtext</code> 。中文默认为“(续)”。

6.3 页面格式设置与汉化

当使用了 C_TE_X 的文档类或是用 `ctex` 宏包加载了 `heading` 选项时,会设置整个文档的页面格式(`page style`)为 `headings`,即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

同时,C_TE_X 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。

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

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

例 10

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

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

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

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

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

例 11

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

\begin{document}

\section{天地玄黄}
```

```

\newpage

\section{宇宙洪荒}

\end{document}

```

本例的页眉效果大致如下(有页眉线):

第 1 节——天地玄黄

— 1 —

第 7 节 章节标题格式设置

CT_EX 宏包对 L^AT_EX 的标准文档类(`article`、`report` 和 `book`)和 `beamer` 进行了扩充。当以 `heading` 选项调用 CT_EX 宏包时(5.2 节),则会启用章节标题的格式设置功能。本节就来介绍有关章节标题的格式选项,所有选项使用 `\ctexset` 命令设置。

第 6.3 节和本节介绍的功能已经被提取到 `ctexheading` 宏包之中,可以在 `ctex` 宏包和 `ctexart` 等文档类之外独立使用。各选项的默认值与 `scheme = plain` 时的情形相同。

章节标题的格式选项是分层设置的。顶层的选项是章节标题名称,次一级的选项是章节标题的格式。章节标题名包括 `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`;而可用的格式包括 `numbering`, `name`, `number`, `format`, `nameformat`, `numberformat`, `aftername`, `titleformat`, `aftertitle`, `runin`, `afterindent`, `beforeskip`, `afterskip`, `indent`, `pagestyle` 等。

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` 产生的标题格式,此时只有 `part`、`section` 和 `subsection` 这三层级别,并且 `runin` 和 `afterindent` 这两个格式无效。

多级选项之间用斜线分开,例如, `part/name` 选项设置 `\part` 标题的在数字前后的名称,而 `section/number` 选项设置 `\section` 标题的数字类型。

使用 `\ctexset` 设置多级选项时,可以在同一个上级选项下设置多个下级选项。

例如,同时设置 `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},
  }
}

```


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

numbering = true|false

控制是否对章节标题编号。对各级标题的默认值均为 true。

我们知道， \LaTeX 带星号的章节标题命令（如 \section* ）不会对标题编号，但也不会将该没编号的标题编入目录中。本选项控制的是不带星号的标题命令是否编号。设置本选项为 false，除了不对标题编号以外，功能与正常标题一致，比如可以编目录，正确的 hyperref 目录超链接位置和页眉标记。

注意，章节标题的编号深度受 \LaTeX 计数器 secnumdepth 的控制。numbering 选项在 secnumdepth 的控制下起作用。

如果没有特别说明，以下将用“...”代表各级章节标题名。

.../name	★
Updated: 2014-03-08	

name = {⟨前名字⟩,⟨后名字⟩}

name = {⟨前名字⟩}

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

例 13

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

会使得 \chapter 标题使用形如“第一章”的名字，而 \section 标题则使用形如“§1”的名字。

表 5 name 选项的默认设置

标题名	scheme = chinese	scheme = plain	注
part	{第,部分}	{\partname\space}	原 \partname 为 Part
chapter	{第,章}	{\chaptername\space}	原 \chaptername 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 \sectionname 为 \translate{Section}
section	同右	{}	
subsection (beamer)	{}	{\subsectionname\space}	原 \subsectionname 为 \translate{Subsection}
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 $\text{\chinese{chapter}}$ 之类。

例 14

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

number 选项的定义同时将控制对章节计数器的交叉引用。在引用计数器时，记录在 \LaTeX 辅助文件中的是 number 选项的定义。

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

表 6 `number` 选项的默认设置

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

```

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

```

以 `\CTEXthe` 开头的这组宏给出结合了 `name` 与 `number` 选项的章节编号输出格式。例如在 `scheme = chinese` 时，默认章编号输出格式就是 `\CTEXthechapter`，形如“第一章”。

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

```

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

```

Updated: 2015-06-30

`format` 选项用于控制章节标题的全局格式，作用域为章节名字和随后的标题内容。可以用于控制章节标题的对齐方式、整体字体字号等格式。

带加号的 `format+` 选项则用于在已有格式之后追加新的格式命令。

例如，设置章格式为无衬线字体左对齐，为节格式增加无衬线字体设置：

例 15

```

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

```

```

.../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 `titleformat` 选项。

`nameformat+` 用于在已有的章节名字格式后附加内容。

`nameformat` 选项的最后一个格式命令可以是一个有一个参数的命令。这个命令的参数用于接受章节名字和编号，实现特殊效果(见例 19)。

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

表 7 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

表 8 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,当你需要编号的格式和前后的章节名字不一样时可以使用。

`numberformat+` 选项用于在原有编号格式后面附加格式命令。

`numberformat` 选项的最后一个格式命令可以是一个有一个参数的命令。这个命令的参数用于接受编号数字。

例如,我们可以使用 `numberformat` 特别强调章标题中的数字:

例 16

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

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

```
.../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+` 选项用于在原有代码后面附加代码。

表 9 `aftername` 选项的默认设置

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

```
.../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+` 选项用于在原有标题格式后面附加格式命令。

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

例 17

```
\usepackage{varwidth} %% 提供 varwidth 环境
\ctexset{
```

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

上面的代码可以做出类似这样的章标题效果：

第三十四回

情中情因情感妹妹
错里错以错劝哥哥

表 10 titleformat 选项的默认设置

标题名	scheme = chinese	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+ *
New: 2015-06-19
```

aftertitle = {<代码>}
aftertitle+= {<代码>}
aftertitle 选项的参数 <代码> 将被插入到章节标题内容之后。
aftertitle+ 选项用于在原有代码后面附加代码。
aftertitle 选项的默认值, 在 scheme 选项的不同取值下相同。
sub3section 或 sub4section 宏包选项(见 5.2 节)会影响 aftertitle 选项的默认值。

表 11 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

表 12 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

```
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。
```

```
paragraph/runin  ★
```

```
subparagraph/runin ★
```

New: 2015-06-27

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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。

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

New: 2015-06-27

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。

表 13 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

```
.../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。

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

表 14 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

```
part/fixbeforeskip ★ fixbeforeskip = true|false
```

```
chapter/fixbeforeskip ★
```

New: 2016-05-10

book 和 report 类的 \part 和 \chapter 标题与正文顶部的距离除了由 beforeskip 选项设置的垂直间距外, 还会有一个空行。fixbeforeskip 选项用于抑制这个多余空行。

.../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。

afterskip 选项的默认值，在 scheme 选项的不同取值下相同。

sub3section 或 sub4section 宏包选项（见 5.2 节）会影响 aftertitle 选项的默认值。

表 15 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

表 16 indent 选项的默认设置

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

.../indent ★ indent = {⟨缩进间距⟩}

Updated: 2015-06-27

indent 选项用于设置章节标题本身的首行缩进。

indent 选项的默认值，在 scheme 选项的不同取值下相同。

例如，设置 \section 标题缩进 20pt：

例 18

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

part/pagestyle ★ 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）。

appendix/numbering ★ numbering = true|false

New: 2015-06-21

控制是否对附录章（对 book 与 report）或附录节（对 article）进行编号。

用法与普通章节 numbering 选项类似。

表 17 pagestyle 选项的默认设置

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

 appendix/name

 Updated: 2014-03-08

name = {<前名字>,<后名字>}

name = {<前名字>}

设置附录章(对 **book** 与 **report**)或附录节(对 **article**)的名字。

用法与普通章节 **name** 选项类似。

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

表 18 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	<code>\section</code>	{}		{}	
book, report	<code>\chapter</code>	<code>\appendixname\space</code>	附录 _□	<code>\appendixname\space</code>	Appendix _□

 appendix/number ★

number = {<数字输出命令>}

设置附录章(对 **book** 与 **report**)或附录节(对 **article**)编号的数字输出格式。

用法与普通章节的 **number** 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

与普通章节的 **number** 选项类似, 同样需要注意, 该选项不会影响计数器本身的输出, 即不影响 `\thesection` 或 `\thechapter` 的定义。

表 19 appendix/number 选项的默认设置

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

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 19

```

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

```

```
.....
\chapter{熟悉 \LaTeX}
```

本例的设置效果大致如下：

第
1
章

熟悉 L^AT_EX

第 8 节 实用命令

8.1 字号与间距

```
\zihao
Updated: 2014-03-08
```

`\zihao {<字号>}`
用于调整字号大小。其中 <字号> 的有效值共有 16 个，如表 20 所示。使用 `\zihao` 命令调整字体大小时，西文字号大小会始终和中文字号保持一致。

表 20 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

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

$\backslash\text{chinese}$	$\backslash\text{chinese}$ $\{(\text{counter})\}$
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 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 计数器上,将计数器的值以中文数字的形式输出。
$\backslash\text{zhnumber}$	$\backslash\text{zhnumber}$ $\{(\text{number})\}$
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。
$\backslash\text{zhdigits}$	$\backslash\text{zhdigits}$ $\{(\text{number})\}$
New: 2014-03-08	将阿拉伯数字转换为中文数字串。
$\backslash\text{CTEXnumber}$	$\backslash\text{CTEXnumber}$ $\backslash(\text{macro})$ $\{(\text{number})\}$ $\backslash(\text{macro})$ 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 $\backslash(\text{macro})$ 里。对 $\backslash(\text{macro})$ 的定义是局部的,将它展开一次就可以得到转换结果。 一般来说,并不需要使用 $\backslash\text{CTEXnumber}$,直接使用 $\backslash\text{zhnumber}$ 即可。但是,如果在文档中需要多次使用同一个数字 $\langle\text{number}\rangle$ 的中文形式,就可以先用 $\backslash\text{CTEXnumber}$ 将结果保存起来备用,而不是每次使用时都用 $\backslash\text{zhnumber}$ 现场转换一次。
$\backslash\text{CTEXdigits}$	$\backslash\text{CTEXdigits}$ $\backslash(\text{macro})$ $\{(\text{number})\}$ $\backslash\text{CTEXdigits}$ 与 $\backslash\text{CTEXnumber}$ 类似,但其转换的结果是中文数字串,而不是中文数字。

8.3 杂项

$\backslash\text{CTeX}$	用于显示 $\text{CT}\text{E}\text{X}$ 标志。
-------------------------	--------------------------------------

第 9 节 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 下的中文支持方式

在 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 下, $\text{CT}\text{E}\text{X}$ 宏集依赖 $\text{Lua}\text{Te}\text{X}\text{-ja}$ 宏包来完成中文支持。该宏包是日本 TEX 社区的北川弘典、前田一贵、八登崇之等人开发的,设计目的主要是在 $\text{Lua}\text{T}\text{E}\text{X}$ 引擎下实现日本

*p*TeX 引擎的(大部分)功能。它为了兼容 *p*TeX 的使用习惯,对 *La*TeX_{2_ε}的 NFSS 作了不少修改和扩充。这对于简体中文用户来说不是必要的,因而 CTeX 禁用了它在 *La*TeX 格式下的大部分设置,只保留了必要的部分。同时修改了它的字体设置方式,使得相关命令与 *xeCJK* 宏包大致相同。

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

9.1 *Lua*TeX 下替代字体的设置

AlternateFont <hr/> <small>New: 2014-04-14</small>	<pre>\setCJKfamilyfont {<family>} [AlternateFont = { {<character range₁>} [<alternate font features₁>] {<alternate font name₁>} {<character range₂>} [<alternate font features₂>] {<alternate font name₂>} } , <base font features>] {<base font name>}</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/> <small>New: 2014-04-14</small>	<pre>\setCJKfamilyfont {<family>} [CharRange = {<character range>} , <alternate font features>] {<alternate font name>}</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 20

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

declarecharrange ★ <hr/> <small>New: 2014-04-14</small>	<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 features>] {<alternate font name>}
```

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

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

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 ~/.texlive2015/texmf-var/tex/latex/ctex/ 目录下,作为本地/用户专有的配置文件。复制配置文件后需要运行 texhash 命令使本地配置文件生效。

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 21

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


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

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

10.2 宏包载入后的配置

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

例 22

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

例 23

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

4.3 节介绍的用于 `fontset` 选项的自定义字库文件, 类似于 C_TE_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 C_TE_X 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

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。

<u>cs4size</u> <u>c5size</u>	分别相当于 zihao=-4 和 zihao=5, 过时选项。
<u>CCT</u> <u>CCTfont</u>	相关选项已删除。
<u>indent</u> <u>noindent</u>	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。
<u>zhmap</u> <u>nozhmap</u>	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。
<u>winfonts</u> <u>adobefonts</u> <u>nofonts</u>	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)。
<u>punct</u> <u>nopunct</u>	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。
<u>cap</u> <u>nocap</u>	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。
<u>space</u> <u>nospace</u>	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。
<u>fancyhdr</u>	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。

<hr/> <hr/> hyperref	<p>新版本宏包中总是自动处理对 <code>hyperref</code> 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 <code>hyperref</code> 宏包。</p> <p><code>hyperref</code> 选项过时, 因兼容性保留, 功能是在导言区末尾载入 <code>hyperref</code> 宏包。</p>
<hr/> <hr/> fntef	<p>旧版本的 <code>fntef</code> 选项用于统一 <code>CCTfntef</code> 与 <code>CJKfntef</code> 的界面, 新版本 <code>CT_EX</code> 宏集不再支持 <code>CCT</code>, 而是直接载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包并禁用其彩色设置。该选项是过时选项。</p>
<hr/> <hr/> \CTEXunderdot \CTEXunderline \CTEXunderdblline \CTEXunderwave \CTEXsout \CTEXxout \CTEXfilltwosides	<p>在调用 <code>fntef</code> 宏包选项的同时, 旧版本 <code>CT_EX</code> 宏包由于需要支持 <code>CCT</code> 系统, 会将以 <code>\CJK</code> 开头的 <code>\CJKunderline</code> 等宏换名为以 <code>\CTEX</code> 开头的 <code>\CTEXunderline</code> 等宏。此功能在新版本的 <code>CT_EX</code> 宏集中已失去意义。</p> <p>此外, 在 <code>pdf_EX</code> 引擎下, 用于设置格式的 <code>\CJKunderdotbasesep</code> 等宏也被换名为 <code>\CTEXunderdotbasesep</code> 等宏。</p> <p>在新版本中, 上述由 <code>fntef</code> 衍生的相关宏都成为过时命令。</p>
<hr/> <hr/> \CTEXsetfont	更新当前的中文字体信息, 包括当前字距 (<code>\ccwd</code>) 和段首缩进 (<code>\parindent</code>)。一般来说, 用户无需使用这个命令。
<hr/> <hr/> \CTEXindent	更新 <code>\ccwd</code> 宽度后设置 <code>\parindent=2\ccwd</code> 。过时命令。
<hr/> <hr/> \CTEXnoindent	设置 <code>\parindent=0pt</code> 。过时命令。
<hr/> <hr/> \CTEXsetup	<p><code>\CTEXsetup[⟨选项⟩]{⟨标题⟩}</code></p> <p>相当于设置了 <code>\ctexset{⟨标题⟩ = {⟨选项⟩}}</code>。过时命令。</p>
<hr/> <hr/> \CTEXoptions	<p><code>\CTEXoptions[⟨选项⟩]</code></p> <p>相当于设置了 <code>\ctexset{⟨选项⟩}</code>。过时命令。</p>
<hr/> <hr/> \Chinese	<p><code>\Chinese{⟨counter⟩}</code></p> <p>新版宏集中 <code>\chinese</code> 统一了旧版本中 <code>\chinese</code> 和 <code>\Chinese</code> 的功能。因此, 该命令已过时。</p>
<hr/> <hr/> captiondelimter	<p>原为 <code>\CTEXoptions</code> 命令的选项, 用于控制 <code>\caption</code> 编号后面的标点。此选项已过时, 并在新版本的 <code>CT_EX</code> 宏包中失效。</p> <p>可以使用 <code>caption</code> 宏包的 <code>labelsep</code> 选项来完成同样的功能。</p>

例 24

```
% 代替 \CTEXoptions[captiondelimter={:}]
\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。

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

第 12 节 宏集依赖情况与手工安装方法

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

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

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 宏集, 它们还依赖 – etex 宏包。
- indentfirst 宏包, 属于 tools 宏集。
- everyisel 宏包, 属于 ms 宏集。
- zhnumber 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- CJK 宏集, 它的下划线功能依赖 ulem 宏包。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包。
- ➡ 以上是使用 pdfL_AT_EX 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项, 其中 zhm-CJK 是可选的。
- xeCJK 宏包, 它还依赖
 - fontspec 宏包, 它还依赖
 - * euenc 宏包。
 - * xunicode 宏包, 它还依赖 tipa 宏包。
- ➡ 以上是使用 X_LL_AT_EX 编译时的依赖项。
- LuaT_EX-j_a 宏集, 它还依赖
 - oberdiek 宏集。
 - xkeyval 宏包。
 - lualibs 宏包。
 - luaotfload 宏包, 它还依赖 luatexbase 宏包。

- ➡ 以上是使用 Lua \LaTeX 编译时的依赖项。
 - zhmetrics-uptex 宏包。
- ➡ 以上是使用 up \LaTeX 编译时的依赖项。

出于一些原因, zhmCJK 尚未被收入 \TeX Live 和 MiK \TeX 。因此, 若你希望使用 zhmCJK 作为 \CTEX 宏集的底层中文支持方式, 那么你需要自行安装该宏包。zhmCJK 的安装较为复杂。我们建议你

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

\CTEX 宏集已被 \TeX Live 和 MiK \TeX 收录, 若无特别理由, 我们强烈建议用户使用宏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 \CTEX 宏集的 **TDS 安装包**,
2. 按目录结构将文件复制到 \TeX 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 \TeX 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)

目前比较活跃的开发维护人员是刘海洋、李清和黄晨成。

参考文献

- [1] DONALD ERVIN KNUTH. *The \TeX book, Computers & Typesetting*, volume A. Addison-Wesley, 1986
- [2] FRANK MITTELBACH and MICHEL GOOSSENS. *The \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 . \currentxt } { 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 } { 2015/12/20 } { }
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 } }

```

(End definition for \c__ctex_engine_str and \c__ctex_engine_file_str.)

54 </class|ctex>


```

55 <*class|ctex|ctexheading>
56 \RequirePackage { ctexhook , ctexpatch }
57 <!ctexheading>\RequirePackage { fix-cm , everyysel }

```

14.1 内部函数与变量

```

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_dim
\l__ctex_tmp_box
58 \tl_new:N \l__ctex_tmp_tl
59 \int_new:N \l__ctex_tmp_int
60 <!ctexheading>\dim_new:N \l__ctex_tmp_dim
61 \box_new:N \l__ctex_tmp_box

```

(End definition for \l__ctex_tmp_tl and others.)

`\ctex_file_wrapper:nnn` 设置文件操作的 `\catcode` 环境,参数 #1 是设置,#2 是文件操作,#3 是恢复。默认关闭 L^AT_EX3 语法环境,并设置 @ 的 `\catcode` 为 11。

```

62 \cs_new_protected:Npn \ctex_file_wrapper:nnn #1#2#3
63 {
64   \use:x
65   {
66     \ExplSyntaxOff
67     \char_set_catcode_letter:n { 64 }
68     #1
69     \exp_not:n {#2}
70     \bool_if:NTF \l__kernel_expl_bool
71     { \ExplSyntaxOn }
72     { \ExplSyntaxOff }
73     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
74     #3
75   }
76 }

```

(End definition for \ctex_file_wrapper:nnn.)

`\ctex_file_input:n` 输入文件。

```

77 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
78 { \ctex_file_wrapper:nnn { } { \file_input:n {#1} } { } }

```

(End definition for \ctex_file_input:n.)

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

```

79 \cs_new_protected_nopar:Npn \ctex_scheme_input:n #1
80 {
81   \ctex_file_wrapper:nnn
82   { }
83   {
84     \tl_if_exist:NTF \c__ctex_class_tl
85     {
86       \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
87       { \file_input:n { ctex-scheme- #1 .def } }
88     }
89     { \file_input:n { ctex-scheme- #1 .def } }
90   }
91   { }
92 }
93 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

(End definition for \ctex_scheme_input:n.)

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

```

94 <!*beamer>
95 \cs_new_eq:NN \g__ctex_section_depth_flag \c_two
96 <!/beamer>

```

(End definition for `\g__ctex_section_depth_flag`.)

```

97 </class|ctex|ctexheading>
98 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

99 \msg_new:nnnn { ctex } { package-too-old }
100 { Support~package~`#1'~too~old. }
101 {
102   Please~update~an~up~to~date~version~of~the~package~`#1'\
103   using~your~TeX~package~manager~or~from~CTAN.
104 }

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

105 \sys_if_output_pdf:TF
106 { \cs_new_eq:NN \ifctexpdf \if_true: }
107 { \cs_new_eq:NN \ifctexpdf \if_false: }

```

(End definition for `\ifctexpdf`.)

`\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。

```

108 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
109 \ctex_after_end_preamble:n { \cs_set_eq:NN \ctex_if_preamble:TF \use_ii:nn }

```

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

`\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。

```

110 \cs_new:Npn \ctex_set_default_ccwd:Nn #1#2
111 {
112   \dim_compare:nNnTF
113     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
114     { \tl_set:Nn #1 {#2} }
115     { \tl_set:Nn #1 { #2 \ccwd } }
116 }

```

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

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

```

117 \cs_new:Npn \ctex_default_pt:n #1
118 {
119   \exp_after:wN \__ctex_default_pt:w
120   \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
121 }
122 \use:x
123 {
124   \cs_new:Npn \exp_not:N \__ctex_default_pt:w
125     ##1 \tl_to_str:n { pt } ##2 \exp_not:N \q_stop
126     { ##1 \tl_to_str:n { pt } }
127 }

```

(End definition for \ctex_default_pt:n.)

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

```
128 \tl_new:N \l__ctex_encoding_tl
129 \tl_set:Nx \l__ctex_encoding_tl
130 { \sys_if_engine_pdftex:TF { GBK } { UTF8 } }
```

(End definition for \l__ctex_encoding_tl.)

\g__ctex_zhmCJK_bool 是否使用 zhmCJK 宏包。

```
131 \bool_new:N \g__ctex_zhmCJK_bool
```

(End definition for \g__ctex_zhmCJK_bool.)

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

```
132 \tl_new:N \l__ctex_autoindent_tl
```

(End definition for \l__ctex_autoindent_tl.)

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

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

(End definition for \ctex_if_autoindent_touched:F.)

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

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

(End definition for \ctex_zhmap_case:nnn.)

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

```
135 <class>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
136 <ctex>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfPackage }
```

(End definition for \ctex_at_end:n.)

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

```
137 <*class>
138 \clist_new:N \g__ctex_std_options_clist
139 </class>
```

(End definition for \g__ctex_std_options_clist.)

对无效选项给出警告。

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

对过时选项或命令给出警告。

```
144 \msg_new:nnn { ctex } { deprecated-option }
145 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
146 \msg_new:nnn { ctex } { deprecated-command }
147 { Command~ #1 is~ deprecated.\\ #2 }
148 \msg_new:nnn { ctex } { deprecated-environment }
149 { Environment~ `#1'~ is~ deprecated.\\ #2 }
```

150 </class|ctex>

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

```

151 <*class|ctex|ctexsize>
152 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one
153 </class|ctex|ctexsize>

```

(End definition for `\g__ctex_font_size_flag`.)

14.2 宏包选项

```

154 <*class|style>
155 \keys_define:nn { ctex / option }
156 {
157 </class|style>

```

zihao

```

158 <*class|ctex|ctexsize>
159   zihao .choice: ,
160   zihao .value_required:n = true ,
161   zihao / 5 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero } ,
162   zihao / -4 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_one } ,
163   zihao / false .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
164 <ctexsize> }
165 </class|ctex|ctexsize>
166 <*class|ctex>
167   c5size .code:n =
168   {
169     \msg_warning:nnn { ctex } { deprecated-option }
170     { Option~`zihao=5'~ is~ set. }
171     \keys_set:nn { ctex / option } { zihao = 5 }
172   } ,
173   cs4size .code:n =
174   {
175     \msg_warning:nnn { ctex } { deprecated-option }
176     { Option~`zihao=-4'~ is~ set. }
177     \keys_set:nn { ctex / option } { zihao = -4 }
178   } ,
179   c5size .value_forbidden:n = true ,
180   cs4size .value_forbidden:n = true ,

```

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

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

```

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

```

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

autoindent 自动调整段落的首行缩进功能。

```

184   autoindent .choice: ,
185   autoindent .default:n = { true } ,
186   autoindent / true .code:n =
187   {
188     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
189     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
190   } ,
191   autoindent / false .code:n =
192   {
193     \tl_clear:N \l__ctex_autoindent_tl
194     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n

```

```

195     } ,
196     autoindent / unknown .code:n =
197     {
198         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
199         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
200     } ,

```

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

indent 仅为兼容性保留,已过时。

```

201     indent .code:n =
202     {
203         \msg_warning:nnn { ctex } { deprecated-option }
204         {
205             The~ functionality~ has~ been~ removed.\\
206             It's~ better~ to~ set~ the~ heading~ styles~ via~ `afterindent'~
207             options.~
208         }
209     } ,
210     indent .value_forbidden:n = true ,
211     noindent .code:n =
212     {
213         \msg_warning:nnn { ctex } { deprecated-option }
214         {
215             The~ functionality~ has~ been~ removed.\\
216             It's~ better~ to~ set~ the~ heading~ styles~ via~ `afterindent'~
217             options.~
218         }
219     } ,
220     noindent .value_forbidden:n = true ,

```

(End definition for indent. This function is documented on page 28.)

GBK

UTF8

```

221     GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
222     UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
223     GBK .value_forbidden:n = true ,
224     UTF8 .value_forbidden:n = true ,

```

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

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

```

225     fontset .tl_gset:N = \g__ctex_fontset_tl ,
226     nofonts .code:n =
227     {
228         \msg_warning:nnn { ctex } { deprecated-option }
229         {
230             Option~ `fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
231             fontset~ option.~
232         }
233         \keys_set:nn { ctex / option } { fontset = none }
234     } ,
235     adobefonts .code:n =
236     {
237         \msg_warning:nnn { ctex } { deprecated-option }
238         {
239             Option~ `fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
240             fontset~ option.~
241         }
242         \keys_set:nn { ctex / option } { fontset = adobe }
243     } ,
244     winfonts .code:n =

```

```

245     {
246       \msg_warning:nnn { ctex } { deprecated-option }
247       {
248         Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
249         fontset~ option.
250       }
251       \keys_set:nn { ctex / option } { fontset = windows }
252     } ,
253     nofonts      .value_forbidden:n = true ,
254     winfonts     .value_forbidden:n = true ,
255     adobefonts   .value_forbidden:n = true ,

```

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

zhmap

```

256     zhmap .choice: ,
257     zhmap .default:n = { true } ,
258     zhmap / zhmCJK .code:n =
259     {
260       \bool_gset_true:N \g__ctex_zhmCJK_bool
261       \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
262     } ,
263     zhmap / true .code:n =
264     {
265       \bool_gset_false:N \g__ctex_zhmCJK_bool
266       \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
267     } ,
268     zhmap / false .code:n =
269     {
270       \bool_gset_false:N \g__ctex_zhmCJK_bool
271       \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
272     } ,
273     nozhmap .code:n =
274     {
275       \msg_warning:nnn { ctex } { deprecated-option }
276       { Option~`zhmap=false'~ is~ set. }
277       \keys_set:nn { ctex / option } { zhmap = false }
278     } ,
279     nozhmap .value_forbidden:n = true ,

```

(End definition for `zhmap`. This function is documented on page 28.)

punct 设置标点符号输出格式。

```

280     punct .tl_set:N = \l__ctex_punct_tl ,
281     punct .default:n = { quanjiao } ,
282     punct .initial:n = { quanjiao } ,
283     nopunct .code:n =
284     {
285       \msg_warning:nnn { ctex } { deprecated-option }
286       { Option~`punct=plain'~ is~ set. }
287       \keys_set:nn { ctex / option } { punct = plain }
288     } ,
289     nopunct .value_forbidden:n = true ,

```

(End definition for `punct`. This function is documented on page 28.)

space

```

290     space .choices:nn =
291     { true , auto , false }
292     {
293       \exp_args:Nx \ctex_at_end:n
294       { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
295     } ,

```

```

296     space .default:n = { true } ,
297     nospace .code:n =
298     {
299         \msg_warning:nnn { ctex } { deprecated-option }
300         { Option~ `space=false'~ is~ set. }
301         \keys_set:nn { ctex / option } { space = false }
302     } ,
303     nospace .value_forbidden:n = true ,

```

(End definition for `space`. This function is documented on page 28.)

heading

```

304     heading .bool_set:N = \l__ctex_heading_bool ,

```

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

```

305 </class|ctex>
306 <*class|ctex|ctexheading>

```

sub3section

sub4section

```

307 <!*beamer>
308     sub3section .code:n =
309     { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
310     sub4section .code:n =
311     { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
312     sub3section .value_forbidden:n = true ,
313     sub4section .value_forbidden:n = true ,
314 <!/beamer>

```

(End definition for `sub3section` and `sub4section`. These functions are documented on page 9.)

scheme

```

315     scheme .tl_set:N = \l__ctex_scheme_tl ,
316 <*ctexheading>
317     scheme .default:n = { plain } ,
318     scheme .initial:n = { plain }
319 }
320 </ctexheading>
321 <!*ctexheading>
322     scheme .default:n = { chinese } ,
323     scheme .initial:n = { chinese } ,
324 <!/ctexheading>
325 </class|ctex|ctexheading>
326 <*class|ctex>
327     cap .code:n =
328     {
329         \msg_warning:nnn { ctex } { deprecated-option }
330         { Option~ `scheme = chinese' ~ is~ set. }
331         \keys_set:nn { ctex / option } { scheme = chinese }
332     } ,
333     nocap .code:n =
334     {
335         \msg_warning:nnn { ctex } { deprecated-option }
336         { Option~ `scheme = plain' ~ is~ set. }
337         \keys_set:nn { ctex / option } { scheme = plain }
338     } ,
339     cap .value_forbidden:n = true ,
340     nocap .value_forbidden:n = true ,

```

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

fntef 这些都是过时的宏包兼容选项,原选项功能总是打开的。
fancyhdr
hyperref

```

341 fntef .code:n =
342 {
343   \msg_warning:nnn { ctex } { deprecated-option }
344   { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
345 },
346 fancyhdr .code:n =
347 {
348   \msg_warning:nnn { ctex } { deprecated-option }
349   { `fancyhdr'~ package~ is~ loaded. }
350   \RequirePackage { fancyhdr }
351 },
352 hyperref .code:n =
353 {
354   \msg_warning:nnn { ctex } { deprecated-option }
355   { `hyperref'~ package~ will~ be~ loaded. }
356   \ctex_at_end:n
357   {
358     \cs_if_exist:NF \hypersetup
359     { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
360   }
361   \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
362 },
363 }
```

(End definition for *fntef*, *fancyhdr*, and *hyperref*. These functions are documented on page 29.)

```

364 </class>ctex>
365 <*class>ctex|ctexsize>
```

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

```

366 \tl_clear:N \l__ctex_tmp_tl
367 \clist_map_inline:nn
368 {
369   10pt , 11pt , 12pt ,
370   8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
371 }
372 {
373   \tl_put_right:Nn \l__ctex_tmp_tl
374   {
375     #1 .code:n =
376     <*!class>
377       { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
378     </!class>
379     <*class>
380       {
381         \cs_gset_eq:NN \g__ctex_font_size_flag \c_two
382         \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
383       } ,
384     </class>
385     #1 .value_forbidden:n = true ,
386   }
387 }
388 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
389 \tl_clear:N \l__ctex_tmp_tl
```

(End definition for 10pt, 11pt, and 12pt. These functions are documented on page 8.)

将未知选项传给标准文档类。

```

390 <*class>
391 \keys_define:nn { ctex / option }
392 {
393   unknown .code:n =
```

```

394     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
395   }
396 </class>

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

399 <*class|style>
400 \ProcessKeysOptions { ctex / option }
401 </class|style>

402 <*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。

```

403 \if_case:w \g__ctex_font_size_flag
404   \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
405 \or:
406   \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
407 \fi:

```

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

```

408 <*article>
409 \tl_const:Nn \c__ctex_class_tl { article }
410 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
411 \LoadClass { article }
412 </article>
413 <*book>
414 \tl_const:Nn \c__ctex_class_tl { book }
415 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
416 \LoadClass { book }
417 </book>
418 <*report>
419 \tl_const:Nn \c__ctex_class_tl { report }
420 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
421 \LoadClass { report }
422 </report>
423 <*beamer>
424 \tl_const:Nn \c__ctex_class_tl { beamer }
425 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
426 \LoadClass { beamer }
427 </beamer>

428 </class>

```

14.3 用户设置接口

\ctexset

```

429 <*class|ctex|ctexheading>
430 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
431 </class|ctex|ctexheading>

```

(End definition for \ctexset. This function is documented on page 5.)

\CTEXsetup 过时命令。
\CTEXoptions

```

432 <*class|ctex>
433 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
434 {
435   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
436   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
437   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
438 }
439 \NewDocumentCommand \CTEXoptions { +o }
440 {

```

```

441 \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
442 { \ctexset~ {~ #1~ }~ is~ set. }
443 \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
444 }
445 </class|ctex>

```

(End definition for \CTEXsetup and \CTEXoptions. These functions are documented on page 29.)

14.4 特定引擎支持与设置

14.4.1 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。

```

446 <*pdfTeX|upTeX|apTeX>
447 \cs_new_protected_nopar:Npn \ctex_set_zhmap:n #1
448 {
449   \AtBeginDvi { #1 }
450   \ctex_at_end_package:nn { atbegshi }
451   { \AtBeginShipoutFirst { #1 } }
452 }
453 \@onlypreamble \ctex_set_zhmap:n
454 </pdfTeX|upTeX|apTeX>

```

(End definition for \ctex_set_zhmap:n.)

```
455 <*pdfTeX>
```

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

```

456 \seq_new:N \c__ctex_cmap_encoding_seq
457 \seq_set_from_clist:Nn \c__ctex_cmap_encoding_seq
458 { C19 , C10 , C00 , C09 , C40 , C60 }

```

(End definition for \c__ctex_cmap_encoding_seq.)

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

```

459 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2
460 {
461   \cs_if_free:cF { #1 + #2 }
462   {
463     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
464     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
465   }
466 }
467 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
468 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn

```

(End definition for \ctex_family_cmap:nn.)

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

```

469 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
470 {
471   \cs_if_free:NF \CJK@plane
472   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
473 }
474 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
475 {
476   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
477   #1
478 }

```

```

479 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
480 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
481 {
482   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
483   \tex_immediate:D \pdfTeX_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
484   \cs_new_protected_nopar:Npx #1
485   {
486     \exp_not:N \pdfTeX_pdffontattr:D \exp_not:N \tex_font:D
487     { /ToUnicode ~ \int_use:N \pdfTeX_pdflastobj:D \c_space_tl 0 ~ R }
488   }
489 }

```

(End definition for \ctex_add_cmap:n.)

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

```

490 \group_begin:
491 \char_set_catcode_other:N \#
492 \sys_if_output_pdf:TF
493 {
494   \group_end:
495   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
496   { \CTEX@Family@CMap {#1} {#2} }
497   {
498     \ctex_at_end_package:nn { cmap }
499     { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
500   }
501   { \ctex_patch_failure:N \DeclareFontFamily }
502 }
503 { \group_end: }

```

(End definition for \DeclareFontFamily.)

首先检查选项,决定是否载入 zhmCJK 宏包。

```

504 \if_bool:N \g__ctex_zhmCJK_bool
505 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
506 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

507 \else:
508   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
509   { \RequirePackage { CJK } }
510   { \RequirePackage { CJKutf8 } }
511   \RequirePackage { CJKpunct , CJKspace }

```

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

```

512 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
513 {
514   \tl_set:Nn \CJKrmdefault {#1}
515   \tl_set:Nn \CJKsfdefault {#2}
516   \tl_set:Nn \CJKttdefault {#3}
517   \ctex_set_zhmap:n { \file_input:n {#4} }
518 }
519 \@onlypreamble \ctex_load_zhmap:nnnn

```

(End definition for \ctex_load_zhmap:nnnn.)

```

520 \tl_if_exist:NF \CJKfamilydefault
521 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
522 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
523 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
524 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
525 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
526 { \CJKfamily { \CJKrmdefault } }
527 { }

```

```

528 { \ctex_patch_failure:N \rmfamily }
529 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
530 { \CJKfamily { \CJKsfdefault } }
531 { }
532 { \ctex_patch_failure:N \sffamily }
533 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
534 { \CJKfamily { \CJKttdefault } }
535 { }
536 { \ctex_patch_failure:N \ttfamily }
537 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
538 { \CJKfamily { \CJKfamilydefault } }
539 { \cs_set_eq:NN \reset@font \normalfont }
540 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```
541 \fi:
```

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

```

542 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
543 {
544   \ctex_file_wrapper:nnn
545   {
546     \char_set_catcode_other:n { 60 } % <
547     \char_set_catcode_math_superscript:n { 94 } % ^
548     \int_set_eq:NN \tex_endlinechar:D \c_minus_one
549   }
550   { \file_input:n {#1} }
551   {
552     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
553     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
554     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
555   }
556 }
557 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

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

`\ctex_plane_to_utfxvibe:Nn` `\CJK@surr` **fancyhdr** 宏包的 `\nouppercase` 会将 `\uppercase` 定义为 `\relax`, 而 `\CJK@surr` 需要用它将 `\CJK@plane` 转化成大写字母, 这就造成了冲突¹⁶。我们在这里给出 `\CJK@surr` 的一个不依赖 `\uppercase` 的实现。

```

558 \if_cs_exist:N \CJK@surr
559 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
560 {
561   \tl_set:Nx \l__ctex_tmp_tl {#2}
562   \int_set:Nn \l__ctex_tmp_int
563   { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
564   \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
565   { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
566   {
567     \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
568     \tl_gset:Nx #1
569     {
570       \int_to_Hex:n
571       { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
572       \int_to_Hex:n
573       { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
574     }
575   }
576 }

```

¹⁶<https://github.com/CTeX-org/ctex-kit/issues/146>

```

577 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
578 \fi:

```

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

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

```

579 \AtBeginDocument
580 {
581   \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
582   { \punctstyle { \l__ctex_punct_tl } }
583 }

```

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

```

584 \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。

```

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

```

在导言区结束时调用 \CJK@envStart 启用完整的中文功能。

\CJK@envStart 的定义是

```

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

```

\CJK@upperReset 可能会有一定风险, 因此我们直到导言区末尾才使用 \CJK@envStart。这样可以避免将 CJK 环境内置入 document 环境的最里层, 最后也就不需要 \clearpage。zhm-CJK 已经提供类似功能。注意先使用 \ctex_update_default_family: 更新 \CJKfamilydefault。

```

591 \exp_args:Nx \ctex_at_end_preamble:n
592 {
593   \exp_not:N \CJK@envStart
594   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
595   \exp_not:N \CJKtilde
596 }

```

zhmCJK 判断结束。

```

597 \fi:

```

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

```
598 <@@=>
599 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@ignorespaces
```

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

```
600 <@@=ctex>
```

(End definition for \ctex_auto_ignorespaces:.)

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

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

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

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

```
609 \ctex_gadd_hook:Nn \CJKhook { \ctex_set_ignorespaces: }
```

(End definition for \CJKhook.)

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

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

(End definition for \ctex_punct_set:n.)

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

```
627 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
628 {
629   \cs_if_free:cF { CJKpunct@ #2 @spaces }
630   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
631 }
```

(End definition for \ctex_punct_map_family:nn.)

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

```

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

```

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

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

```

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

```

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

`\ctex_punct_space:nn` 定义标点的边界信息。

`\ctexspadef`

```

658 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
659 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
660 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

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

载入边界信息文件。

```

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

```

14.4.2 ctex-engine-xetex.def

```

663 <*xetex>
664 \RequirePackage { xeCJK }
665 \exp_args:Nx \xeCJKsetup
666 {
667   LoadFandol = false ,
668   AutoFakeBold = true ,
669   PunctStyle = \l__ctex_punct_tl
670 }

```

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`。

```

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

```

673 `</xetex>`

14.4.3 ctex-engine-luatex.def

674 `<*luatex>`

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

```
675 \msg_new:nnn { ctex } { luatexja-loaded }
676 {
677   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
678   Loading~file~`#1'~will~abort!
679 }
680 \ifpackageloaded { luatexja }
681 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
682 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
683 \RequirePackage { luatexja }
684 \ifpackagelater { luatexja } { 2015/09/21 } { }
685 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
686 \RequirePackage { fontspec }
687 \ifpackagelater { fontspec } { 2014/05/25 } { }
688 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
```

14.4.3.1 LuaTeX-jā 的默认设置

689 `\ExplSyntaxOff`

以下设置抄录自 ltjdefs.sty。

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

以下设置抄录自 ltj-latex.sty。

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

14.4.3.2 LuaTeX-jā 的补丁

719 `<@@=ctex_ltj>`

在 L^AT_EX 下, LuaTeX-jā 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`。

```

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

```

对 `fontspec` 沿用 `LuaTeX-ja` 的补丁。但恢复对 `\emshape` 和 `\eminnershape` 的修改。

```

728 \RequirePackage { lltjp-fontspec }
729 \cs_set:Npn \emshape { \itshape }
730 \cs_set:Npn \eminnershape { \upshape }

```

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

```

731 \cs_new_protected_nopar:Npn \__ctex_ltj_um_define_math_chars:
732 {
733   \group_begin:
734   \cs_set_protected:Npn \__um_sym:nnn ##1##2##3
735   {
736     \tl_if_in:nnT
737     {
738       \mathord \mathalpha \mathbin \mathrel
739       \mathpunct \mathop \mathfence
740     }
741     { ##3 }
742     { \__ctex_ltj_um_char:Nn ##2 { ##1 } }
743   }
744   \__um_input_math_symbol_table:
745   \group_end:
746 }
747 \cs_new_protected:Npn \__ctex_ltj_um_char:Nn #1#2
748 {
749   \__ctex_ltj_um_char_aux:Nx #1 { \char_generate:nn {#2} { 12 } }
750   \ltjsetmathletter {#2}
751 }
752 \cs_new_protected:Npn \__ctex_ltj_um_char_aux:Nn #1#2
753 {
754   \cs_gset_protected_nopar:Npn #1
755   {
756     \mode_if_math:TF
757     {#2}
758     {
759       {
760         \lua_now_x:n { tex.globaldefs = 0 }
761         \ltj@allalchar #2
762       }
763     }
764   }
765 }
766 \cs_generate_variant:Nn \__ctex_ltj_um_char_aux:Nn { Nx }
767 \ctex_at_end_package:nn { unicode-math }
768 {
769   \cs_set_eq:NN \__um_define_math_chars: \__ctex_ltj_um_define_math_chars:
770   \@ifpackagelater { unicode-math } { 2015/06/28 }
771   {
772     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
773     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
774     { \__um_switchto_literal: }
775   }
776   { }
777 }

```

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

```

778 \ctex_at_end_package:n { listings }
779 {
780   \use:x
781   {
782     \exp_not:N \RequirePackage { lltjp-listings }
783     \tl_set:Nn \exp_not:N \lstlistingname
784       { \exp_not:o { \lstlistingname } }
785     \tl_set:Nn \exp_not:N \lstlistlistingname
786       { \exp_not:o { \lstlistlistingname } }
787   }
788 }

```

14.4.3.3 字体切换方式

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

```

\CTK@family
789 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
790 {
791   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
792   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
793 }
794 \tl_new:N \CJK@family
795 \tl_new:N \l__ctex_ltj_current_font_tl
796 \tl_set:Nn \l__ctex_ltj_current_font_tl
797 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

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

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

```

798 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
799 {
800   \group_begin:
801     \tl_set_eq:NN \f@encoding \CJK@encoding
802     \tl_set_eq:NN \f@family \CJK@family
803     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
804     \ctex_ltj_pickup_font:
805   \group_end:
806   \font@name
807   \__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。

```

808   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
809   { \__ctex_ltj_select_font_aux: }
810 }
811 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
812 {
813   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
814   \cs_gset_nopar:Npx \font@name {#1}
815 }
816 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
817 { \cs_gset_eq:NN \font@name \__ctex_ltj_save_fontname:w }

```

(End definition for __ctex_ltj_select_font_aux:.)

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

```

818 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
819 {

```

```

820 \exp_after:wN \cs_if_exist:NF \font@name
821 {
822   \group_begin:
823   \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
824   \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
825   \define@newfont
826   \group_end:
827 }
828 }
829 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

(End definition for \ctex_ltj_pickup_font:.)

```

\ctex_ltj_extract_font: LuaTeX-j_a 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}')}
\def\ltj@@jfont{\directlua{luatexja.jfont.jfontdefY()}}

```

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

```

tex.sprint(cat_lp, global_flag, '\protected\expandafter\def\csname ',
  (cstemp==' ') 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。

```

830 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
831 {
832   \get@external@font
833   \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
834   {
835     \tl_set:Nx \external@font
836     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
837   }
838   \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。

```

839 \font@name
840 \lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
841 \use:c { \f@encoding + \f@family }
842 \use:c { \curr@fontshape }
843 }

```

(End definition for \ctex_ltj_extract_font:.)

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

```

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

```

```

\aftergroup\subst@correction
}

```

我们在这里不需要定义新字体,而是设置对应字体的命令。

```

844 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
845 {
846   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
847   {
848     \group_begin:
849     \tl_set_eq:NN \CJK@family \f@family
850     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
851     {
852       \cs_gset_protected_nopar:Npx \subst@correction
853       {
854         \cs_new_eq:NN
855         \exp_not:c { \l__ctex_ltj_current_font_tl }
856         \font@name
857       }
858       \group_insert_after:N \group_insert_after:N
859       \group_insert_after:N \subst@correction
860     }
861     \group_end:
862   }
863 }

```

(End definition for \ctex_ltj_subst_font:.)

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

```

864 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
865 {
866   \lua_now_x:n { luatexja.jfont.does_alt_set (' \lua_escape_x:n { #1 }') }
867   \prg_return_true: \else: \prg_return_false: \fi:
868 }

```

(End definition for \ctex_ltj_if_alternate_shape_exist:nTF.)

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

```

function print_aftl_address()
  tex.sprint(cat_lp, ';ltjaltfont' .. tostring(aftl_base):sub(8))
end

```

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

```

869 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
870 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for __ctex_ltj_patch_external_font:w.)

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

```

871 \cs_new_protected_nopar:Npn \ctex_ltj_select_alternate_font:
872 {
873   \ctex_ltj_if_alternate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
874   {
875     \lua_now_x:n
876     {
877       luatexja.jfont.output_alt_font_cmd
878       ('y', '\lua_escape_x:n { \l__ctex_ltj_current_shape_tl }')
879     }
880     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size') }
881   }

```

```

882 }
883 \tl_new:N \l__ctex_ltj_current_shape_tl
884 \tl_set:Nn \l__ctex_ltj_current_shape_tl
885 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for \ctex_ltj_select_alterate_font:.)

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

```

886 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
887 {
888   \cs_if_exist:cF { #1/\f@size }
889   {
890     \group_begin:
891     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
892     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
893     \ctex_ltj_pickup_font:
894     \group_end:
895     \__ctex_ltj_pop_fontname:
896   }
897 }

```

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

898 <@@=>

\ltj@pickup@altfont@copy 被用在函数 pickup_alt_font_a 中。 \ltj@@getjfontnumber 的作用是将字体命令 #1 对应的 font.id 保存到 \ltj@tempcntc 中。

```

899 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
900 {
901   \ltj@@getjfontnumber #1
902   \lua_now_x:n
903   {
904     luatexja.jfont.pickup_alt_font_b
905     ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
906   }
907 }

```

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

14.4.3.4 数学字体族

908 <@@=ctex_ltj>

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

\ctex_ltj_if_jfont:nTF 参数 #1 是一个 L^AT_EX_{2_ε} 编码名称或者字体命令。L^AT_EX_{2_ε} 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。

```

909 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
910 {
911   \lua_now_x:n
912   { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[~/]+') ) }
913   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
914 }

```

(End definition for \ctex_ltj_if_jfont:nTF.)

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


```

915 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
916 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
917 \group_begin:
918 \char_set_catcode_other:N M
919 \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
920 { \ctex_ltj_if_jfont:nTF {#3} }
921 \group_end:

```

(End definition for \ctex_ltj_if_jfont_math:NTF.)

922 <@@@=>

\getanddefine@fonts
\ctex_ltj_get_and_define_fonts:nN

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。

```

923 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
924 {
925   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
926   { \ctex_ltj_get_and_define_fonts_ja:nN }
927   { \ctex_ltj_get_and_define_fonts_al:nN }
928   {#1} #2
929 }
930 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
931 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
932 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
933 {
934   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
935   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
936   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
937   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
938   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
939   \ctex_ltj_pickup_font:
940   \tl_put_right:Nx \math@fonts
941   {
942     \ltj@setpar@global
943     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
944     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
945     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
946   }
947 }

```

(End definition for \getanddefine@fonts and \ctex_ltj_get_and_define_fonts:nN.)

948 <@@@=ctex_ltj>

\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。

```

949 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
950 {
951   \mode_if_math:T
952   {
953     \math@bgroup
954     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
955     \ctex_ltj_math_group_hook:
956     \ctex_ltj_if_jfont_math:NTF #1
957     { \jfam } { \mathgroup } #2 \scan_stop:
958     \math@egroup
959   }
960 }
961 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
962 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

(End definition for \use@mathgroup and \ctex_ltj_use_math_group:Nn.)

14.4.3.5 字体族的定义与使用

`\ctex_mono_jfm:n` LuaTeX-j_a 中与标点格式 plain 对应的 JFM 是 mono。
`\l__ctex_ltj_jfm_tl`

```

963 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
964   {
965     \str_if_eq:nnTF {#1} { plain }
966       { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
967       { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
968   }
969 \tl_new:N \l__ctex_ltj_jfm_tl
970 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
971 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

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

`\CJK@encoding` 在 L^AT_EX 下, LuaTeX-j_a 依赖字体编码来实现特殊设置。例如上述的 `\ctex_ltj_if_-`
`__ctex_ltj_change_encoding:` `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。

```

972 \tl_const:Nn \CJK@encoding { LTJY3 }
973 \DeclareFontEncoding { \CJK@encoding } { } { }
974 \use:x
975   {
976     \exp_not:N \DeclareFontSubstitution
977       { \CJK@encoding } { song } { \mddefault } { \updefault }
978   }
979 \lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
980 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
981   { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
982 \DeclareFontFamily { \CJK@encoding } { song } { }
983 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
984   { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
985 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
986   { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
987 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
988 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
989   { \CJK@encoding } { song } { \mddefault } { \updefault }
990 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
991   { \CJK@encoding } { song } { \bfdefault } { \updefault }
992 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
993 \jfam \c__ctex_ltj_math_fam_int

```

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

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

994 \newfontfeature { CID }      { cid = #1 }
995 \newfontfeature { JFM }      { jfm = #1 }
996 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

997 \keys_define:nn { fontspec-parse-external }
998   {
999     NoEmbed .code:n =
1000       { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
1001   }
1002 \cs_new:Npn \__ctex_ltj_noembed_wrap:n #1 { psft: #1 }

```

`\ctex_ltj_set_family:nnn` 将自定义的字体族名与 `fontspec` 实际设置的名字对应起来。

```

1003 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1004   {
1005     \group_begin:
1006     \clist_clear:N \l__ctex_ltj_char_range_clist

```

```

1007 \seq_clear:N \l__ctex_ltj_alternate_seq
1008 \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1009 \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1010 \clist_set:Nn \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1011 \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1012 {
1013   \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1014   \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1015     {#1} { \l__ctex_ltj_font_options_clist }
1016   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1017   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1018   \__ctex_ltj_gset_family_cs:nn {#1} {#3}
1019 }
1020 \group_end:
1021 }
1022 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1023 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1024 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1025 \clist_new:N \l__ctex_ltj_font_options_clist
1026 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1027 {
1028   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1029   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1030 }

```

(End definition for \ctex_ltj_set_family:nnn.)

\g__ctex_ltj_family_name_prop 分别保存 fontspec 设置的字体族名、字体名称和字体选项。
 \g__ctex_ltj_family_font_name_prop
 \g__ctex_ltj_family_font_options_prop

```

1031 \prop_new:N \g__ctex_ltj_family_name_prop
1032 \prop_new:N \g__ctex_ltj_family_font_name_prop
1033 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

(End definition for \g__ctex_ltj_family_name_prop, \g__ctex_ltj_family_font_name_prop, and \g__ctex_ltj_family_font_options_prop.)

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

```

1034 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1035 {
1036   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1037   {
1038     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1039     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1040     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1041     {
1042       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1043       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1044       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1045       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1046     }
1047     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1048   }
1049 }
1050 \tl_new:N \l__ctex_ltj_tmp_tl
1051 \msg_new:nnn { ctex } { redefine-family }
1052 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

(End definition for __ctex_ltj_check_family:n.)

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

```

1053 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1054 {
1055   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }

```

```

1056 {
1057   \group_begin:
1058   \__ctex_ltj_change_encoding:
1059   \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1060   \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1061   { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1062   \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1063   { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1064   \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1065   \exp_not:N \g__ctex_ltj_fontspec_family_tl
1066   \__ctex_ltj_set_alternate_family:n {#1}
1067   \group_end:
1068 }
1069 }
1070 \tl_new:N \l__ctex_ltj_base_family_tl
1071 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1072 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1073 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1074 {
1075   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1076   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1077   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1078   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1079 }
1080 \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

(End definition for __ctex_ltj_gset_family_cs:nn.)

\CJKfamily 切换字体。

```

1081 \NewDocumentCommand \CJKfamily { m }
1082 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1083 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1084 {
1085   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1086   {
1087     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1088     \selectfont
1089   }
1090   { \__ctex_ltj_family_unknown_warning:n {#1} }
1091 }
1092 \tl_new:N \l_ctex_ltj_family_tl
1093 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

(End definition for \CJKfamily.)

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

```

1094 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1095 {
1096   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1097   { \prg_return_true: }
1098   {
1099     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1100     {
1101       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1102       \prg_return_true:
1103     }
1104     { \prg_return_false: }
1105   }
1106 }
1107 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

__ctex_ltj_family_unknown_warning:n

```

1108 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1109 {
1110   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1111   {
1112     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1113     {
1114       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1115       \msg_warning:nnn { ctex } { family-unknown } {#1}
1116     }
1117   }
1118 }
1119 \seq_new:N \g__ctex_ltj_unknown_family_seq
1120 \msg_new:nnn { ctex } { family-unknown }
1121 {
1122   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1123   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1124 }
1125 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1126 {
1127   \str_case:x:nnF {#1}
1128   {
1129     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1130     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1131     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1132   }
1133   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1134   [...] \{...\}
1135 }
1136 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1137 {
1138   \str_case:x:nnF {#1}
1139   {
1140     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1141     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1142     \CJKttdefault { \token_to_str:N \CJKttdefault }
1143   }
1144   {#1}
1145 }

```

(End definition for __ctex_ltj_family_unknown_warning:n.)

\ctex_ltj_fontspec:nn

```

1146 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1147 {
1148   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1149   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1150   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1151   {
1152     \int_gincr:N \g__ctex_ltj_family_int
1153     \__ctex_ltj_fontspec:xnn
1154     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1155     {#1} {#2}
1156   }
1157 }
1158 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1159 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1160 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1161 {
1162   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1163   {
1164     \cs_if_free:cF
1165     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1166     {
1167       \cs_gset_eq:cc
1168       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1169       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }

```

```

1170         \cs_gset_eq:cc
1171         { \l__ctex_ltj_altername_cs:n { clear / #1 } }
1172         { \l__ctex_ltj_altername_cs:n { clear / \l_ctex_ltj_family_tl } }
1173     }
1174     \bool_set_false:N \l__ctex_ltj_add_altername_bool
1175 }
1176 \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1177 \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1178 \ctex_ltj_switch_family:n {#1}
1179 }
1180 \cs_generate_variant:Nn \l__ctex_ltj_fontspec:nnn { x }
1181 \prop_new:N \g__ctex_ltj_fontspec_prop

```

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

`\ctex_ltj_add_font_features:n`

`\ctex_ltj_add_font_features:nn`

```

1182 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1183 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1184 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1185 {
1186     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1187     {#1} \l__ctex_ltj_tmp_tl
1188     {
1189         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1190         {#1} \l__ctex_ltj_font_options_clist
1191         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1192         \bool_set_true:N \l__ctex_ltj_add_altername_bool
1193         \ctex_ltj_fontspec:xx
1194         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1195         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1196     }
1197     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1198 }
1199 \bool_new:N \l__ctex_ltj_add_altername_bool
1200 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1201 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1202 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1203 {
1204     \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1205     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1206 }

```

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

`\setCJKfamilyfont`

`\newCJKfontfamily`

`\CJKfontspec`

`\addCJKfontfeatures`

```

1207 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
1208 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1209 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
1210 {
1211     \tl_set:Nx \l__ctex_ltj_tmp_tl
1212     { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1213     \cs_new_protected_nopar:Npx #2
1214     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1215     \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1216 }
1217 \NewDocumentCommand \CJKfontspec { O { } m }
1218 {
1219     \ctex_ltj_fontspec:xx {#1} {#2}
1220     \tex_ignorespaces:D
1221 }
1222 \NewDocumentCommand \addCJKfontfeatures { m }
1223 {
1224     \ctex_ltj_add_font_features:x {#1}
1225     \tex_ignorespaces:D
1226 }

```

```

1227 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

(End definition for \setCJKfamilyfont and others.)

\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1228 \NewDocumentCommand \setCJKmainfont { 0 { } m }
1229 {
1230   \ctex_ljtj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1231   \normalfont
1232 }
1233 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1234 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1235 {
1236   \ctex_ljtj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1237   \normalfont
1238 }
1239 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1240 {
1241   \ctex_ljtj_set_family:xxx { \CJKttdefault } {#1} {#2}
1242   \normalfont
1243 }
1244 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1245 { \ctex_ljtj_set_family:xxx { \c__ctex_ljtj_math_tl } {#1} {#2} }
1246 \NewDocumentCommand \defaultCJKfontfeatures { m }
1247 { \clist_gset:Nn \g__ctex_ljtj_default_features_clist {#1} }
1248 \clist_new:N \g__ctex_ljtj_default_features_clist
1249 \@onlypreamble \setCJKmainfont
1250 \@onlypreamble \setCJKsansfont
1251 \@onlypreamble \setCJKmonofont
1252 \@onlypreamble \setCJKmathfont
1253 \@onlypreamble \setCJKromanfont
1254 \@onlypreamble \defaultCJKfontfeatures

(End definition for \setCJKmainfont and others.)

1255 \tl_if_exist:NF \CJKfamilydefault
1256 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1257 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1258 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1259 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1260 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1261 { \CJKfamily { \CJKrmdefault } }
1262 { }
1263 { \ctex_patch_failure:N \rmfamily }
1264 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1265 { \CJKfamily { \CJKsfdefault } }
1266 { }
1267 { \ctex_patch_failure:N \sffamily }
1268 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1269 { \CJKfamily { \CJKttdefault } }
1270 { }
1271 { \ctex_patch_failure:N \ttfamily }
1272 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1273 { \CJKfamily { \CJKfamilydefault } }
1274 { \cs_set_eq:NN \reset@font \normalfont }
1275 { \ctex_patch_failure:N \normalfont }

```

在导言区末尾更新 \CJKfamilydefault。

```

1276 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

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

```

1277 \cs_new_protected_nopar:Npn \ctex_ljtj_ensure_default_family:
1278 {
1279   \prop_if_empty:NF \g__ctex_ljtj_family_font_name_prop

```



```

1280 {
1281   \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmpa_tl
1282   {
1283     \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1284     { \use:n }
1285     {
1286       \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmpa_tl
1287       { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1288       { \use:n }
1289     }
1290     {
1291       \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1292       {
1293         \prop_map_break:n
1294         { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1295       }
1296     }
1297   }
1298   \normalfont
1299   \ctex_ltj_update_mathfont:
1300 }
1301 }

```

(End definition for \ctex_ltj_ensure_default_family:.)

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

```

1302 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1303 {
1304   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1305   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1306   {
1307     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1308     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1309   }
1310 }
1311 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1312 {
1313   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1314   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1315   { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1316   \cs_if_free:cTF
1317   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1318   {
1319     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1320     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1321   }
1322   {
1323     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1324     { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1325   }
1326 }

```

(End definition for \ctex_ltj_update_mathfont:.)

14.4.3.6 替代字体的设置

AlternateFont 设置替代字体的选项。

CharRange

```

1327 \keys_define:nn { ctex_ltj / fontspec }
1328 {
1329   AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1330   AlternateFont .value_required:n = true ,
1331   CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1332   CharRange .value_required:n = true
1333 }

```

(End definition for AlternateFont and CharRange. These functions are documented on page 25.)

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

```

1334 \group_begin:
1335   \char_set_catcode_other:N \l
1336   \cs_set:Npn \__ctex_ltj_tmp:w #1
1337   {
1338     \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1339     {
1340       \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1341       {
1342         \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1343         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1344         \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1345         \seq_set_filter:NnN \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1346           { ! \tl_if_blank_p:n { #####1 } }
1347         \seq_concat:NNN \l__ctex_ltj_alternate_seq
1348           \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1349       }
1350     }
1351   }
1352   \char_set_catcode_active:N \l
1353   \__ctex_ltj_tmp:w { || }
1354 \group_end:
1355 \seq_new:N \l__ctex_ltj_tmp_seq
1356 \seq_new:N \l__ctex_ltj_alternate_seq

```

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

`\ctex_ltj_set_alternate_family:nnF` 如果在字体的选项中设置了 CharRange,则只设置替代字体。

```

1357 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1358 {
1359   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1360   {
1361     \__ctex_ltj_check_family:n {#1}
1362     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1363     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1364     #3
1365   }
1366   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1367 }

```

(End definition for `\ctex_ltj_set_alternate_family:nnF`.)

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

```

\ctex_ltj_save_alternate_seq:Nnnwn
1368 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1369 {
1370   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1371   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1372 }
1373 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1374 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1375 { m m m +O{ } m u{ \q_stop } }
1376 {
1377   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1378   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1379   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1380   \tl_if_blank:nTF {#5}
1381   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1382   {
1383     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1384     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}

```

```

1385     }
1386     \use:x
1387     {
1388         \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1389         { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1390         { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1391         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1392     }
1393 }
1394 \clist_new:N \l__ctex_ltj_alternate_options_clist

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

```

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

```

1395 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1396 {
1397     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1398     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1399     \ctex_ltj_set_alternate_family:coonn
1400     { \__ctex_ltj_alternate_cs:n {#1} }
1401     { \l__ctex_ltj_char_range_clist }
1402     { \l__ctex_ltj_font_options_clist } {#2} {#1}
1403 }
1404 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1405 {
1406     \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1407     { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1408     \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1409 }
1410 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

(End definition for \ctex_ltj_set_alternate_family:nn.)

```

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

```

1411 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1412 {
1413     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1414     \cs_gset_protected_nopar:Npx #1
1415     { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1416 }

(End definition for \ctex_ltj_save_alternate_family:Nnnn.)

```

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

```

1417 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1418 {
1419     \group_begin:
1420     \__ctex_ltj_change_encoding:
1421     \cs_set_eq:NN \CJKfamily \use_none:n
1422     \ctex_ltj_swap_cs:NN
1423     \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1424     \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1425     \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1426     \group_end:
1427 }
1428 \tl_new:N \l__ctex_ltj_alternate_family_tl

(End definition for \ctex_ltj_set_alternate_family:nnn.)

```

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

```

1429 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2

```

```

1430 {
1431   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1432   \cs_set_eq:NN #1 #2
1433   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1434   \cs_undefine:N \__ctex_ltj_tmp:w
1435 }

```

(End definition for \ctex_ltj_swap_cs:NN.)

LTJFONTUID
 __ctex_ltj_update_family_uid:N

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。

```

1436 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1437 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1438 {
1439   \int_gincr:N \g__ctex_ltj_family_int
1440   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1441 }
1442 \int_new:N \g__ctex_ltj_family_int

```

(End definition for LTJFONTUID and __ctex_ltj_update_family_uid:N.)

\ctex_ltj_declare_alter_nate_shape:nnnnnn

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

```

1443 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1444 {
1445   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1446   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1447   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1448   { \l_fontspec_family_tl } {#3} {#4}
1449 }

```

(End definition for \ctex_ltj_declare_alter_nate_shape:nnnnnn.)

\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。

```

1450 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1451 {
1452   \clist_map_inline:Nn #1
1453   {
1454     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1455     {
1456       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1457       \l__ctex_ltj_char_range_tl
1458     }
1459     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1460   }
1461   \__ctex_ltj_save_alter_nate_shape:cnn
1462   { \__ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1463   { luatexja.jfont.clear_alt_font_latex }
1464   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1465 }

```

(End definition for \ctex_ltj_set_alter_nate_shape:Nnnnnnn.)

\ctex_ltj_set_alter_nate_shape:nnn

我们使用 `->` 而不是像 **LuaTeX-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。

```

1466 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1467 { m m > { \SplitArgument { \c_one } { -> } } m }

```

```

1468 { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
1469 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
1470 {
1471   \ctex_ltj_set_alternate_shape:n
1472   {
1473     \IfNoValueTF {#4}
1474     { \int_eval:n {#3} , \int_eval:n {#3} , }
1475     {
1476       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1477       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1478     }
1479     '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1480     '\lua_escape_x:n { \CJK@encoding/#1 }'
1481   }
1482 }
1483 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1484 {
1485   \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1486   \__ctex_ltj_save_alternate_shape:cnn
1487   { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1488   { luatexja.jfont.set_alt_font_latex } {#1}
1489 }

```

(End definition for \ctex_ltj_set_alternate_shape:nnn.)

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

```

1490 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1491 {
1492   \tl_map_inline:Nn #3
1493   {
1494     \ctex_ltj_set_alternate_shape:n
1495     {
1496       ##1 ,
1497       '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1498       '\lua_escape_x:n { \CJK@encoding/#1 }'
1499     }
1500   }
1501 }

```

(End definition for \ctex_ltj_set_alternate_shape:nnN.)

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

```

1502 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1503 {
1504   \group_begin:
1505   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1506   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1507   \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1508   \cs_gset_protected_nopar:Npx #1
1509   { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1510   \group_end:
1511 }
1512 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nnn { c }

```

(End definition for __ctex_ltj_save_alternate_shape:Nnn.)

clearalternatefont 清除和重置操作总是全局的。

resetalternatefont

```

1513 \keys_define:nn { ctex }
1514 {
1515   clearalternatefont .code:n =
1516   { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,

```

```

1517   resetalternatefont .code:n =
1518   { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1519   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1520   resetalternatefont .default:n = \l_ctex_ltj_family_tl
1521 }
1522 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1523 {
1524   \group_begin:
1525   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1526   {
1527     \cs_if_exist_use:cT { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1528     {
1529       \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1530       {#1} { \l__ctex_ltj_base_family_tl }
1531       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1532       \selectfont
1533     }
1534   }
1535   { \__ctex_ltj_family_unknown_warning:n {#1} }
1536   \group_end:
1537 }
1538 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1539 {
1540   \group_begin:
1541   \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1542   {
1543     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1544     \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1545     \selectfont
1546   }
1547   \group_end:
1548 }
1549 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1550 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalternatefont` and `resetalternatefont`. These functions are documented on page 26.)

declarecharrange 预先声明字符范围。

```

1551 \keys_define:nn { ctex }
1552 {
1553   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1554   declarecharrange .value_required:n = true
1555 }
1556 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1557 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1558 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1559 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1560 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

(End definition for `declarecharrange`. This function is documented on page 25.)

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

```

1561 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1562 {
1563   \tl_clear:N \l__ctex_ltj_char_range_tl
1564   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1565   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1566   \ctex_ltj_def_char_range_key:n {#1}
1567   \tl_clear:N \l__ctex_ltj_char_range_tl
1568 }
1569 \tl_new:N \l__ctex_ltj_char_range_tl
1570 \prop_new:N \g__ctex_ltj_char_range_prop

```

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

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

```

1571 \NewDocumentCommand \ctex_ljt_save_char_range:n
1572 { > { \SplitArgument { \c_one } { -> } } m }
1573 { \ctex_ljt_save_char_range:nn #1 }
1574 \cs_new_protected_nopar:Npn \ctex_ljt_save_char_range:nn #1#2
1575 {
1576   \tl_put_right:Nx \l__ctex_ljt_char_range_tl
1577   { {
1578     \IfNoValueTF {#2}
1579     { \int_eval:n {#1} , \int_eval:n {#1} }
1580     {
1581       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1582       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1583     }
1584   } }
1585 }

(End definition for \ctex_ljt_save_char_range:n.)

```

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

```

1586 \cs_new_protected_nopar:Npn \ctex_ljt_def_char_range_key:n #1
1587 {
1588   \keys_if_exist:nnF { ctex_ljt / fontspec } {#1}
1589   {
1590     \keys_define:nn { ctex_ljt / fontspec }
1591     { #1 .code:n = \ctex_ljt_char_range_key:nn {#1} { ##1 } }
1592   }
1593 }

(End definition for \ctex_ljt_def_char_range_key:n.)

```

`\ctex_ljt_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。

```

1594 \cs_new_protected:Npn \ctex_ljt_char_range_key:nn #1#2
1595 {
1596   \tl_if_blank:nTF {#2}
1597   { \tl_set:Nn \l__ctex_ljt_char_range_clist {#1} }
1598   {
1599     \clist_if_empty:NT \l__ctex_ljt_char_range_clist
1600     {
1601       \tl_set:Nn \l__ctex_ljt_tmp_tl { {#1} }
1602       \__ctex_ljt_char_range_parse_feature:w #2 \q_stop
1603     }
1604   }
1605 }

(End definition for \ctex_ljt_char_range_key:nn.)

```

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

```

1606 \NewDocumentCommand \__ctex_ljt_char_range_parse_feature:w
1607 { +o o u { \q_stop } }
1608 {
1609   \exp_args:NNf \tl_put_right:Nn \l__ctex_ljt_tmp_tl
1610   {
1611     \IfNoValueTF {#1} { {#3} }
1612     {
1613       \IfNoValueTF {#2}
1614       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] { #3 } } }
1615       { [ { #1 } ] { [ #2 ] } }
1616     }
1617   }

```

```

1618 \seq_put_right:No \l__ctex_ltj_alterate_seq { \l__ctex_ltj_tmp_tl }
1619 }

```

(End definition for \l__ctex_ltj_char_range_parse_feature:w.)

14.4.3.7 其它设置

在抄录环境中禁用 autospacing 和 autoxspacing。然而, LuaTeX-ja 还是会使 JAchar 自动折行。没有看到有简单的禁用折行的办法, 可能需要设置所有的 JAchar 的 prebreakpenalty 或 postbreakpenalty 为 10000:

```

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

1620 \AtBeginDocument
1621 {
1622   \ctex_appto_cmd:NnnTF \verbatim@font
1623   { \char_set_catcode_letter:n { 64 } }
1624   { \CTEX@verbatim@font@hook }
1625   { }
1626   { \ctex_patch_failure:N \verbatim@font }
1627 }
1628 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1629 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1630 <@@=>

```

\@@italiccorr L^AT_EX 的倾斜校正也要重新定义。

```

1631 \cs_set_eq:NN \@@italiccorr \/

(End definition for \@@italiccorr.)

1632 <@@=ctex>
1633 </luatex>

```

14.4.4 ctex-engine-uptex.def

```

1634 <*uptex|aptex>

```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```

1635 \tl_set:Nn \CJKrmdefault { rm }
1636 \tl_set:Nn \CJKsfdefault { sf }
1637 \tl_set:Nn \CJKttdefault { tt }
1638 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1639 \tl_put_left:Nn \rmfamily { \kanjifamily { \CJKrmdefault } }
1640 \tl_put_left:Nn \sffamily { \kanjifamily { \CJKsfdefault } }
1641 \tl_put_left:Nn \ttfamily { \kanjifamily { \CJKttdefault } }
1642 \NewDocumentCommand \CJKfamily { m }
1643 {
1644   \kanjifamily { #1 }
1645   \selectfont
1646 }
1647 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```


在导言区末尾更新 \CJKfamilydefault。

```
1648 \ctex_at_end_preamble:n { \ctex_update_default_family: }
```

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

```
1649 \ctex_patch_cmd_once:NnnnTF \em
1650 { \ExplSyntaxOff }
1651 { \eminnershape \else \gtfamily \itshape }
1652 { \eminnershape \else \itshape }
1653 { }
1654 {
1655   \ctex_patch_cmd:Nnn \em
1656   { \mcfamily \upshape \else \gtfamily \itshape }
1657   { \eminnershape \else \itshape }
1658 }
1659 \cs_set_nopar:Npn \eminnershape { \upshape }
```

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

\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。

```
1660 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1661 {
1662   \DeclareKanjiFamily{JY2}{#1}{-}
1663   \DeclareKanjiFamily{JT2}{#1}{-}
1664   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{-}
1665   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{-}
1666   \tl_if_empty:nF { #3 }
1667   {
1668     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{-}
1669     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{-}
1670   }
1671 }
1672 \@onlypreamble \ctex_set_upfamily:nnn
```

(End definition for \ctex_set_upfamily:nnn.)

\ctex_set_upmap:nnn 设置 up \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。

```
1673 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1674 {
1675   \ctex_set_zhmap:n
1676   {
1677     \special{ pdf:mapline~ #1-h~ UniGB-UTF16-H~ #2 }
1678     \special{ pdf:mapline~ #1-v~ UniGB-UTF16-V~ #2 }
1679     \tl_if_empty:nF { #3 }
1680     {
1681       \special{ pdf:mapline~ #1b-h~ UniGB-UTF16-H~ #3 }
1682       \special{ pdf:mapline~ #1b-v~ UniGB-UTF16-V~ #3 }
1683     }
1684   }
1685 }
1686 \@onlypreamble \ctex_set_upmap:nnn
```

(End definition for \ctex_set_upmap:nnn.)

\ctex_set_upfonts:nnnnnn 设置 up \TeX 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。

```

1687 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1688 {
1689   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1690   \ctex_set_upmap:nnn { upserifit } { #3 } {}
1691   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1692   \ctex_set_upmap:nnn { upmono } { #6 } {}
1693 }
1694 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

(End definition for \ctex_set_upfonts:nnnnnn.)

\@EverySelectfont@Init **everyysel** 宏包 (2011/10/28) 未考虑 **upL^AT_EX** 对 \selectfont 的修改, 因此这里需要对其给出正确的定义。因为不会使用 **CJK**, 这里忽略有关 **CJK** 的定义。

```

1695 \cs_set_nopar:Npn \@EverySelectfont@Init
1696 {
1697   \ctex_patch_cmd:Nnn \selectfont
1698   { \enc@update }
1699   {
1700     \@EverySelectfont@EveryHook
1701     \@EverySelectfont@AtNextHook
1702     \gdef \@EverySelectfont@AtNextHook { }
1703     \enc@update
1704   }
1705   \ifpackageloaded { tracefmt }
1706   {
1707     \EverySelectfont
1708     {
1709       \ifnum \tracingfonts > \tw@
1710         \font@info { Switching~ to~ \font@name }
1711       \fi
1712     }
1713   }
1714   { }
1715   \let \@EverySelectfont@Init \undefined
1716 }

```

(End definition for \@EverySelectfont@Init.)

1717 </uptex|aptex>

14.4.5 调整 \CJKfamilydefault

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

```

1718 <*pdf|luatex|uptex|aptex>
1719 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1720 {
1721   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1722   {
1723     \group_begin:
1724     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1725     \tl_gset:Nx \CJKfamilydefault
1726     {
1727       \str_case:onF { \familydefault }
1728       {
1729         { \rmdefault } { \exp_not:N \CJKrmdefault }
1730         { \sfdefault } { \exp_not:N \CJKsfdefault }
1731         { \ttdefault } { \exp_not:N \CJKttdefault }
1732       }
1733       { \CJKfamilydefault }
1734     }
1735     \group_end:
1736   }

```

使用 Lua_{TEX} 时, 自动调整得到的 `\CJKfamilydefault` 可能没有定义, 需要确认它的存在性。使用 CJK 宏包或 up_{TEX} 时, `C19rm`、`JY2rm` 等总是有定义的, 不需要确认。

```
1737 <*luatex>
1738     \ctex_ltj_ensure_default_family:
1739 </luatex>
1740 }
```

(End definition for `\ctex_update_default_family:.`)

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

```
1741 \tl_new:N \l__ctex_family_default_init_tl
1742 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1743 \tl_set:Nx \l__ctex_family_default_init_tl
1744 {
1745     \exp_not:N \__ctex_family_default_wrap:n
1746     { \exp_not:o { \CJKfamilydefault } }
1747 }
1748 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1749 </pdf|luatex|uptex|aptex>
```

(End definition for `\l__ctex_family_default_init_tl`.)

14.4.6 操作系统的判断

`\ctex_detected_platform:` 在 Lua_{TEX} 下直接用调用 `os.name` 来判断。

```
1750 <*luatex>
1751 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1752 {
1753     \tl_gset:Nx \g__ctex_fontset_tl
1754     {
1755         \lua_now_x:n
1756         {
1757             if ~ os.name == 'windows' then ~
1758                 tex.sprint ( 'windows' )
1759             elseif ~ os.name == 'macosx' then ~
1760                 tex.sprint ( 'mac' )
1761             else ~
1762                 tex.sprint ( 'fandol' )
1763             end
1764         }
1765     }
1766 }
1767 </luatex>
```

Ap_{TEX} 可以使用 `\ngostype` 来判断。

```
1768 <*aptex>
1769 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1770 {
1771     \tl_gset:Nx \g__ctex_fontset_tl
1772     {
1773         \str_case:onF { \ngostype }
1774         {
1775             { Win32 } { windows }
1776             { Win64 } { windows }
1777             { Darwin } { mac }
1778         }
1779         { fandol }
1780     }
1781 }
1782 </aptex>
```

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

```

1783 <*xetex|pdfTeX|uptex>
1784 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1785 {
1786   \file_if_exist:nTF { /dev/null }
1787   {
1788     \file_if_exist:nTF { nul: }
1789     {
1790       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1791       { \ctex_if_macosx:TF { mac } { windows } }
1792       { \ctex_if_macosx:TF { mac } { fandol } }
1793     }
1794     { \ctex_if_macosx:TF { mac } { fandol } }
1795   }
1796   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1797 }
1798 \str_const:Nx \c__ctex_upper_case_file_str
1799 { \exp_args:No \str_upper_case:n { \g_file_current_name_tl } }

(End definition for \ctex_detected_platform:.)

```

`\ctex_if_macosx:TF` 以 `/Library/Fonts/Songti.ttc` 为特征文件判断 Mac OS X。

```

1800 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1801 {
1802   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1803   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1804   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1805 }
1806 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1807 </xetex|pdfTeX|uptex>

(End definition for \ctex_if_macosx:TF.)

```

14.4.7 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。

```

1808 <*class|ctex>
1809 \@ifpackageloaded { hyperref }
1810 {
1811   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1812   { \hypersetup {#1} }
1813 }
1814 {
1815   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1816   { \PassOptionsToPackage {#1} { hyperref } }
1817 }
1818 </class|ctex>

(End definition for \ctex_hypersetup:n.)

```

在 pdfTeX 下使用 GBK 编码, DVIPDFMx 驱动可以直接用它的 `\special` 命令, 其它模式用 `xCJK2uni` 宏包处理。使用 UTF-8 编码时, `CJKutf8` 已经处理了书签问题, 但仍需要设置

pdfencoding 为 unicode, 目的是在书签的开头写入 BOM (\376\377), 提示这是 UTF-16BE 字节流。

```

1819 <*pdfTeX>
1820 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1821 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1822 {
1823   \ctex_hypersetup:n { CJKbookmarks = true }
1824   \sys_if_output_pdf:TF
1825     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1826     {
1827       \ctex_at_end_package:nn { hyperref }
1828       {
1829         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1830         {
1831           \AtBeginShipoutFirst
1832             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1833         }
1834         { \RequirePackage { xCJK2uni } }
1835       }
1836     }
1837 }
1838 { \ctex_hypersetup:n { pdfencoding = unicode } }
1839 </pdfTeX>

```

在 X_YTeX 下,hyperref 在处理带有非 ASCII 字符和 \% 的书签时有问题¹⁷。事实上,hyperref 在驱动文件 h_xetex.def 中设置了 \Hy@unicodetrue,从而书签总是会被 \HyPsd@ConvertToUnicode 转化成 UTF-16BE 编码的形式(抄录自 \pdfstringdef 的定义):

```

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

```

通过宏包选项 pdfencoding=unicode 设置 \HyPsd@pdfencoding 为 unicode,可以避免随后再将书签从 UTF-16BE 字节流转化回正常字符(其中使用的 \HyPsd@ToBigChars 没有考虑书签中含有 \% 的情况)。Heiko Oberdiek 在 README 中说明了将书签转化回正常字符的意图:避免 X_DVIPDFMX 的警告¹⁸:

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

X_YTeX 的维护者 Khaled Hosny 已经注意到了这个问题¹⁹。需要注意的是,h_xetex.def 重载了宏包选项 unicode,目的是不能设置它为 false,但也导致它不会改变 \HyPsd@pdfencoding。如果 hyperref 先于 C_TEX 被载入,那么 unicode 选项是没有意义的。因此要通过意义相同但在 X_YTeX 下更保险的 pdfencoding 选项来设置。为了与 X_YTeX 下的行为一致(使用 \HyPsd@LoadUnicode 载入 puenc.def),在 Lua_TEX 下也启用这个选项。

```
1840 <*xetex|luatex>
```

¹⁷<https://github.com/CTeX-org/ctex-kit/issues/39>

¹⁸<http://project.ktug.org/dvipdfmx/mailman/dvipdfmx/2009-December/000153.html>

¹⁹<http://tug.org/pipermail/tex-live/2013-December/034613.html>

```

1841 \ctex_hypersetup:n { pdfencoding = unicode }
1842 </xetex| luatex>

```

我们假定 $\text{upT}_{\text{E}}\text{X}$ 使用 DVIPDFM_x 驱动输出, 于是使用与 $\text{pdfT}_{\text{E}}\text{X}$ 类似的设置。注意 $\text{upT}_{\text{E}}\text{X}$ 需要使用 UTF8-UTF16 的编码转换。

```

1843 <*uptex| aptex>
1844 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1845 \ctex_at_end_package:nn { hyperref }
1846 {
1847   \AtBeginShipoutFirst{ \special{ pdf:tounicode~ UTF8-UTF16 } }
1848 }
1849 </uptex| aptex>

1850 <*pdfTeX|xetex| luatex| uptex| aptex>

```

14.4.8 CJKfntef、xeCJKfntef 相关设置

$\text{CT}_{\text{E}}\text{X}$ 宏集对 $\text{pdfT}_{\text{E}}\text{X}$ 与 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 引擎, 分别载入 CJKfntef 或 xeCJKfntef 宏包, 并关闭宏包默认的彩色等多余格式。

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

```

1851 <*pdfTeX>
1852 \RequirePackage { CJKfntef }
1853 \normalem
1854 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1855 { \tl_clear:c { CJK#1color } }
1856 </pdfTeX>
1857 <*xetex>
1858 \RequirePackage { xeCJKfntef }
1859 \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
1860 {
1861   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1862   { \xeCJKsetup { #1 / format = { } } }
1863 }
1864 {
1865   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1866   { \tl_clear:c { CJK#1color } }
1867 }
1868 </xetex>
1869 <*luatex| uptex| aptex>
1870 \msg_new:nnn { ctex } { fntef-not-available }
1871 <luatex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1872 <uptex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
1873 <aptex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1874 \msg_warning:nn { ctex } { fntef-not-available }
1875 </luatex| uptex| aptex>
1876 \clist_map_inline:nn
1877 { underdot , underline , underdblline , underwave , sout , xout }
1878 <*pdfTeX|xetex>
1879 {
1880   \__ctex_clear_fntef_color:n {#1}
1881   \cs_new_protected_nopar:cpx { CTEX#1 }
1882   {
1883     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1884     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N CJK~ instead. }
1885     \exp_not:c { CJK#1 }
1886   }
1887 }
1888 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1889 {
1890   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1891   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1892   \CJKfilltwosides

```

```

1893 }
1894 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1895 </pdfTeX|xetex>
1896 <*/LaTeX|uptex|aptex>
1897 { \cs_new_eq:cN { CTEX#1 } \use:n }
1898 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1899 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1900 </LaTeX|uptex|aptex>
1901 <*/pdfTeX>
1902 \clist_map_inline:nn
1903 {
1904     underdotbasesep , underdotsep , underlinebasesep ,
1905     underlinesep , underdbllinesep , underdbllinebasesep ,
1906     underwavebasesep , underwavesep , southeight ,
1907     underdotcolor , underwavecolor , underlinecolor ,
1908     underdbllinecolor , soutcolor , xoutcolor
1909 }
1910 {
1911     \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1912     \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1913 }
1914 </pdfTeX>

```

14.4.9 \ccwd 的更新

```

\ctex_update_ccwd:
  \ccwd
1915 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1916 <*/pdfTeX|xetex>
1917 {
1918     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1919     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1920 }
1921 </pdfTeX|xetex>
1922 <*/LaTeX>
1923 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1924 </LaTeX>
1925 <*/uptex|aptex>
1926 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1927 </uptex|aptex>
1928 \dim_new:N \ccwd

```

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

\ctex_update_ccglue: 更新字间距。

```

1929 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1930 <*/pdfTeX>
1931 {
1932     \cs_set_protected_nopar:Npn \CJKglue
1933     { \skip_horizontal:N \l__ctex_ccglue_skip }
1934 }
1935 </pdfTeX>
1936 <*/xetex>
1937 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1938 </xetex>
1939 <*/LaTeX>
1940 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1941 </LaTeX>
1942 <*/uptex|aptex>
1943 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1944 </uptex|aptex>
1945 \skip_new:N \l__ctex_ccglue_skip

```

(End definition for \ctex_update_ccglue:.)

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

```
\ctex_if_ccglue_touched:TF
1946 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1947 {
1948   <*<pdf<pdf>tex|<xetex>
1949   \if_meaning:w \CJKglue \__ctex_ccglue:
1950   \prg_return_false: \else: \prg_return_true: \fi:
1951   </pdf>tex|<xetex>
1952   <*<lua<lua>tex>
1953   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1954   { \prg_return_false: } { \prg_return_true: }
1955   </lua>tex>
1956   <*<uptex|aptex>
1957   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1958   { \prg_return_false: } { \prg_return_true: }
1959   </uptex|aptex>
1960 }
```

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

```
1961 <*<pdf<pdf>tex|<xetex>
1962 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1963 </pdf>tex|<xetex>
```

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

`\ctex_update_em_unit:` 将当前汉字的宽度保存到 `\ccwd` 中备用。不采用 `1em`, 因为这时的 `1em` 实际上来自西文字体的信息, 未必等于汉字的宽度, 这似乎在传统的 `.tfm` 字体上表现更明显。在 `pdfTeX` 和 `XTeX` 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 `XTeX` 可以用 `\fontcharwd` 来改进。而在 `pdfTeX` 下, 若使用 `zhmetrics` 技术, 所有的汉字共享同一个 `.tfm`, `\fontcharwd` 也就没有意义。在 `LuaTeX` 下, `LuaTeX-jā` 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 `\zw` 作为汉字宽度。 `upTeX` 可以直接使用原生的长度单位 `zw`。

```
1964 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1965 <*<pdf<pdf>tex|<xetex> { \dim_set:Nn \ccwd { \f@size \p@ } }
1966 <*<lua<lua>tex> { \dim_set:Nn \ccwd { \zw } }
1967 <*<uptex|aptex> { \dim_set:Nn \ccwd { 1zw } }
```

(End definition for `\ctex_update_em_unit:.`)

14.4.10 其它

`\ctex_add_to_selectfont:n` `\CTEX@selectfont@hook` `\EverySelectfont` 直到文档开始时才有效。为了 `\ccwd` 和 `LuaTeX-jā` 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。 `everysel` 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。

```
1968 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1969 {
1970   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1971   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1972 }
1973 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1974 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1975 \ctex_preto_cmd:NnnTF \selectfont { \ExplSyntaxOff }
1976 { \CTEX@selectfont@hook }
1977 {
1978   \tl_put_left:Nn \@EverySelectfont@Init
1979   { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1980 }
1981 { \ctex_patch_failure:N \selectfont }
```


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

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

```

1982 <*pdfTeX>
1983 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1984 </pdfTeX>
1985 <*xetex|luatex|uptex|aptex>
1986 \EverySelectfont { \CTEX@selectfont@hook }
1987 </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。

```

1988 <*luatex>
1989 \ctex_add_to_selectfont:n
1990 {
1991   \ctex_ltj_select_font:
1992   \ctex_ltj_select_alternate_font:
1993 }
1994 \tl_set:Nn \CJK@family { song } \selectfont
1995 \tl_clear:N \CJK@family
1996 </luatex>

```

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

```

1997 <*luatex|uptex|aptex>
1998 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
1999 {
2000   \skip_if_eq:nnT
2001   <luatex> { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
2002   <uptex|aptex> { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2003   {
2004     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
2005     <luatex> \ltjsetxkanjiskip \l__ctex_xkanjiskip_skip
2006     <uptex|aptex> \skip_set_eq:NN \ptex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2007   }
2008 }
2009 \tl_new:N \l__ctex_xkanjiskip_tl
2010 \tl_set:Nn \l__ctex_xkanjiskip_tl
2011 <luatex> { .25\zw plus 1pt minus 1pt }
2012 <uptex|aptex> { .25zw plus 1pt minus 1pt }
2013 \skip_new:N \l__ctex_xkanjiskip_skip
2014 \skip_set:Nn \l__ctex_xkanjiskip_skip
2015 <luatex> { \ltjgetparameter { xkanjiskip } }
2016 <uptex|aptex> { \ptex_xkanjiskip:D }

```

(End definition for `\ctex_update_xkanjiskip:` and `\l__ctex_xkanjiskip_skip`.)

```

2017 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2018 </luatex|uptex|aptex>

```

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

```

2019 \keys_define:nn { ctex }
2020 {
2021   <*pdfTeX|xetex>
2022   space .choice: ,
2023   space / true .code:n =
2024   <pdfTeX> { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
2025   <xetex> { \xeCJKsetup { CJKspace = true } } ,
2026   space / auto .code:n =

```

```

2027 <pdfTeX>      { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
2028 <xetex>        { \xeCJKsetup { CJKspace = false } } ,
2029     space / false .code:n =
2030 <pdfTeX>      { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
2031 <xetex>        { \xeCJKsetup { CJKspace = false } } ,
2032     space .default:n = { true } ,
2033     space .initial:n = { auto }
2034 </pdfTeX|xetex>
2035 <*luatex|uptex|aptex>
2036     space .code:n =
2037     { \msg_warning:nn { ctex } { invalid-option } }
2038 </luatex|uptex|aptex>
2039 }

```

(End definition for space. This function is documented on page 28.)

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-jā 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。

```

2040 \keys_define:nn { ctex }
2041 {
2042     punct .code:n =
2043     {
2044         \tl_set:Nx \l__ctex_punct_tl { #1 }
2045 <pdfTeX>      \punctstyle { \l__ctex_punct_tl }
2046 <xetex>      \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2047 <luatex>      \ctex_mono_jfm:o { \l__ctex_punct_tl }
2048 <uptex|aptex> \msg_warning:nn { ctex } { invalid-option }
2049     } ,
2050     punct .default:n = { quanjiao } ,
2051 }

```

(End definition for punct. This function is documented on page 28.)

X_YLaTeX、LuaLaTeX 和 upLaTeX 总是使用 UTF8 编码。

```

2052 <*xetex|luatex|uptex|aptex>
2053 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2054 </xetex|luatex|uptex|aptex>

2055 </pdfTeX|xetex|luatex|uptex|aptex>
2056 <*class|ctex>

```

14.4.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

2057 \ctex_file_input:n { \c__ctex_engine_file_str }

```

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。

```

2058 \keys_define:nn { ctex }
2059 {
2060     autoindent .choice: ,
2061     autoindent .default:n = { true } ,
2062     autoindent / true .code:n =
2063     {
2064         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2065         \ctex_select_size:
2066     } ,
2067     autoindent / false .code:n =
2068     { \tl_clear:N \l__ctex_autoindent_tl } ,
2069     autoindent / unknown .code:n =

```

```

2070     {
2071         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2072         \ctex_select_size:
2073     }
2074 }

```

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

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2075 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2076 \cs_new_protected_nopar:Npn \ctex_select_size:
2077 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

(End definition for \CTEXsetfont. This function is documented on page 29.)

\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。

```

2078 \cs_new_protected_nopar:Npn \ctex_update_size:
2079 {
2080     \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2081     {
2082         \ctex_update_stretch:
2083         \ctex_update_parindent:
2084     }
2085     { \ctex_update_ziju: }
2086 }
2087 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2088 \tl_new:N \l__ctex_ziju_tl
2089 \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。

```

2090 \ctex_add_to_selectfont:n
2091 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

(End definition for \ctex_update_size:.)

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

```

2092 \keys_define:nn { ctex }
2093 {
2094     linestretch .code:n =
2095     {
2096         \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2097         \ctex_select_size:
2098     } ,
2099     linestretch .value_required:n = true
2100 }
2101 \tl_new:N \l__ctex_line_stretch_tl
2102 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

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

\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 everyysel 的钩子位于 \size@update 之前, \baselineskip 还未更新, 不能直接使用它。

```

2103 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2104 {
2105   \ctex_update_em_unit:
2106   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2107   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2108   {
2109     \ctex_if_ccglue_touched:TF
2110     { \ctex_update_ccwd: }
2111     {
2112       \dim_set:Nn \l__ctex_tmp_dim
2113       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2114       \skip_set:Nn \l__ctex_ccglue_skip
2115       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2116       \ctex_update_ccglue:
2117     }
2118   }
2119   {
2120     \int_set:Nn \l__ctex_tmp_int
2121     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2122     \tex_divide:D \l__ctex_tmp_int \ccwd
2123     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2124     {
2125       \skip_set:Nn \l__ctex_ccglue_skip
2126       {
2127         \c_zero_dim plus \dim_eval:n
2128         {
2129           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2130           \l__ctex_tmp_int
2131         }
2132       }
2133     }
2134     { \skip_zero:N \l__ctex_ccglue_skip }
2135     \ctex_update_ccglue:
2136   }
2137 }

```

(End definition for \ctex_update_stretch:.)

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

```

2138 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2139 {
2140   \tl_if_empty:NF \l__ctex_autoindent_tl
2141   {
2142     \dim_compare:nNnF \parindent = \c_zero_dim
2143     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2144   }
2145 }

```

(End definition for \ctex_update_parindent:.)

\ziju 若参数为 0, 则恢复正常间距。

```

2146 \NewDocumentCommand \ziju { m }
2147 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2148 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2149 {
2150   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2151   \ctex_select_size:
2152 }

```

(End definition for \ziju. This function is documented on page 24.)

\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。

```

2153 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2154 {
2155   \ctex_update_em_unit:
2156   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2157   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2158   \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。

```

2159   {
2160     \dim_set:Nn \l__ctex_tmp_dim
2161       { \linewidth - \ccwd + \l__ctex_ziju_dim }
2162     \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2163     \tex_divide:D \l__ctex_tmp_int \ccwd
2164     \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。

```

2165     \dim_compare:nNnF \parindent = \c_zero_dim
2166     {
2167       \int_compare:nNnF \l__ctex_tmp_int < \c_three
2168       { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2169     }
2170     \skip_set:Nn \l__ctex_ccglue_skip
2171     {
2172       \l__ctex_ziju_dim
2173       plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2174       minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2175       { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2176     }
2177   }
2178   { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2179   \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 $2\backslash\mathrm{ccwd}$ 。

```

2180   \ctex_update_parindent:
2181 }
2182 \dim_new:N \l__ctex_ziju_dim

```

(End definition for `\ctex_update_ziju:`)

`\CTEXindent` 过时命令。
`\CTEXnoindent`

```

2183 \NewDocumentCommand \CTEXindent { }
2184 {
2185   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2186   { \parindent is~ set~ to~ 2\ccwd. }
2187   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2188 }
2189 \NewDocumentCommand \CTEXnoindent { }
2190 {
2191   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2192   { \parindent is~ set~ to~ 0pt. }
2193   \dim_zero:N \parindent
2194 }

```

(End definition for `\CTEXindent` and `\CTEXnoindent`. These functions are documented on page 29.)

14.6 中文数字与日期

```

2195 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2196 \RequirePackage { zhnumber }

```

\chinese

```

2197 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2198 \cs_new_eq:NN \@chinese \@zhnum
2199 \cs_new_eq:NN \Chinese \chinese
2200 \cs_new_eq:NN \CTEXcounter \use_none:n

```

(End definition for \chinese. This function is documented on page 24.)

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

```

2201 \ctex_at_end_package:nn { enumitem }
2202 {
2203   \cs_if_free:NF \AddEnumerateCounter
2204   {
2205     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2206     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2207     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2208   }
2209 }

```

\CTEXnumber

\CTEXdigits

```

2210 \NewDocumentCommand \CTEXnumber { m m }
2211 { \protected@edef #1 { \zhnumber {#2} } }
2212 \NewDocumentCommand \CTEXdigits { m m }
2213 { \protected@edef #1 { \zhdigits {#2} } }

```

(End definition for \CTEXnumber and \CTEXdigits. These functions are documented on page 24.)

today

```

2214 \cs_set_eq:NN \CTEX@todayold \today
2215 \keys_define:nn { ctex }
2216 {
2217   today .choice: ,
2218   today / old .code:n =
2219     { \cs_set_eq:NN \today \CTEX@todayold } ,
2220   today / small .code:n =
2221     {
2222       \cs_set_eq:NN \today \zhtoday
2223       \zhnumsetup { time = Arabic }
2224     } ,
2225   today / big .code:n =
2226     {
2227       \cs_set_eq:NN \today \zhtoday
2228       \zhnumsetup { time = Chinese }
2229     } ,
2230   today / unknown .code:n =
2231     { \msg_error:nnx { ctex } { today-undef } {#1} }
2232 }
2233 \msg_new:nnnn { ctex } { today-undef }
2234 { Today~format~`#1'~is~undefined. }
2235 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

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

14.7 其它中文标题定义

```

2236 \keys_define:nn { ctex }
2237 {
2238   contentsname .tl_set:N = \contentsname ,
2239   listfigurename .tl_set:N = \listfigurename ,

```

```

2240 listtablename .tl_set:N = \listtablename ,
2241 figurename .tl_set:N = \figurename ,
2242 tablename .tl_set:N = \tablename ,
2243 abstractname .tl_set:N = \abstractname ,
2244 indexname .tl_set:N = \indexname ,
2245 appendixname .tl_set:N = \appendixname ,
2246 proofname .tl_set:N = \proofname ,
2247 <article> bibname .tl_set:N = \refname
2248 <book|report> bibname .tl_set:N = \bibname
2249 <*beamer>
2250 algorithmname .tl_set:N = \algorithmname ,
2251 bibname .tl_set:N = \bibname ,
2252 refname .tl_set:N = \refname ,
2253 continuation .tl_set:N = \insertcontinuationtext
2254 </beamer>
2255 }
2256 <*ctex>
2257 \msg_new:nnn { ctex } { ctexbibname }
2258 {
2259   Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2260   The~key~\bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2261 }
2262 \tl_if_exist:NTF \insertcontinuationtext
2263 {
2264   \keys_define:nn { ctex }
2265   {
2266     algorithmname .tl_set:N = \algorithmname ,
2267     bibname .tl_set:N = \bibname ,
2268     refname .tl_set:N = \refname ,
2269     continuation .tl_set:N = \insertcontinuationtext
2270   }
2271 }
2272 {
2273   \tl_if_exist:NTF \bibname
2274   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2275   {
2276     \tl_if_exist:NTF \refname
2277     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2278     {
2279       \msg_warning:nn { ctex } { ctexbibname }
2280       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2281     }
2282   }
2283 }
2284 </ctex>
2285 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 CT_EX 文档类或打开 heading 选项下生效。

```
2286 <*class|heading>
```

14.8.1 定义标题格式选项

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

```

2287 <*article|book|report>
2288 \seq_new:N \c__ctex_section_headings_seq
2289 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2290 { section , subsection , subsubsection , paragraph , subparagraph }
2291 </article|book|report>

```

(End definition for \c__ctex_section_headings_seq.)

\c__ctex_headings_seq

```

2292 \seq_new:N \c__ctex_headings_seq
2293 <*article|book|report>
2294 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
2295 <book|report>\seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2296 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2297 </article|book|report>
2298 <*beamer>
2299 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2300 { part , section , subsection }
2301 </beamer>

(End definition for \c__ctex_headings_seq.)

```

__ctex_initial_heading:n

```

2302 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2303 {
2304   \tl_new:c { CTEX@pre#1 }
2305   \tl_new:c { CTEX@post#1 }
2306   \tl_const:cx { CTEX@the#1 }
2307   {
2308     \exp_not:c { CTEX@pre#1 }
2309     \exp_not:c { CTEX@the#1 }
2310     \exp_not:c { CTEX@post#1 }
2311   }
2312   \tl_const:cx { CTEX@#1name }
2313   {
2314     \group_begin:
2315     \exp_not:c { CTEX@#1@nameformat }
2316     {
2317       \exp_not:c { CTEX@pre#1 }
2318       \exp_not:N \tl_if_empty:NTF
2319       \exp_not:c { CTEX@#1@numberformat }
2320       { \exp_not:c { CTEX@the#1 } }
2321       {
2322         \group_begin:
2323         \exp_not:c { CTEX@#1@numberformat }
2324         \exp_not:c { CTEX@the#1 }
2325         \group_end:
2326       }
2327       \exp_not:c { CTEX@post#1 }
2328     }
2329   \group_end:
2330 }
2331 }

```

(End definition for __ctex_initial_heading:n.)

__ctex_def_heading_keys:n

```

2332 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2333 {
2334   \tl_put_right:Nx \l__ctex_tmp_tl
2335   {
2336     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2337     #1 / name .code:n =
2338     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2339     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2340     #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2341     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2342     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2343     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2344     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2345     #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2346     #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1@beforekip } ,

```



```

2347      #1 / afterskip      .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2348      #1 / indent        .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2349      #1 / numbering     .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2350      < *article|book|report >
2351      #1 / afterindent    .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2352      #1 / runin          .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2353      < /article|book|report >
2354      #1 / format+        .code:n =
2355      { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2356      #1 / nameformat+    .code:n =
2357      { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2358      #1 / numberformat+ .code:n =
2359      { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2360      #1 / aftername+     .code:n =
2361      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2362      #1 / titleformat+   .code:n =
2363      { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2364      #1 / aftertitle+    .code:n =
2365      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2366      #1 / numbering      .initial:n = true ,
2367      #1 / beforeskip      .initial:n = \c_zero_skip ,
2368      #1 / afterskip       .initial:n = \c_zero_skip ,
2369      #1 / indent         .initial:n = \c_zero_dim ,
2370      #1 / beforeskip      .value_required:n = true ,
2371      #1 / afterskip       .value_required:n = true ,
2372      #1 / indent         .value_required:n = true ,
2373    }
2374  }

```

(End definition for __ctex_def_heading_keys:n.)

`ctex_assign_heading_name:nn` `__ctex_assign_heading_name:nnn` `name` 的值是一个至多两个元素的逗号分隔列表。由于 \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。

```

2375 \NewDocumentCommand \ctex_assign_heading_name:nn
2376 { m > { \SplitArgument { \c_one } { , } } +m }
2377 { \__ctex_assign_heading_name:nnn {#1} #2 }
2378 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2379 {
2380   \tl_set:cn { CTEX@pre#1 } {#2}
2381   \IfNoValueTF {#3}
2382   { \tl_clear:c { CTEX@post#1 } }
2383   { \tl_set:cn { CTEX@post#1 } {#3} }
2384 }

```

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

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

```

2385 \group_begin:
2386 < *book|report >
2387 \tl_set:Nn \l__ctex_tmp_tl
2388 {
2389   part / pagestyle      .tl_set:N = \CTEX@part@pagestyle ,
2390   chapter / pagestyle   .tl_set:N = \CTEX@chapter@pagestyle ,
2391   part / fixbeforeskip .bool_set:N = \CTEX@part@fixbeforeskip ,
2392   chapter / fixbeforeskip .bool_set:N = \CTEX@chapter@fixbeforeskip ,
2393 }
2394 < /book|report >
2395 < *article|beamer >
2396 \tl_clear:N \l__ctex_tmp_tl
2397 < /article|beamer >

```

(End definition for `part/pagestyle` and others. These functions are documented on page 21.)

定义标题键值选项。

```

2398 \seq_map_inline:Nn \c__ctex_headings_seq
2399 {
2400   \__ctex_initial_heading:n {#1}
2401   \__ctex_def_heading_keys:n {#1}
2402 }
2403 \use:x
2404 {
2405   \group_end:
2406   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2407 }
2408 <*article|book|report>

```

14.8.2 标准标题命令的修改

`\CTEX@makeheadinganchor` 用于 `numbering=false` 时, 设置超链接跳转描点。

```

2409 \cs_new_protected_nopar:Npn \CTEX@makeheadinganchor { }
2410 \ctex_at_end_package:nn { hyperref }
2411 { \cs_gset_protected_nopar:Npn \CTEX@makeheadinganchor { \phantomsection } }

(End definition for \CTEX@makeheadinganchor.)

```

`\CTEX@fixbeforeskip` 修正 `book` 和 `report` 类的 `\part` 和 `\chapter` 标题之前的多余空行。事实上, 减掉 `\parskip` 有一定的风险。如果接下来的内容不会进入水平模式(例如在 `format` 选项中使用 `\hrule` 或者 `\hbox`), $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 就不会加上 `\parskip`。这时候就需要用户把 `\parskip` 加到 `beforeskip` 上作为修正。

```

2412 <*book|report>
2413 \cs_new_protected_nopar:Npn \CTEX@fixbeforeskip
2414 {
2415   \par
2416   \ifdim \pagegoal = \maxdimen
2417     \advance \@tempskipa -\topskip
2418     \advance \@tempskipa -\parskip
2419     \nointerlineskip
2420   \fi
2421 }
2422 </book|report>

(End definition for \CTEX@fixbeforeskip.)

```

14.8.2.1 part 的标题

```

2423 <@@=>

```

`\part`

```

2424 <*article>
2425 \renewcommand\part{%
2426   \if@noskipsec \leavevmode \fi
2427   \par
2428   % \addvspace{4ex}%
2429   \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
2430   \ifodd \CTEX@part@afterindent
2431     \@afterindenttrue
2432   \else
2433     \@afterindentfalse
2434   \fi
2435   \secdef\@part\@spart}
2436 </article>
2437 <*book|report>
2438 \renewcommand\part{%

```

```

2439 \if@openright
2440 \cleardoublepage
2441 \else
2442 \clearpage
2443 \fi
2444 % \thispagestyle{plain}%
2445 \thispagestyle{\CTEX@part@pagestyle}%
2446 \if@twocolumn
2447 \onecolumn
2448 \@tempwattrue
2449 \else
2450 \@tempwafalse
2451 \fi
2452 % \null\vfil
2453 \@tempskipa = \glueexpr \CTEX@part@beforeskip \relax
2454 \ifodd \CTEX@part@fixbeforeskip
2455 \CTEX@fixbeforeskip
2456 \fi
2457 \vspace*{\@tempskipa}%
2458 \secdef\@part\@spart}
2459 </book|report>

```

(End definition for \part.)

\@part

```

2460 <*article>
2461 \def\@part[#1]#2{%
2462 \ifnum \c@secnumdepth >\m@ne
2463 \ifodd \CTEX@part@numbering
2464 \refstepcounter{part}%
2465 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2466 \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2467 \else
2468 \CTEX@makeheadinganchor
2469 \addcontentsline{toc}{part}{#1}%
2470 \fi
2471 \else
2472 \addcontentsline{toc}{part}{#1}%
2473 \fi
2474 {\interlinepenalty \@M
2475 % \normalfont \parindent \z@ \raggedright
2476 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2477 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2478 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2479 \CTEX@partname \CTEX@part@aftername
2480 \fi \fi
2481 % \huge\bfseries #2%
2482 \CTEX@part@titleformat{#2}%
2483 \markboth{}{}%
2484 \CTEX@part@aftertitle}%
2485 \nobreak
2486 % \vskip 3ex
2487 \vskip \glueexpr \CTEX@part@afterskip \relax
2488 \@afterheading}
2489 </article>
2490 <*book|report>
2491 \def\@part[#1]#2{%
2492 \ifnum \c@secnumdepth >-2\relax
2493 \ifodd \CTEX@part@numbering
2494 \refstepcounter{part}%
2495 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2496 \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2497 \else
2498 \CTEX@makeheadinganchor
2499 \addcontentsline{toc}{part}{#1}%
2500 \fi

```

```

2501 \else
2502   \addcontentsline{toc}{part}{#1}%
2503 \fi
2504 \markboth{}{}%
2505 {\interlinepenalty \@M
2506 % \normalfont \centering
2507 \normalfont \CTEX@part@format
2508 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2509 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2510 \CTEX@partname \CTEX@part@aftername
2511 \fi \fi
2512 % \Huge\bfseries #2\par}%
2513 \CTEX@part@titleformat{#2}%
2514 \CTEX@part@aftertitle}%
2515 \@endpart}
2516 </book|report>

```

(End definition for \@part.)

\@spart

```

2517 <*article>
2518 \def\@spart#1{%
2519   {\interlinepenalty \@M
2520   % \normalfont \parindent \z@ \raggedright
2521   \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2522   \CTEX@part@format
2523   % \huge \bfseries #1\par}%
2524   \CTEX@part@titleformat{#1}%
2525   \CTEX@part@aftertitle}%
2526   \nobreak
2527   % \vskip 3ex
2528   \vskip \glueexpr \CTEX@part@afterskip \relax
2529   \@afterheading}
2530 </article>
2531 <*book|report>
2532 \def\@spart#1{%
2533   {\interlinepenalty \@M
2534   % \normalfont \centering
2535   \normalfont \CTEX@part@format
2536   % \Huge \bfseries #1\par}%
2537   \CTEX@part@titleformat{#1}%
2538   \CTEX@part@aftertitle}%
2539   \@endpart}
2540 </book|report>

```

(End definition for \@spart.)

\@endpart

```

2541 <*book|report>
2542 \def\@endpart{%
2543   % \vfil
2544   \vskip \glueexpr \CTEX@part@afterskip \relax
2545   \newpage
2546   \if@twoside
2547   \if@openright
2548   \null
2549   \thispagestyle{empty}%
2550   \newpage
2551   \fi
2552   \fi
2553   \if@tempwa
2554   \twocolumn
2555   \fi}
2556 </book|report>

```

(End definition for \@endpart.)

14.8.2.2 chapter 的标题

2557 <*book|report>

\chapter

```

2558 \renewcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
2559 % \thispagestyle{plain}%
2560 % \thispagestyle{\CTEX@chapter@pagestyle}%
2561 % \global\@topnum\z@
2562 % \@afterindentfalse
2563 % \ifodd \CTEX@chapter@afterindent
2564 % \@afterindenttrue
2565 % \else
2566 % \@afterindentfalse
2567 % \fi
2568 \secdef\@chapter\@schapter}

```

(End definition for \chapter.)

\@chapter

```

2569 \def\@chapter[#1]#2{%
2570   \ifnum \c@secnumdepth >\m@ne
2571   <book> \if@mainmatter
2572     \ifodd \CTEX@chapter@numbering
2573       \refstepcounter{chapter}%
2574       % \typeout{\@chapapp\space\thechapter.}%
2575       % \typeout{\CTEXthechapter}%
2576       \addcontentsline{toc}{chapter}
2577       % {\protect\numberline{\thechapter}#1}%
2578       {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2579     \else
2580       \CTEX@makeheadinganchor
2581       \addcontentsline{toc}{chapter}{#1}%
2582     \fi
2583   <book> \else
2584   <book> \addcontentsline{toc}{chapter}{#1}%
2585   <book> \fi
2586   \else
2587     \addcontentsline{toc}{chapter}{#1}%
2588   \fi
2589   \chaptermark{#1}%
2590   \addtocontents{lof}{\protect\addvspace{10\p@}}%
2591   \addtocontents{lot}{\protect\addvspace{10\p@}}%
2592   \if@twocolumn
2593     \topnewpage[\@makechapterhead{#2}]%
2594   \else
2595     \@makechapterhead{#2}%
2596   \afterheading
2597   \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2598 \def\@makechapterhead#1{%
2599 % \vspace*{50\p@}%
2600 % \@tempskipa = \glueexpr \CTEX@chapter@beforeskip \relax
2601 % \ifodd \CTEX@chapter@fixbeforeskip
2602 % \CTEX@fixbeforeskip
2603 % \fi
2604 % \vspace*{\@tempskipa}%
2605 % {\normalfont \parindent \z@ \raggedright
2606 % {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2607 % \CTEX@chapter@format
2608 % \ifnum \c@secnumdepth >\m@ne

```

```

2609 <book> \if@mainmatter
2610 \ifodd \CTEX@chapter@numbering
2611 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2612 \CTEX@chaptername \CTEX@chapter@aftername
2613 \fi
2614 <book> \fi
2615 \fi
2616 \interlinepenalty\@M
2617 % \Huge \bfseries #1\par\nobreak
2618 \CTEX@chapter@titleformat{#1}%
2619 \CTEX@chapter@aftertitle
2620 \nobreak
2621 % \vskip 40\p@
2622 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2623 }}

```

(End definition for \makechapterhead.)

\@makeschapterhead

```

2624 \def\@makeschapterhead#1{%
2625 % \vspace*{50\p@}%
2626 \@tempskipa = \glueexpr \CTEX@chapter@beforeskip \relax
2627 \ifodd \CTEX@chapter@fixbeforeskip
2628 \CTEX@fixbeforeskip
2629 \fi
2630 \vspace*{\@tempskipa}%
2631 % {\normalfont \parindent \z@ \raggedright
2632 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2633 \CTEX@chapter@format
2634 \interlinepenalty\@M
2635 % \Huge \bfseries #1\par\nobreak
2636 \CTEX@chapter@titleformat{#1}
2637 \CTEX@chapter@aftertitle
2638 \nobreak
2639 % \vskip 40\p@
2640 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2641 }}

```

(End definition for \@makeschapterhead.)

2642 </book|report>

14.8.2.3 section 类的标题

\@startsection

```

2643 \def\@startsection#1#2#3#4#5#6{%
2644 \if@noskipsec \leavevmode \fi
2645 \par
2646 % \@tempskipa #4\relax
2647 % \@afterindenttrue
2648 % \ifdim \@tempskipa <\z@
2649 % \CTEX@chapter@afterindent
2650 % \fi
2651 \CTEX@update@sectionformat@n{#1}%
2652 \ifodd \CTEX@afterindent
2653 \CTEX@afterindenttrue
2654 \else
2655 \CTEX@afterindentfalse
2656 \fi
2657 \if@nobreak
2658 \everypar{}%
2659 \else
2660 % \addpenalty\@secpenalty\addvspace\@tempskipa
2661 \addpenalty\@secpenalty\addvspace{\glueexpr #4\relax}%

```

```

2662 \fi
2663 \@ifstar
2664 {\@ssect{#3}{#4}{#5}{#6}}%
2665 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

```

(End definition for \@startsection.)

\@seccntformat

```

2666 \def\@seccntformat#1{%
2667 % \csname the#1\endcsname\quad}%
2668 \csname CTEX@#1name\endcsname \csname CTEX@#1aftername\endcsname}

```

(End definition for \@seccntformat.)

\@sect

```

2669 \def\@sect#1#2#3#4#5#6[#7]#8{%
2670 \ifnum #2>\c@secnumdepth
2671 \let\@svsec\@empty
2672 \else
2673 \ifodd \csname CTEX@#1numbering\endcsname
2674 \refstepcounter{#1}%
2675 \protected@edef\@svsec{\@seccntformat{#1}\relax}%
2676 \else
2677 \CTEX@makeheadinganchor
2678 \let\@svsec\@empty
2679 \fi
2680 \fi
2681 % \@tempskipa #5\relax
2682 % \ifdim \@tempskipa>\z@
2683 \unless \ifodd \CTEX@runin
2684 \begingroup
2685 #6{%
2686 \@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2687 % \interlinepenalty \@M #8\@par}%
2688 \interlinepenalty \@M
2689 \csname CTEX@#1titleformat\endcsname{#8}%
2690 \csname CTEX@#1aftertitle\endcsname}%
2691 \endgroup
2692 \csname #1mark\endcsname{#7}%
2693 \addcontentsline{toc}{#1}{%
2694 \ifnum #2>\c@secnumdepth \else
2695 \ifodd \csname CTEX@#1numbering\endcsname
2696 % \protect\numberline{\csname the#1\endcsname}%
2697 \protect\numberline{\csname CTEXthe#1\endcsname}%
2698 \fi
2699 \fi
2700 #7}%
2701 \else
2702 \def\@svsechd{%
2703 #6{\hskip\glueexpr #3\relax
2704 % \@svsec #8}%
2705 \@svsec
2706 \csname CTEX@#1titleformat\endcsname{#8}%
2707 \csname CTEX@#1aftertitle\endcsname}%
2708 \csname #1mark\endcsname{#7}%
2709 \addcontentsline{toc}{#1}{%
2710 \ifnum #2>\c@secnumdepth \else
2711 \ifodd \csname CTEX@#1numbering\endcsname
2712 % \protect\numberline{\csname the#1\endcsname}%
2713 \protect\numberline{\csname CTEXthe#1\endcsname}%
2714 \fi
2715 \fi
2716 #7}}%
2717 \fi
2718 \@xsect{#5}}

```

(End definition for \@sect.)

\@ssect

```

2719 \def\@ssect#1#2#3#4#5{%
2720 % \@tempskipa #3\relax
2721 % \ifdim \@tempskipa>\z@
2722 \unless \ifodd \CTEX@runin
2723 \begingroup
2724 #4{%
2725 \hangfrom{\hskip\glueexpr #1\relax}%
2726 % \interlinepenalty \@M #5\@par}%
2727 \interlinepenalty \@M
2728 \CTEX@titleformat@n{#5}%
2729 \CTEX@aftertitle}%
2730 \endgroup
2731 \else
2732 % \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2733 \def\@svsechd{#4{\hskip\glueexpr #1\relax
2734 \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2735 \fi
2736 \@xsect{#3}}

```

(End definition for \@ssect.)

\@xsect

```

2737 \def\@xsect#1{%
2738 % \@tempskipa #1\relax
2739 % \ifdim \@tempskipa>\z@
2740 \unless \ifodd \CTEX@runin
2741 \par \nobreak
2742 % \vskip \@tempskipa
2743 \vskip\glueexpr #1\relax
2744 \@afterheading
2745 \else
2746 \@nobreakfalse
2747 \global\@noskipsectrue
2748 \everypar{%
2749 \if@noskipsec
2750 \global\@noskipsecfalse
2751 {\setbox\z@\lastbox}%
2752 \clubpenalty\@M
2753 \begingroup \@svsechd \endgroup
2754 \unskip
2755 % \@tempskipa #1\relax
2756 % \hskip -\@tempskipa
2757 \hskip\glueexpr #1\relax
2758 \else
2759 \clubpenalty \@clubpenalty
2760 \everypar{}%
2761 \fi}%
2762 \fi
2763 \ignorespaces}

```

(End definition for \@xsect.)

2764 <@@=ctex>

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

```

2765 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2766 {
2767 \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2768 \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2769 \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2770 \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }

```



```

2771 }
2772 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2773 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2774 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2775 \cs_new_eq:NN \CTEX@runin \c_false_bool

(End definition for \CTEX@update@sectionformat@n.)

2776 \int_zero:N \l__ctex_tmp_int
2777 \seq_map_inline:Nn \c__ctex_section_headings_seq
2778 {
2779   \int_incr:N \l__ctex_tmp_int
2780   \cs_gset_protected_nopar:cpx {#1}
2781   {
2782     \exp_not:N \@startsection {#1}
2783     { \int_use:N \l__ctex_tmp_int }
2784     { \exp_not:c { CTEX@#1@indent } }
2785     { \exp_not:c { CTEX@#1@beforeskip } }
2786     { \exp_not:c { CTEX@#1@afterskip } }
2787     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2788   }
2789 }

```

14.8.2.4 附录标题

```

appendix/name
appendix/number
appendix/numbering
2790 \keys_define:nn { ctex }
2791 {
2792   appendix .meta:nn = { ctex / appendix } {#1} ,
2793   appendix / name .code:n =
2794   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2795   appendix / number .tl_set:N = \CTEX@appendix@number ,
2796   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2797   appendix / numbering .initial:n = true
2798 }
2799 \tl_new:N \CTEX@preappendix
2800 \tl_new:N \CTEX@postappendix

```

(End definition for appendix/name, appendix/number, and appendix/numbering.)

\appendix

```

2801 \cs_new_eq:NN \CTEX@save@appendix \appendix
2802 \cs_gset_protected_nopar:Npn \appendix
2803 {
2804   \CTEX@save@appendix
2805   <*article>
2806   \gdef \CTEX@presection { \CTEX@preappendix }
2807   \gdef \CTEX@thesection { \CTEX@appendix@number }
2808   \gdef \CTEX@postsection { \CTEX@postappendix }
2809   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2810   </article>
2811   <*book|report>
2812   \gdef \CTEX@prechapter { \CTEX@preappendix }
2813   \gdef \CTEX@thechapter { \CTEX@appendix@number }
2814   \gdef \CTEX@postchapter { \CTEX@postappendix }
2815   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2816   </book|report>
2817 }

```

(End definition for \appendix.)

14.8.2.5 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 `\tts@<section>` 之中, 它的内容是

```
{<left>}{<right>}{<before>}{<after>}{<afterindent>}
```

其中 `<afterindent>` 为 1 或 0, 分别对应是否保留段首缩进。我们需要根据 CTeX 的 `runin` 和 `afterindent` 选项调整 `\ttl@<shape>` 和 `<afterindent>`。注意, 由 `\ttl@extract` 得的 `<before>` 和 `<after>` 的值总是非负的, 而 CTeX 的 `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 `\tts@<section>`。

```
2818 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
2819 {
2820   \@ifpackagewith { titlesec } { explicit }
2821   {
2822     \cs_set_eq:NN \__ctex_titlesec_format:Nn
2823       \__ctex_titlesec_format_explicit:Nn
2824   }
2825   { }
2826   \clist_map_inline:nn
2827     { indentafter , noindentafter , indentfirst , nonindentfirst }
2828     {
2829       \@ifpackagewith { titlesec } { ##1 }
2830       {
2831         \clist_map_break:n
2832           { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
2833       }
2834       { }
2835     }
2836   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
2837 }
2838 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
2839 {
2840   \__ctex_titlesec_format:n {#1}
2841   \exp_args:Nc \__ctex_titlesec_spacing:Nn { tts@#1 } {#1}
2842 }
2843 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
2844 {
2845   \cs_if_free:cF { ttl@#1 }
2846   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttl@#1 } {#1} }
2847 }
2848 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
2849 {
2850   \tl_set:Nx #1
2851   {
2852     \bool_if:cTF { CTEX@#2@runin }
2853     { \exp_not:N \ttl@runin }
2854     { \exp_not:N \ttl@hang }
2855     \tl_tail:N #1
2856   }
2857 }
```

```

2858 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
2859 {
2860   \cs_set_nopar:Npx #1 ##1
2861   {
2862     \bool_if:cTF { CTEX@#2@runin }
2863     { \exp_not:N \ttlh@runin }
2864     { \exp_not:N \ttlh@hang }
2865     \exp_args:No \tl_tail:n { #1 { } }
2866   }
2867 }
2868 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
2869 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
2870 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
2871 {
2872   \exp_not:n { {#1} {#2} {#3} {#4} }
2873   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
2874 }

```

(End definition for \ctex_titlesec_hook:.)

```

2875 \@ifpackageloaded { titlesec }
2876 { }
2877 {
2878   \ctex_at_end_package:nn { titlesec }
2879   {
2880     \@ifpackagewith { titlesec } { loadonly }
2881     { }
2882     { \ctex_titlesec_hook: }
2883   }
2884 }

```

14.8.2.6 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²⁰。

\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。

```

2885 \group_begin:
2886 \char_set_catcode_other:N \#
2887 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
2888 {
2889   \ctex_patch_cmd:Nnn \ttl@settopmark
2890   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
2891   {
2892     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
2893     \CTEX@titlepslabel@set {#1}
2894   }
2895   \ctex_patch_cmd:Nnn \ttl@setsubmark
2896   { \protect \@namedef { the#1 } { } }
2897   {
2898     \protect \@namedef { the#1 } { }
2899     \CTEX@titlepslabel@clear {#1}
2900   }
2901   \ctex_patch_cmd:Nnn \ttl@setsubmark
2902   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
2903   {
2904     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
2905     \CTEX@titlepslabel@set {#2}
2906   }
2907 }
2908 \group_end:

```

²⁰<https://github.com/CTeX-org/ctex-kit/issues/217>

(End definition for `\ctex_titleps_hook:`.)

`\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。

```
2909 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
2910 {
2911   \cs_if_free:cF { CTEXthe#1 }
2912   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
2913 }
2914 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
2915 {
2916   \cs_if_free:cF { CTEXthe#1 }
2917   { \protect \@namedef { CTEXthe#1 } { } }
2918 }
```

(End definition for `\CTEX@titlepslabel@set` and `\CTEX@titlepslabel@clear`.)

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

```
2919 \ctex_at_end_package:nn { titlesec }
2920 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
2921 \ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }
```

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

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

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

14.8.3 目录标签的宽度

`\numberline`

```
2922 \cs_new_protected:Npn \CTEX@toc@width@n #1
2923 {
2924   \hbox_set:Nn \l__ctex_tmp_box {#1}
2925   \dim_set:Nn \@tempdima
2926   {
2927     \dim_max:nn { \@tempdima }
2928     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2929   }
2930 }
2931 \group_begin:
2932 \char_set_catcode_other:N \#
2933 \use:n
2934 {
2935   \group_end:
2936   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
2937   { \CTEX@toc@width@n {#1} }
2938   { }
2939   { \ctex_patch_failure:N \numberline }
2940   \ifpackageloaded { tocloft }
2941   { }
2942   {
2943     \ctex_at_end_package:nn { tocloft }
2944     {
2945       \ctex_preto_cmd:NnnTF \numberline
2946       { \char_set_catcode_letter:n { 64 } }
2947       { \CTEX@toc@width@n {#1} }
2948       { }
2949       { \ctex_patch_failure:N \numberline }
2950     }
2951   }
2952 }
```

(End definition for \numberline.)

14.8.4 页眉信息的修改

\ps@headings

```

2953 <*article>
2954 \ctex_patch_cmd:Nnn \ps@headings
2955   { \thesection \quad }
2956   { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2957 \if@twoside
2958   \ctex_patch_cmd:Nnn \ps@headings
2959     { \thesubsection \quad }
2960     { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2961 \fi:
2962 </article>
2963 <*book|report>
2964 \ctex_patch_cmd:Nnn \ps@headings
2965   { \@chapapp\ \thechapter.\ }
2966   { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2967 \if@twoside
2968   \ctex_patch_cmd:Nnn \ps@headings
2969     { \thesection.\ }
2970     { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2971 \fi:
2972 </book|report>

```

(End definition for \ps@headings.)

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

```

2973 \ctex_at_end_package:nn { fancyhdr }
2974 {
2975 <*article>
2976   \ctex_patch_cmd:Nnn \ps@fancy
2977     { \thesection \hskip 1em \relax }
2978     { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2979   \ctex_patch_cmd:Nnn \ps@fancy
2980     { \thesubsection \hskip 1em \relax }
2981     { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2982 </article>
2983 <*book|report>
2984   \ctex_patch_cmd:Nnn \ps@fancy
2985     { \@chapapp\ \thechapter.\ }
2986 <book>   { \if@mainmatter \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi \fi }
2987 <report> { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2988   \ctex_patch_cmd:Nnn \ps@fancy
2989     { \thesection.\ }
2990     { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2991 </book|report>
2992 }

```

(End definition for \ps@fancy.)

2993 </article|book|report>

14.8.5 beamer 标题页模板的修改

```

2994 <*beamer>
2995 \ExplSyntaxOff

    对应 \partpage。

2996 \defbeamer*template*{part page}{CTEX}[1] [] {

```

```

2997 \beginngroup
2998 % \centering
2999 % {\usebeamerfont{part name}\usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3000 % \vskip1em\par
3001 \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3002 \parindent \dimexpr \CTEX@part@indent \relax
3003 \CTEX@part@format
3004 \ifodd \CTEX@part@numbering
3005 \CTEX@partname \CTEX@part@aftername
3006 \fi
3007 \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3008 % \usebeamerfont{part title}\insertpart\par
3009 \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3010 \end{beamercolorbox}
3011 \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3012 \endgroup
3013 }

```

对应 \sectionpage。

```

3014 \defbeamertemplate*{section page}{CTEX}[1][]{
3015 \beginngroup
3016 % \centering
3017 % {\usebeamerfont{section name}\usebeamercolor[fg]{section name}\sectionname~\insertsectionname}
3018 % \vskip1em\par
3019 \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3020 \parindent \dimexpr \CTEX@section@indent \relax
3021 \CTEX@section@format
3022 \ifodd \CTEX@section@numbering
3023 \CTEX@sectionname \CTEX@section@aftername
3024 \fi
3025 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3026 % \usebeamerfont{section title}\insertsection\par
3027 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3028 \end{beamercolorbox}
3029 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3030 \endgroup
3031 }

```

对应 \subsectionpage。

```

3032 \defbeamertemplate*{subsection page}{CTEX}[1][]{
3033 \beginngroup
3034 % \centering
3035 % {\usebeamerfont{subsection name}\usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionname}
3036 % \vskip1em\par
3037 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3038 \parindent \dimexpr \CTEX@subsection@indent \relax
3039 \CTEX@subsection@format
3040 \ifodd \CTEX@subsection@numbering
3041 \CTEX@subsectionname \CTEX@subsection@aftername
3042 \fi
3043 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3044 % \usebeamerfont{subsection title}\insertsubsection\par
3045 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3046 \end{beamercolorbox}
3047 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3048 \endgroup
3049 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3050 \defbeamertemplatealias{part page}{default}{CTEX}
3051 \defbeamertemplatealias{section page}{default}{CTEX}
3052 \defbeamertemplatealias{subsection page}{default}{CTEX}
3053 \ExplSyntaxOn
3054 </beamer>

```

14.8.6 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。

```

3055 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
3056 {
3057   \protected@edef \@currentlabel
3058   {
3059     \cs_if_exist:cTF { CTEX@the#1 }
3060     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3061     { \exp_not:o { \@currentlabel } }
3062   }
3063 }
```

(End definition for `\refstepcounter`.)

`\ctex_varioref_hook:` 关于标签引用的宏包可能会修改 `\refstepcounter`。其中 `cleveref` 和 `hyperref` 宏包都会保存之前的定义, 并且它们都要求尽可能晚的被载入, 所以对我们上述的修改影响不大。需要注意的是 `varioref` 宏包, 如果它在 `CTEX` 之后被载入, 我们之前的修改将会被覆盖。

```

3064 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3065 {
3066   \seq_map_inline:Nn \c__ctex_headings_seq
3067   { \ctex_fix_varioref_label:n { ##1 } }
3068 }
```

(End definition for `\ctex_varioref_hook:`.)

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

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

它的第二个参数必须放在花括号内, 否则将会被作为宏的定界符号。

```

3069 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3070 {
3071   \tl_if_empty:cT { p@#1 }
3072   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3073 }
```

(End definition for `__ctex_fix_varioref_label:n`.)

如果 `varioref` 已经被载入, 则使用它来设置。

```

3074 \@ifpackageloaded { varioref }
3075 { \ctex_varioref_hook: }
3076 {
3077   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3078   \RenewDocumentCommand \refstepcounter { m }
3079   {
3080     \CTEX@save@refstepcounter {#1}
3081     \CTEX@setcurrentlabel@n {#1}
3082   }
3083   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3084 }
```

14.8.7 载入 `<scheme>` 文件

```

3085 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3086 </class|heading>
```

14.8.8 标题格式的 scheme 定义

下面使用 C_T_E_X 文档类的设置方式,plain 模拟标准文档类直接定义或以 \startsection 设定的章节标题格式,chinese 汉化的标题格式。

```

3087 <*scheme&(article|book|report|beamer)>

3088 \keys_set:nn { ctex / part }
3089 {
3090     aftertitle = \par ,
3091     <*plain>
3092     name       = \partname \space ,
3093     <*article|book|report>
3094     number     = \thepart ,
3095     </article|book|report>
3096     <*beamer>
3097     number     = \insertromanpartnumber ,
3098     </beamer>
3099     </plain>
3100     <*chinese>
3101     number     = \chinese { part } ,
3102     </chinese>
3103     <*article>
3104     beforeskip = 4ex ,
3105     afterskip  = 3ex ,
3106     <*plain>
3107     format     = \raggedright ,
3108     nameformat = \Large \bfseries ,
3109     aftername   = \par \nobreak ,
3110     titleformat = \huge \bfseries ,
3111     afterindent = false
3112     </plain>
3113     <*chinese>
3114     format     = \Large \bfseries \centering ,
3115     aftername   = \quad ,
3116     afterindent = true
3117     </chinese>
3118     </article>
3119     <*book|report>
3120     aftername   = \par \vskip 20 \p@ ,
3121     beforeskip  = 0pt \@plus 1fil ,
3122     afterskip   = 0pt \@plus 1fil ,
3123     pagestyle   = plain ,
3124     <*plain>
3125     format     = \centering ,
3126     nameformat = \huge \bfseries ,
3127     titleformat = \Huge \bfseries
3128     </plain>
3129     <*chinese>
3130     format     = \huge \bfseries \centering
3131     </chinese>
3132     </book|report>
3133     <*beamer>
3134     format     = \centering ,
3135     nameformat = \usebeamerfont { part ~ name }
3136                \usebeamercolor [fg] { part ~ name } ,
3137     aftername   = \vskip 1em \par ,
3138     titleformat = \usebeamerfont { part ~ title }
3139     </beamer>
3140 }

3141 <*book|report>
3142 \keys_set:nn { ctex / chapter }
3143 {
3144     pagestyle = plain ,
3145     aftertitle = \par ,
3146     beforeskip = 50 \p@ ,

```



```

3147     afterskip    = 40 \p@ ,
3148     <*plain>
3149     name         = \chaptername \space ,
3150     number       = \thechapter ,
3151     format       = \raggedright ,
3152     nameformat    = \huge \bfseries ,
3153     aftername     = \par \nobreak \vskip 20 \p@ ,
3154     titleformat   = \Huge \bfseries ,
3155     afterindent   = false
3156 </plain>
3157 <*chinese>
3158     number       = \chinese { chapter } ,
3159     format       = \huge \bfseries \centering ,
3160     aftername     = \quad ,
3161     afterindent   = true
3162 </chinese>
3163 }
3164 </book|report>

3165 <@@=->

3166 \keys_set:nn { ctex / section }
3167 {
3168     <*article|book|report>
3169     number       = \thesection ,
3170     aftername     = \quad ,
3171     aftertitle    = \@@par ,
3172     beforekip     = 3.5ex \@plus 1ex \@minus .2ex ,
3173     afterskip     = 2.3ex \@plus .2ex ,
3174     runin         = false ,
3175     <*plain>
3176     format       = \Large \bfseries ,
3177     afterindent   = false
3178 </plain>
3179 <*chinese>
3180     format       = \Large \bfseries \centering ,
3181     afterindent   = true
3182 </chinese>
3183 </article|book|report>
3184 <*beamer>
3185 <*plain>
3186     name         = \sectionname \space ,
3187 </plain>
3188     format       = \centering ,
3189     number       = \insertsectionnumber ,
3190     nameformat    = \usebeamerfont { section ~ name }
3191                   \usebeamercolor [fg] { section ~ name } ,
3192     aftername     = \vskip 1em \par ,
3193     titleformat   = \usebeamerfont { section ~ title } ,
3194     aftertitle    = \par
3195 </beamer>
3196 }

3197 \keys_set:nn { ctex / subsection }
3198 {
3199     <*article|book|report>
3200     number       = \thesubsection ,
3201     format       = \large \bfseries ,
3202     aftername     = \quad ,
3203     aftertitle    = \@@par ,
3204     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3205     afterskip     = 1.5ex \@plus .2ex ,
3206     runin         = false ,
3207     <*plain>
3208     afterindent   = false
3209 </plain>
3210 <*chinese>
3211     afterindent   = true

```

```

3212 </chinese>
3213 </article|book|report>
3214 <*beamer>
3215 <*plain>
3216     name          = \subsectionname \space ,
3217     number         = \insertsubsectionnumber ,
3218 </plain>
3219 <*chinese>
3220     number         = \arabic { section } . \arabic { subsection } ,
3221 </chinese>
3222     format         = \centering ,
3223     nameformat      = \usebeamerfont { subsection ~ name }
3224                     \usebeamercolor [fg] { subsection ~ name } ,
3225     aftername       = \vskip 1em \par ,
3226     titleformat     = \usebeamerfont { subsection ~ title } ,
3227     aftertitle      = \par
3228 </beamer>
3229 }

3230 <*article|book|report>

3231 \keys_set:nn { ctex / subsubsection }
3232 {
3233     number         = \thesubsubsection ,
3234     format         = \normalsize \bfseries ,
3235     aftername       = \quad ,
3236     aftertitle      = \@@par ,
3237     beforekip       = 3.25ex \@plus 1ex \@minus .2ex ,
3238     afterskip       = 1.5ex \@plus .2ex ,
3239     runin           = false ,
3240 <*plain>
3241     afterindent     = false
3242 </plain>
3243 <*chinese>
3244     afterindent     = true
3245 </chinese>
3246 }

3247 \keys_set:nn { ctex / paragraph }
3248 {
3249     number         = \theparagraph ,
3250     format         = \normalsize \bfseries ,
3251     aftername       = \quad ,
3252     beforekip       = 3.25ex \@plus 1ex \@minus .2ex ,
3253 <*plain>
3254     afterindent     = false
3255 </plain>
3256 <*chinese>
3257     afterindent     = true
3258 </chinese>
3259 }

3260 \keys_set:nn { ctex / subparagraph }
3261 {
3262     number         = \thesubparagraph ,
3263     format         = \normalsize \bfseries ,
3264     aftername       = \quad ,
3265     beforekip       = 3.25ex \@plus 1ex \@minus .2ex ,
3266 <*plain>
3267     afterindent     = false
3268 </plain>
3269 <*chinese>
3270     afterindent     = true
3271 </chinese>
3272 }

```

处理 sub3section 与 sub4section 的格式。

```

3273 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3274 {
3275   \keys_set:nn { ctex / paragraph }
3276   {
3277     aftertitle = \@@par ,
3278     afterskip  = 1ex \@plus .2ex ,
3279     runin      = false
3280   }
3281 }
3282 {
3283   \keys_set:nn { ctex / paragraph }
3284   {
3285     afterskip  = 1em ,
3286     runin      = true
3287   }
3288 }
3289 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3290 {
3291   \keys_set:nn { ctex / subparagraph }
3292   {
3293     aftertitle = \@@par ,
3294     afterskip  = 1ex \@plus .2ex ,
3295     runin      = false
3296   }
3297 }
3298 {
3299   \keys_set:nn { ctex / subparagraph }
3300   {
3301     afterskip  = 1em ,
3302     runin      = true
3303   }
3304 }
3305 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3306 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3307 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3308 <@@=ctex>

```

处理附录的格式。

```

3309 \keys_set:nn { ctex / appendix }
3310 <*article>
3311 { number      = \@Alph \c@section }
3312 </article>
3313 <*book|report>
3314 {
3315   name        = \appendixname \space ,
3316   number      = \@Alph \c@chapter
3317 }
3318 </book|report>
3319 </article|book|report>
3320 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

```

3321 <*ctex|ctexheading>

```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3322 \clist_map_inline:nn { article , book , report , beamer }
3323 {
3324   \@@ifclassloaded {#1}
3325   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3326   { }
3327 }

```

(End definition for \c__ctex_std_class_tl.)

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。

```

3328 \msg_new:nnn { ctex } { not-standard-class }
3329 {
3330     None~of~the~standard~document~classes~was~loaded.\
3331     Heading~`#1'~is~selected.\
3332     ctex~may~not~work~as~expected.
3333 }
3334 <ctex>\bool_if:NTF \l__ctex_heading_bool
3335 <ctexheading>\use:n
3336 {
3337     \tl_if_exist:NTF \c__ctex_std_class_tl
3338     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3339     {
3340         \cs_if_exist:NTF \chapter
3341         {
3342             \cs_if_exist:NF \if@mainmatter
3343             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3344             \tl_const:Nn \c__ctex_class_tl { book }
3345         }
3346         { \tl_const:Nn \c__ctex_class_tl { article } }
3347         \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3348     }
3349     \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3350 }
3351 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3352 </ctex|ctexheading>

```

14.8.10 标题配置文件

```

3353 <*name>
3354 \keys_set:known:nn { ctex }
3355 {
3356     contentsname = 目录 ,
3357     listfigurename = 插图 ,
3358     listtablename = 表格 ,
3359     figurename = 图 ,
3360     tablename = 表 ,
3361     abstractname = 摘要 ,
3362     indexname = 索引 ,
3363     bibname = 参考文献 ,
3364     appendixname = 附录 ,
3365     proofname = 证明 ,
3366     algorithmname = 算法 ,
3367     refname = 参考文献 ,
3368     continuation = (续) ,
3369     part / name = { 第 , 部分 } ,
3370     chapter / name = { 第 , 章 }
3371 }
3372 </name>

```

14.9 chinese 方案的其他设置

3373 <*scheme&chinese>

chinese 在标准文档类下的页面格式总采用 headings。

3374 <article|book|report>\pagestyle { headings }

日期格式。

3375 \keys_set:nn { ctex } { today = small }

若用户未设置宏包选项 autoindent, 则自动调整首行缩进。

```

3376 \ctex_if_autoindent_touched:F
3377 { \keys_set:nn { ctex } { 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。

```

3378 <!*generic>
3379 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3380 <!*beamer>
3381 {
3382   \uselanguage { ChineseGBK }
3383   \languagealias { chinese } { ChineseGBK }
3384   \ctex_file_input:n { ctex-name-gbk.cfg }
3385 }
3386 {
3387   \uselanguage { ChineseUTF8 }
3388   \languagealias { chinese } { ChineseUTF8 }
3389   \ctex_file_input:n { ctex-name-utf8.cfg }
3390 }

```

让 **translator** 包优先查找中文翻译。

```

3391 \clist_put_left:Nn \trans@languagepath { chinese }
3392 </beamer>
3393 <!*beamer>
3394 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3395 { \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。

```

3396 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3397 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }

```

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。

```

3398 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3399 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3400 </beamer>
3401 </!generic>

```

不使用标题定义时的通用设置。

```

3402 <!*generic>
3403 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3404 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3405 {
3406   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3407   {
3408     \uselanguage { ChineseGBK }
3409     \languagealias { chinese } { ChineseGBK }
3410     \ctex_file_input:n { ctex-name-gbk.cfg }
3411   }
3412   {
3413     \uselanguage { ChineseUTF8 }
3414     \languagealias { chinese } { ChineseUTF8 }
3415     \ctex_file_input:n { ctex-name-utf8.cfg }
3416   }
3417   \clist_put_left:Nn \trans@languagepath { chinese }
3418 }
3419 {
3420   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3421   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3422   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3423   \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3424   { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3425   \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3426   { \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。

```

3427 \tl_if_exist:NT \c__ctex_std_class_tl
3428 { \RequirePackage { indentfirst } }
3429 }
3430 </generic>
3431 </scheme&chinese>

```

14.10 中文字号

```

3432 <*class|ctex|ctexsize>

```

`\zihao`

```

3433 \NewDocumentCommand \zihao { m }
3434 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

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

`\ctex_zihao:n`

```

3435 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3436 {
3437   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3438   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3439   { \msg_error:nnn { ctex } { fontsize } {#1} }
3440 }
3441 \msg_new:nnnn { ctex } { fontsize }
3442 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3443 {
3444   The~old~font~size~is~used~if~you~continue.\\
3445   The~available~font~sizes~are~listed~as~follow.\\
3446   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3447 }

```

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

14.10.1 定义中文字号

`\c__ctex_font_size_prop` 基础行距是字号的 1.2 倍,采用 ϵ -TeX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确²¹。

```

3448 \prop_new:N \c__ctex_font_size_prop
3449 \seq_new:N \c__ctex_font_size_seq
3450 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3451 {
3452   \use:x
3453   {
3454     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3455     {
3456       { \dim_to_decimal:n {#2} }
3457       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3458     }
3459   }
3460   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3461 }
3462 \clist_map_inline:nn
3463 {
3464   { 8 } { 5 bp } ,
3465   { 7 } { 5.5 bp } ,
3466   { -6 } { 6.5 bp } ,
3467   { 6 } { 7.5 bp } ,
3468   { -5 } { 9 bp } ,
3469   { 5 } { 10.5 bp } ,

```

²¹<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

3470 { -4 } { 12 bp } ,
3471 { 4 } { 14 bp } ,
3472 { -3 } { 15 bp } ,
3473 { 3 } { 16 bp } ,
3474 { -2 } { 18 bp } ,
3475 { 2 } { 22 bp } ,
3476 { -1 } { 24 bp } ,
3477 { 1 } { 26 bp } ,
3478 { -0 } { 36 bp } ,
3479 { 0 } { 42 bp }
3480 }
3481 { \__ctex_save_font_size:nn #1 }

```

(End definition for \c__ctex_font_size_prop.)

\ctex_declare_math_sizes:nnnn

```

3482 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3483 {
3484   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3485   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3486 }

```

(End definition for \ctex_declare_math_sizes:nnnn.)

__ctex_get_font_sizes:Nn

```

3487 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3488 {
3489   \tl_clear:N #1
3490   \tl_map_inline:nn {#2}
3491   {
3492     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3493     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3494     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3495   }
3496 }

```

(End definition for __ctex_get_font_sizes:Nn.)

```

3497 \clist_map_inline:nn
3498 {
3499   { 8 } { 8 } { 5pt } { 5pt } ,
3500   { 7 } { 7 } { 5pt } { 5pt } ,
3501   { -6 } { -6 } { 5pt } { 5pt } ,
3502   { 6 } { 6 } { 5pt } { 5pt } ,
3503   { -5 } { -5 } { 6pt } { 5pt } ,
3504   { 5 } { 5 } { 7pt } { 5pt } ,
3505   { -4 } { -4 } { 8pt } { 6pt } ,
3506   { 4 } { 4 } { 5 } { 6 } ,
3507   { -3 } { -3 } { -4 } { -5 } ,
3508   { 3 } { 3 } { 4 } { 5 } ,
3509   { -2 } { -2 } { -3 } { -4 } ,
3510   { 2 } { 2 } { 3 } { 4 } ,
3511   { -1 } { -1 } { -2 } { -3 } ,
3512   { 1 } { 1 } { 2 } { 3 } ,
3513   { -0 } { -0 } { -1 } { -2 } ,
3514   { 0 } { 0 } { 1 } { 2 }
3515 }
3516 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

\ctex_set_font_size:Nnn

```

3517 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3518 {
3519     \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3520     { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3521     { \msg_error:nnn { ctex } { fontsize } {#2} }
3522 }
3523 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3524 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

(End definition for \ctex_set_font_size:Nnn.)

3525 \if_case:w \g__ctex_font_size_flag
3526   \ctex_file_input:n { ctex-c5size.clo }
3527 \or:
3528   \ctex_file_input:n { ctex-cs4size.clo }
3529 \fi:

3530 </class|ctex|ctexsize>

3531 <*c5size>
3532 \ctex_set_font_size:Nnn \normalsize { 5 }
3533 {
3534     \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3535     \abovedisplayshortskip \z@ \@plus3\p@
3536     \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3537     \belowdisplayskip \abovedisplayskip
3538     \let\@listi\@listI
3539 }
3540 \ctex_set_font_size:Nnn \small { -5 }
3541 {
3542     \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3543     \abovedisplayshortskip \z@ \@plus2\p@
3544     \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3545     \def\@listi{\leftmargin\leftmarginI
3546               \topsep 4\p@ \@plus2\p@ \@minus2\p@
3547               \parsep 2\p@ \@plus\p@ \@minus\p@
3548               \itemsep \parsep}
3549     \belowdisplayskip \abovedisplayskip
3550 }
3551 \ctex_set_font_size:Nnn \footnotesize { 6 }
3552 {
3553     \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3554     \abovedisplayshortskip \z@ \@plus\p@
3555     \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3556     \def\@listi{\leftmargin\leftmarginI
3557               \topsep 3\p@ \@plus\p@ \@minus\p@
3558               \parsep 2\p@ \@plus\p@ \@minus\p@
3559               \itemsep \parsep}
3560     \belowdisplayskip \abovedisplayskip
3561 }
3562 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3563 \ctex_set_font_size:Nnn \tiny { 7 } { }
3564 \ctex_set_font_size:Nnn \large { -4 } { }
3565 \ctex_set_font_size:Nnn \Large { -3 } { }
3566 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3567 \ctex_set_font_size:Nnn \huge { 2 } { }
3568 \ctex_set_font_size:Nnn \Huge { 1 } { }
3569 </c5size>
3570 <*cs4size>
3571 \ctex_set_font_size:Nnn \normalsize { -4 }
3572 {
3573     \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3574     \abovedisplayshortskip \z@ \@plus3\p@
3575     \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3576     \belowdisplayskip \abovedisplayskip
3577     \let\@listi\@listI
3578 }
3579 \ctex_set_font_size:Nnn \small { 5 }

```



```

3580 {
3581   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3582   \abovedisplayshortskip \z@ \@plus3\p@
3583   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3584   \def\@listi{\leftmargin\leftmargini
3585             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3586             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3587             \itemsep \parsep}
3588   \belowdisplayskip \abovedisplayskip
3589 }
3590 \ctex_set_font_size:Nnn \footnotesize { -5 }
3591 {
3592   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3593   \abovedisplayshortskip \z@ \@plus3\p@
3594   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3595   \def\@listi{\leftmargin\leftmargini
3596             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3597             \parsep 3\p@ \@plus2\p@ \@minus\p@
3598             \itemsep \parsep}
3599   \belowdisplayskip \abovedisplayskip
3600 }
3601 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3602 \ctex_set_font_size:Nnn \tiny { -6 } { }
3603 \ctex_set_font_size:Nnn \large { -3 } { }
3604 \ctex_set_font_size:Nnn \Large { -2 } { }
3605 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3606 \ctex_set_font_size:Nnn \huge { -1 } { }
3607 \ctex_set_font_size:Nnn \Huge { 1 } { }
3608 </cs4size>
3609 <ctexsize>\normalsize
3610 <*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`。

```

3611 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3612 {
3613   \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。

```

3614   \group_begin: \footnotesize \exp_args:NNNo \group_end:
3615   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3616 }

```

(End definition for `\footnotesep`.)

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

```

3617 \normalsize

```

14.12 其它功能

CTeX `ctex-faq.sty` 中的定义是

```

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

```

然而 `\mathbb` 未必有定义，这里就不采用它了，只定义最简单的形式。同 `hologo` 宏包的设置类似，`CTeX` 可以用在 `\csname` 和 PDF 书签中。

```

3618 \NewDocumentCommand \CTeX { }
3619 { \ifincsname CTeX \else: C \TeX \fi: }
3620 \ctex_at_end_package:nn { hyperref }
3621 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

(End definition for \CTeX. This function is documented on page 24.)

captiondelimit 过时选项。

```

3622 \keys_define:nn { ctex }
3623 {
3624   captiondelimit .code:n =
3625   {
3626     \msg_warning:nnn { ctex } { deprecated-option }
3627     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3628   }
3629 }

```

(End definition for captiondelimit. This function is documented on page 29.)

```
3630 </class|ctex>
```

14.12.1 列表环境的缩进

\verse 只在使用文档类的时候修改诗歌和引用环境的缩进。
\quotation

```

3631 <*scheme&chinese&(article|book|report)>
3632 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3633 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3634 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3635 </scheme&chinese&(article|book|report)>

```

(End definition for \verse and \quotation.)

```
3636 <*class|ctex>
```

14.13 载入中文字体

\ctex_fontset_error:n 字库不可用时给出紧急错误信息, 停止读取定义文件。

```

3637 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3638 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3639 \msg_new:nnn { ctex } { fontset-unavailable }
3640 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }

```

(End definition for \ctex_fontset_error:n.)

\ctex_load_fontset: 如果用户没有指定字体, 则探测操作系统, 载入相应的字体配置。

```

3641 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3642 {
3643   \tl_if_empty:NTF \g__ctex_fontset_tl
3644   {
3645     \ctex_detected_platform:
3646     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3647   }
3648   {
3649     \file_if_exist:NTF { ctex-fontset- \g__ctex_fontset_tl .def }
3650     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3651     {
3652       \use:x
3653       {
3654         \ctex_detected_platform:
3655         \msg_error:nxxx { ctex } { fontset-not-found }

```

```

3656             { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3657         }
3658         \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3659     }
3660 }
3661 }
3662 \@onlypreamble \ctex_load_fontset:
3663 \msg_new:nnnn { ctex } { fontset-not-found }
3664 {
3665     CTex~fontset~`#1'~could~not~be~found.\\
3666     Fontset~`#2'~will~be~used~instead.
3667 }
3668 { You~may~run~`mktexlsr'~firstly. }

```

(End definition for \ctex_load_fontset:.)

fontset 在导言区通过 \ctexset 载入中文字库的选项。

```

3669 \keys_define:nn { ctex }
3670 {
3671     fontset .code:n =
3672     {
3673         \ctex_if_preamble:TF
3674         {
3675             \str_if_eq_x:nnTF {#1} { none }
3676             { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3677             {
3678                 \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3679                 {
3680                     \tl_gset:Nx \g__ctex_fontset_tl {#1}
3681                     \ctex_load_fontset:
3682                 }
3683                 {
3684                     \msg_error:nnxx { ctex } { fontset-loaded }
3685                     { \g__ctex_fontset_tl } {#1}
3686                 }
3687             }
3688         }
3689         { \msg_error:nn { ctex } { fontset-only-preamble } }
3690     }
3691 }
3692 \msg_new:nnnn { ctex } { fontset-loaded }
3693 {
3694     CTex~fontset~`#1'~has~been~loaded.
3695     \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
3696 }
3697 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3698 \msg_new:nnn { ctex } { fontset-only-preamble }
3699 {
3700     The~`fontset'~option~can~be~used~only~in~preamble.
3701 }

```

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

载入中文字库。

```

3702 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3703 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

3704 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3705 </class|ctex>

```

```

3706 <*config>
3707 %%
3708 </config>

```

14.14.2 ctexopts.cfg

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

```

3709 <*ctexopts>
3710 %%
3711 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3712 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

3713 <*c19|c70>
3714 %%
3715 %% Chinese characters
3716 %%
3717 <c19>%% character set: GBK (extension of GB 2312)
3718 <c70>%% character set: Unicode
3719 %% font encoding: Unicode
3720 %%
3721 </c19|c70>

```

CJK 宏包使用的字体族。

```

3722 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3723 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3724 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3725 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3726 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3727 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

upTeX 使用的字体族。upTeX 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。

```

3728 <*uptex|aptex>
3729 \DeclareKanjiFamily{JY2}{rm}{-}{}
3730 \DeclareKanjiFamily{JY2}{sf}{-}{}
3731 \DeclareKanjiFamily{JY2}{tt}{-}{}
3732 \DeclareKanjiFamily{JT2}{rm}{-}{}
3733 \DeclareKanjiFamily{JT2}{sf}{-}{}
3734 \DeclareKanjiFamily{JT2}{tt}{-}{}
3735 </uptex|aptex>

3736 <*rm>
3737 <*c19>
3738 \DeclareFontShape{C19}{rm}{m}{n}{-}{} CJK * gbksong}{\CJKnormal}
3739 \DeclareFontShape{C19}{rm}{b}{n}{-}{} CJK * gbkhei}{\CJKnormal}
3740 \DeclareFontShape{C19}{rm}{bx}{n}{-}{} CJK * gbkhei}{\CJKnormal}
3741 \DeclareFontShape{C19}{rm}{m}{sl}{-}{} CJK * gbksongsl}{\CJKnormal}
3742 \DeclareFontShape{C19}{rm}{b}{sl}{-}{} CJK * gbkheisl}{\CJKnormal}
3743 \DeclareFontShape{C19}{rm}{bx}{sl}{-}{} CJK * gbkheisl}{\CJKnormal}
3744 \DeclareFontShape{C19}{rm}{m}{it}{-}{} CJK * gbkkai}{\CJKnormal}
3745 \DeclareFontShape{C19}{rm}{b}{it}{-}{} CJKb * gbkkai}{\CJKbold}
3746 \DeclareFontShape{C19}{rm}{bx}{it}{-}{} CJKb * gbkkai}{\CJKbold}
3747 </c19>
3748 <*c70>
3749 \DeclareFontShape{C70}{rm}{m}{n}{-}{} CJK * unisong}{\CJKnormal}
3750 \DeclareFontShape{C70}{rm}{b}{n}{-}{} CJK * unihei}{\CJKnormal}
3751 \DeclareFontShape{C70}{rm}{bx}{n}{-}{} CJK * unihei}{\CJKnormal}
3752 \DeclareFontShape{C70}{rm}{m}{sl}{-}{} CJK * unisongsl}{\CJKnormal}
3753 \DeclareFontShape{C70}{rm}{b}{sl}{-}{} CJK * uniheisl}{\CJKnormal}
3754 \DeclareFontShape{C70}{rm}{bx}{sl}{-}{} CJK * uniheisl}{\CJKnormal}

```

```

3755 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3756 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3757 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3758 </c70>
3759 </rm>
3760 <*uptex|aptex>
3761 \DeclareFontShape{JY2}{rm}{m}{n}{<-> upzhserif-h}{\}
3762 \DeclareFontShape{JY2}{rm}{m}{it}{<-> upzhserifit-h}{\}
3763 \DeclareFontShape{JY2}{rm}{bx}{n}{<-> upzhserifb-h}{\}
3764 \DeclareFontShape{JT2}{rm}{m}{n}{<-> upzhserif-v}{\}
3765 \DeclareFontShape{JT2}{rm}{m}{it}{<-> upzhserifit-v}{\}
3766 \DeclareFontShape{JT2}{rm}{bx}{n}{<-> upzhserifb-v}{\}
3767 </uptex|aptex>
3768 <*sf>
3769 <*c19>
3770 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3771 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3772 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3773 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3774 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3775 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3776 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3777 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3778 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3779 </c19>
3780 <*c70>
3781 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyousl}{\CJKnormal}
3782 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyousl}{\CJKbold}
3783 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyousl}{\CJKbold}
3784 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
3785 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3786 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3787 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyousl}{\CJKnormal}
3788 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyousl}{\CJKbold}
3789 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyousl}{\CJKbold}
3790 </c70>
3791 </sf>
3792 <*uptex|aptex>
3793 \DeclareFontShape{JY2}{sf}{m}{n}{<-> upzhsans-h}{\}
3794 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{\}
3795 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{\}
3796 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{\}
3797 </uptex|aptex>
3798 <*tt>
3799 <*c19>
3800 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
3801 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
3802 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
3803 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
3804 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3805 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3806 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
3807 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
3808 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
3809 </c19>
3810 <*c70>
3811 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
3812 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
3813 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
3814 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
3815 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
3816 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
3817 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
3818 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
3819 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
3820 </c70>

```

```

3821 </tt>
3822 <*uptex|aptex>
3823 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{}
3824 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{}
3825 </uptex|aptex>
3826 <*fontset>

```

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

ctex 包利用 C:\bootfont.bin 文件的存在性来判断是否使用 Windows XP 版本, 分别载入新旧字体设置。(注: ctex 包不支持 Windows 2000 以前使用 simsun.ttf 的旧宋体文件名。)

```

3827 <*windows>
3828 \file_if_exist:nTF { C:/bootfont.bin }
3829 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3830 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3831 </windows>

```

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

```

3832 <*windowsnew>
3833 \tl_new:N \l__ctex_msyh_suffix_tl
3834 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
3835 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
3836 {
3837   \file_if_exist:nF { msyh.ttc }
3838   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
3839 }
3840 </windowsnew>
3841 <*windowsnew|windowsold>
3842 \sys_if_engine_pdftex:TF
3843 {
3844   \ctex_zhmap_case:nnn
3845   {
3846     \ctex_punct_set:n { windows }
3847     \setCJKmainfont
3848     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3849 <*windowsold>
3850     \setCJKsansfont { simhei.ttf }
3851     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3852 </windowsold>
3853 <*windowsnew>
3854     \setCJKsansfont [ BoldFont = msyhbdl__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
3855     \setCJKfamilyfont { zhyahei }
3856     [ BoldFont = msyhbdl__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
3857     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3858     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3859 </windowsnew>
3860     \setCJKmonofont { simfang.ttf }
3861     \setCJKfamilyfont { zhkai } { simkai.ttf }
3862     \setCJKfamilyfont { zhfs } { simfang.ttf }
3863     \setCJKfamilyfont { zhsong } { simsun.ttc }
3864     \setCJKfamilyfont { zhhei } { simhei.ttf }
3865     \setCJKfamilyfont { zhli } { simli.ttf }
3866     \setCJKfamilyfont { zhyou } { simyou.ttf }
3867     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3868     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3869     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3870     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3871   }
3872   {
3873     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3874     \ctex_punct_set:n { windows }

```

```

3875     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3876     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3877     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3878   }
3879   {
3880     \tl_set:Nn \CJKrmdefault { rm }
3881     \tl_set:Nn \CJKsfdefault { sf }
3882     \tl_set:Nn \CJKttdefault { tt }
3883   }
3884 }
3885 {
3886   \sys_if_engine_uptex:TF
3887   {
3888     \ctex_set_upfonts:nnnnnn
3889       {simsun.ttc} {simhei.ttf} {simkai.ttf}
3890     <windowsnew>      {msyh\l__ctex_msyh_suffix_tl} {msyhbdl__ctex_msyh_suffix_tl}
3891     <windowsold>      {simhei.ttf} {simhei.ttf}
3892       {simfang.ttf}
3893     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3894     \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
3895     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3896     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3897     <windowsnew>      \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
3898     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
3899     \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
3900     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
3901     \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
3902   }
3903   {
3904     <*windowsold>
3905     \setCJKmainfont
3906       [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3907     \setCJKsansfont { SimHei }
3908     \setCJKmonofont { FangSong_GB2312 }
3909     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3910     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3911     </windowsold>
3912     <*windowsnew>
3913     \setCJKmainfont
3914       [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3915     \setCJKsansfont
3916       [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3917     \setCJKmonofont { FangSong }
3918     \setCJKfamilyfont { zhkai } { KaiTi }
3919     \setCJKfamilyfont { zhfs } { FangSong }
3920     </windowsnew>
3921     \setCJKfamilyfont { zhsong } { SimSun }
3922     \setCJKfamilyfont { zhhei } { SimHei }
3923     \setCJKfamilyfont { zhli } { LiSu }
3924     \setCJKfamilyfont { zhyou } { YouYuan }
3925     <*windowsnew>
3926     \setCJKfamilyfont { zhyahei }
3927       [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3928     </windowsnew>
3929   }
3930 }
3931 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```

3932 <*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。

```

3933 \sys_if_engine_pdftex:TF
3934 {

```



```

3935 \sys_if_output_pdf:TF
3936 { \ctex_fontset_error:n { adobe } }
3937 {
3938   \ctex_zhmap_case:nnn
3939   {
3940     \setCJKmainfont
3941     [
3942       cmap = UniGB-UTF16-H ,
3943       BoldFont = AdobeHeitiStd-Regular.otf ,
3944       ItalicFont = AdobeKaitiStd-Regular.otf
3945     ] { AdobeSongStd-Light.otf }
3946     \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3947     \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3948     \setCJKfamilyfont { zhsong }
3949     [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3950     \setCJKfamilyfont { zhhei }
3951     [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3952     \setCJKfamilyfont { zhkai }
3953     [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3954     \setCJKfamilyfont { zhfs }
3955     [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3956     \ctex_punct_set:n { adobe }
3957     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3958     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3959     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3960     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3961     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3962   }
3963   {
3964     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3965     \ctex_punct_set:n { adobe }
3966     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3967     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3968     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3969   }
3970   { \ctex_fontset_error:n { adobe } }
3971 }
3972 }
3973 {
3974   \sys_if_engine_uptex:TF
3975   {
3976     \ctex_set_upfonts:nnnnnn
3977     {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
3978     {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
3979     {AdobeFangsongStd-Regular.otf}
3980     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3981     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
3982     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3983     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3984   }
3985   {
3986     \setCJKmainfont
3987     [
3988       BoldFont = AdobeHeitiStd-Regular ,
3989       ItalicFont = AdobeKaitiStd-Regular
3990     ] { AdobeSongStd-Light }
3991     \setCJKsansfont { AdobeHeitiStd-Regular}
3992     \setCJKmonofont { AdobeFangsongStd-Regular}
3993     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3994     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3995     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3996     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3997   }
3998 }
3999 </adobe>
4000 <*fandol>

```


14.15.4 ctex-fontset-fandol.def

```

4001 \sys_if_engine_pdftex:TF
4002 {
4003   \sys_if_output_pdf:TF
4004   { \ctex_fontset_error:n { fandol } }
4005   {
4006     \ctex_zhmap_case:nnn
4007     {
4008       \setCJKmainfont
4009       [
4010         cmap = UniGB-UTF16-H ,
4011         BoldFont = FandolSong-Bold.otf ,
4012         ItalicFont = FandolKai-Regular.otf
4013       ] { FandolSong-Regular.otf }
4014       \setCJKsansfont
4015       [
4016         cmap = UniGB-UTF16-H ,
4017         BoldFont = FandolHei-Bold.otf
4018       ] { FandolHei-Regular.otf }
4019       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4020       \setCJKfamilyfont { zhsong }
4021       [
4022         cmap = UniGB-UTF16-H ,
4023         BoldFont = FandolSong-Bold.otf
4024       ] { FandolSong-Regular.otf }
4025       \setCJKfamilyfont { zhhei }
4026       [
4027         cmap = UniGB-UTF16-H ,
4028         BoldFont = FandolHei-Bold.otf
4029       ] { FandolHei-Regular.otf }
4030       \setCJKfamilyfont { zhfs }
4031       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4032       \setCJKfamilyfont { zhkai }
4033       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4034       \ctex_punct_set:n { fandol }
4035       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4036       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4037       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4038       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4039       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4040       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4041     }
4042     {
4043       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4044       \ctex_punct_set:n { fandol }
4045       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4046       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4047       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4048     }
4049     { \ctex_fontset_error:n { fandol } }
4050   }
4051 }
4052 {
4053   \sys_if_engine_uptex:TF
4054   {
4055     \ctex_set_upfonts:nnnnnn
4056     {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4057     {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4058     {FandolFang-Regular.otf}
4059     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4060     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4061     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4062     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4063   }
4064   {
4065     \setCJKmainfont

```

```

4066      [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
4067      { FandolSong-Regular }
4068      \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4069      \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4070      \setCJKfamilyfont { zhsong }
4071      [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4072      \setCJKfamilyfont { zhhei }
4073      [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4074      \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4075      \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4076    }
4077  }
4078 </fandol>
4079 <*mac>

```

14.15.5 ctex-fontset-mac.def

OS X Mavericks (10.9) 预装的主要简体中文字体如下²²:

```

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

```

在 DVIPDFMx 下,可以通过下述方式使用华文宋体和华文楷体:

```

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

```

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

```

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

```

DVIPDFMx 不能生成 PDF,报下述错误:

²²<http://support.apple.com/kb/HT5944>

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

如果将 CMap 改为 UniGB-UTF16-H, 错误信息是

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

在 pdfTeX 下生成的 PDF 只有方框²³。华文细黑和华文仿宋的情况类似。这确认为 DVIPDFMx 的 bug²⁴。

```
4080 \sys_if_engine_pdftex:TF
4081 { \ctex_fontset_error:n { mac } }
4082 {
4083   \sys_if_engine_uptex:TF
4084   { \ctex_fontset_error:n { mac } }
4085   {
4086     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4087     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4088     \setCJKmonofont { STFangsong }
4089     \setCJKfamilyfont { zhsong } { STSong }
4090     \setCJKfamilyfont { zhhei } { STHeiti }
4091     \setCJKfamilyfont { zhfs } { STFangsong }
4092     \setCJKfamilyfont { zhkai } { STKaiti }
4093   }
4094 }
4095 </mac>
4096 <*founder>
```

14.15.6 ctex-fontset-founder.def

```
4097 \sys_if_engine_pdftex:TF
4098 {
4099   \ctex_zhmap_case:nnn
4100   {
4101     \setCJKmainfont
4102     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4103     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4104     \setCJKmonofont { FZFSK.TTF }
4105     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4106     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4107     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4108     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4109     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4110     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4111     \ctex_punct_set:n { founder }
4112     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4113     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4114     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4115     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4116     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4117     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4118     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4119   }
4120   {
4121     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4122     \ctex_punct_set:n { founder }
4123     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4124     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4125     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4126   }
4127   { \ctex_fontset_error:n { founder } }
```

²³<http://www.newsmth.net/bbscon.php?bid=460&id=312640>

²⁴<https://github.com/clerkma/ptex-ng/blob/master/texk/libdpx/cidtype2.c#L597>

```

4128 }
4129 {
4130   \sys_if_engine_uptex:TF
4131   {
4132     \ctex_set_upfonts:nnnnnn
4133     { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4134     { FZXH1K.TTF } { FZHTK.TTF }
4135     { FZFSK.TTF }
4136     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4137     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4138     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4139     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4140     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4141     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4142     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4143     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4144   }
4145   {
4146     \setCJKmainfont
4147     [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }

```

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

```

4148   \setCJKsansfont [ BoldFont = FZHei-B01 ]
4149   { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4150   \setCJKmonofont { FZFangSong-Z02 }
4151   \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4152   \setCJKfamilyfont { zhhei } { FZHei-B01 }
4153   \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4154   \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4155   \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4156   \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4157 }
4158 }
4159 </founder>
4160 <*ubuntu>

```

14.15.7 ctex-fontset-ubuntu.def

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

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

需要注意的是 uming.ttc 和 ukai.ttc 看起来像有四种字形的样子,但其实只有“令”和“骨”这区区两个字有新字形,其余都取旧字形²⁵。

```

4161 \sys_if_engine_pdftex:TF
4162 {
4163   \ctex_zhmap_case:nnn
4164   {
4165     \setCJKmainfont
4166     [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4167     \setCJKsansfont { wqy-zenhei.ttc }
4168     \setCJKmonofont { uming.ttc }
4169     \setCJKfamilyfont { zhsong } { uming.ttc }
4170     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4171     \setCJKfamilyfont { zhkai } { ukai.ttc }

```

²⁵<http://www.freedesktop.org/wiki/Software/CJKUnifonts/Download/>

```

4172 \ctex_punct_set:n { ubuntu }
4173 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4174 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4175 \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4176 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4177 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4178 }
4179 {
4180 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4181 \ctex_punct_set:n { ubuntu }
4182 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4183 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4184 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4185 }
4186 { \ctex_fontset_error:n { ubuntu } }
4187 }
4188 {
4189 \sys_if_engine_uptex:TF
4190 {
4191 \ctex_set_upfonts:nnnnnn
4192 { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4193 { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4194 { uming.ttc }
4195 \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4196 \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4197 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4198 }
4199 {
4200 \setCJKmainfont
4201 [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4202 \setCJKsansfont { WenQuanYi~Zen~Hei }
4203 \setCJKmonofont { AR~PL~UMing~CN }
4204 \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4205 \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4206 \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4207 }
4208 }
4209 </ubuntu>

```

14.15.8 中文字体命令

```

4210 <*/windows>
4211 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4212 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
4213 <!ubuntu>\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4214 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4215 <*windowsnew|windowsold|founder>
4216 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4217 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4218 </windowsnew|windowsold|founder>
4219 <windowsnew>\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4220 </!windows>
4221 </fontset>
4222 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4223 \begingroup\catcode61\catcode48\catcode32=10\relax%
4224 \catcode 35=6 % #
4225 \catcode123=1 % {
4226 \catcode125=2 % }
4227 \toks0{\endlinechar=\the\endlinechar\relax}%
4228 \toks2{\endlinechar=13 }%

```

```

4229 \def\x#1 #2 {%
4230   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4231   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4232 \x 13 5 % carriage return
4233 \x 32 10 % space
4234 \x 35 6 % #
4235 \x 40 12 % (
4236 \x 41 12 % )
4237 \x 45 12 % -
4238 \x 46 12 % .
4239 \x 47 12 % /
4240 \x 58 12 % :
4241 \x 60 12 % <
4242 \x 61 12 % =
4243 \x 64 11 % @
4244 \x 91 12 % [
4245 \x 93 12 % ]
4246 \x 123 1 % {
4247 \x 125 2 % }
4248 \edef\x#1{\endgroup%
4249   \edef\noexpand#1{%
4250     \the\toks0 %
4251     \let\noexpand\noexpand\noexpand#1%
4252     \noexpand\noexpand\noexpand\noexpand\undefined%
4253     \noexpand\noexpand\noexpand\noexpand\endinput}%
4254     \the\toks2}%
4255 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4256 \begingroup\expandafter\endgroup
4257 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4258 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4259   \expandafter\ifx\csname pdfoutput\endcsname\relax
4260     iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4261   \else ifctexpdf\fi
4262 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4263 \begingroup
4264 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4265   \long\def\x#1\ProvidesFile#2[#3]{%
4266     #1%
4267     \immediate\write-1{File: #2 #3}%
4268     \expandafter\xdef\csname ver@#2\endcsname{#3}}
4269   \expandafter\x%
4270 \fi
4271 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4272 <*windows>
4273 \ProvidesFile{zhwindowsfonts.tex}%
4274 [2016/05/16 v2.4.2 Windows font map loader for pdfTeX and DVIPDFx (CTEX)]
4275
4276 \ifzhmappdf
4277   \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4278   \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4279   \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4280   \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4281   \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4282   \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4283   \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4284
4285   \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4286   \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4287   \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4288   \pdfmapline{=unihei@Unicode@ <simhei.ttf}

```

```

4289 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4290 \pdfmapline{=unili@Unicode@ <simli.ttf}
4291 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4292
4293 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4294 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4295 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4296 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4297 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4298 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4299
4300 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4301 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4302 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4303 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4304 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4305 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4306
4307 \else
4308 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4309 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4310 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4311 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4312 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4313 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4314 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4315
4316 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4317 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4318 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4319 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4320 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4321 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4322 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4323
4324 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4325 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4326 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4327 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4328 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4329 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4330
4331 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4332 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4333 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4334 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4335 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4336 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4337
4338 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4339 <*adobe>
4340 \ProvidesFile{zhadobefonts.tex}%
4341 [2016/05/16 v2.4.2 Adobe font map loader for DVIPDFMx (CTEX)]
4342
4343 \ifzhmappdf
4344 %% pdfTeX does not support OTF fonts
4345
4346 \else
4347 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4348 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4349 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4350 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4351 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4352
4353 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}

```



```

4354 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4355 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4356 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4357 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4358
4359 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4360 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4361 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4362 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4363
4364 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4365 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4366 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4367 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4368
4369 </adobe>

```

14.15.9.3 zhfandolfonts.tex

```

4370 <*fandol>
4371 \ProvidesFile{zhfandolfonts.tex}%
4372 [2016/05/16 v2.4.2 Fandol font map loader for DVIPDFMx (CTEX)]
4373
4374 \ifzhmappdf
4375 %% pdfTeX does not support OTF fonts
4376
4377 \else
4378 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4379 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4380 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4381 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4382 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4383
4384 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4385 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4386 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4387 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4388 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4389
4390 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4391 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4392 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4393 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4394
4395 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4396 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4397 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4398 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4399
4400 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4401 <*founder>
4402 \ProvidesFile{zhfounderfonts.tex}%
4403 [2016/05/16 v2.4.2 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4404
4405 \ifzhmappdf
4406 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4407 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4408 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4409 \pdfmapline{=gbkhei@UGBK@ <FZHtk.TTF}
4410 \pdfmapline{=gbkfs@UGBK@ <FZFsk.TTF}
4411 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4412 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4413
4414 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4415 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}

```



```

4416 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4417 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4418 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4419 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4420 \pdfmapline{=uniyoud@Unicode@ <FZY1K.TTF}
4421
4422 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4423 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4424 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4425 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4426 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4427 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4428
4429 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4430 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4431 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4432 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4433 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4434 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4435
4436 \else
4437 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4438 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4439 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4440 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4441 \special{pdf:mapline gbkfsl@UGBK@ unicode FZFSK.TTF}
4442 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF}
4443 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4444
4445 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4446 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4447 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4448 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4449 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4450 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4451 \special{pdf:mapline uniyoud@Unicode@ unicode FZY1K.TTF}
4452
4453 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4454 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4455 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4456 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4457 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4458 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4459
4460 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4461 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4462 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4463 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4464 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4465 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4466
4467 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4468 <*ubuntu>
4469 \ProvidesFile{zhubuntufonts.tex}%
4470 [2016/05/16 v2.4.2 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4471
4472 \ifzhmappdf
4473 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4474 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4475 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4476 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4477 \pdfmapline{=gbkfsl@UGBK@ <uming.ttc}
4478 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4479
4480 \pdfmapline{=cyberb@Unicode@ <uming.ttc}

```

```

4481 \pdfmapline{=unisong@Unicode@ <uming.ttc>
4482 \pdfmapline{=unikai@Unicode@ <ukai.ttc>
4483 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc>
4484 \pdfmapline{=unifs@Unicode@ <uming.ttc>
4485 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc>
4486
4487 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc>
4488 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc>
4489 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc>
4490 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc>
4491 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc>
4492
4493 \pdfmapline{=unisongsl@Unicode@ <uming.ttc>
4494 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc>
4495 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc>
4496 \pdfmapline{=unifssl@Unicode@ <uming.ttc>
4497 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc>
4498
4499 \else
4500 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4501 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4502 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4503 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4504 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4505
4506 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4507 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4508 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4509 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4510 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4511
4512 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4513 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4514 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4515 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4516
4517 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4518 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4519 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4520 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4521
4522 </ubuntu>
4523 \fi
4524
4525 \ctex@zhmap@endinput
4526 </zhmap>

```

14.15.10 制作 spa 文件

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

```

4527 <*spa>
4528 <*macro>
4529 \input expl3-generic %
4530 \ExplSyntaxOn
4531 \sys_if_engine_xetex:F
4532 {
4533   \msg_new:nn { ctex } { xetex }
4534   { XeTeX~is~required~to~compile~this~document! }
4535   \msg_fatal:nn { ctex } { xetex }
4536 }

```

CJKpunct 定义的标点符号是:

‘ “ [] { } < » [] 【 —…、。：！？%） ） } > » » 】 ’ ” | |

注意顺序不能改变。

```

4537 \seq_new:N \c__ctex_punct_seq
4538 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4539 {
4540   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4541   "3008 , "300A , "3016 , "3010 ,
4542   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4543   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4544   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4545 }

```

`\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。

```

4546 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4547 {
4548   \iow_open:Nn \g__ctex_spa_iow {#1}
4549   \clist_map_inline:nn {#2}
4550     { \__ctex_write_family:nn ##1 }
4551   \iow_close:N \g__ctex_spa_iow
4552 }
4553 \iow_new:N \g__ctex_spa_iow
4554 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

(End definition for \ctex_make_spa:nn.)

4555 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4556 {
4557   \group_begin:
4558   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4559   \l__ctex_punct_font
4560   \clist_clear:N \l__ctex_punct_bounds_clist
4561   \seq_map_inline:Nn \c__ctex_punct_seq
4562     { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4563   \iow_now:Nx \g__ctex_spa_iow
4564   {
4565     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4566     { \l__ctex_punct_bounds_clist , , , }
4567   }
4568   \group_end:
4569 }
4570 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4571 {
4572   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4573     {
4574       \__ctex_calc_bounds:nn { \c_one } {#1} ,
4575       \__ctex_calc_bounds:nn { \c_three } {#1}
4576     }
4577 }
4578 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1em 的比值的一百倍。

```

4579 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4580 {
4581   \fp_eval:n
4582     {
4583       round
4584       (
4585         \dim_to_decimal_in_unit:nn
4586           { 100 \xetex_glyphbounds:D #1 ~ #2 }
4587           { 1 em }
4588       )
4589     }
4590 }

```

```

4591 \ExplSyntaxOff
4592 </macro>

```

下面是 C_T_EX 定义的一些字体。

```

4593 <*make>
4594 \input ctexspamacro %
4595
4596 \MAKESPA {ctexpunct.spa}
4597 {
4598   {adobezhsong}      {AdobeSongStd-Light} ,
4599   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4600   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4601   {adobezhfs}        {AdobeFangsongStd-Regular} ,
4602   {fandolzhsong}     {FandolSong} ,
4603   {fandolzhsongb}    {FandolSong-Bold} ,
4604   {fandolzhhei}      {FandolHei} ,
4605   {fandolzhheib}     {FandolHei-Bold} ,
4606   {fandolzhkai}      {FandolKai} ,
4607   {fandolzhfs}       {FandolFang} ,
4608   {founderzhsong}    {FZShuSong-Z01} ,
4609   {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
4610   {founderzhhei}     {FZHei-B01} ,
4611   {founderzhheil}    {FZXiHeiI-Z08} ,
4612   {founderzhkai}     {FZKai-Z03} ,
4613   {founderzhfs}      {FZFangSong-Z02} ,
4614   {founderzhli}      {FZLiShu-S01} ,
4615   {founderzhyou}     {FZXiYuan-M01} ,
4616   {founderzhyoub}    {FZZhunYuan-M02} ,
4617   {ubuntuzhsong}     {AR PL UMinG CN} ,
4618   {ubuntuzhhei}      {WenQuanYi Zen Hei} ,
4619   {ubuntuzhkai}      {AR PL UKai CN} ,
4620   {windowszhsong}    {SimSun} ,
4621   {windowszhhei}     {SimHei} ,
4622   {windowszhkai}     {KaiTi} ,
4623   {windowszhfs}      {FangSong} ,
4624   {windowszhli}      {LiSu} ,
4625   {windowszhyou}     {YouYuan} ,
4626   {windowszhyawei}    {Microsoft YaHei} ,
4627   {windowszhyaweib}  {Microsoft YaHei Bold}
4628 }
4629
4630 \primitive\end
4631 </make>
4632 </spa>

```

14.16 translator 宏包的中文字典

```

4633 <*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。

```

4634 <*theorem>
4635 \providetranslation{Comments}{评论}
4636 \providetranslation{comments}{评论}
4637 \providetranslation{Comment}{评论}
4638 \providetranslation{comment}{评论}
4639 \providetranslation{Corollaries}{推论}
4640 \providetranslation{corollaries}{推论}
4641 \providetranslation{Corollary}{推论}
4642 \providetranslation{corollary}{推论}
4643 \providetranslation{Definitions}{定义}
4644 \providetranslation{definitions}{定义}
4645 \providetranslation{Definition}{定义}
4646 \providetranslation{definition}{定义}
4647 \providetranslation{Examples}{例}

```

```

4648 \providetranslation{examples}{例}
4649 \providetranslation{Example}{例}
4650 \providetranslation{example}{例}
4651 \providetranslation{Exercises}{练习}
4652 \providetranslation{exercises}{练习}
4653 \providetranslation{Exercise}{练习}
4654 \providetranslation{exercise}{练习}
4655 \providetranslation{Facts}{事实}
4656 \providetranslation{facts}{事实}
4657 \providetranslation{Fact}{事实}
4658 \providetranslation{fact}{事实}
4659 \providetranslation{Key Lemmas}{关键引理}
4660 \providetranslation{key lemmas}{关键引理}
4661 \providetranslation{Key Lemma}{关键引理}
4662 \providetranslation{key lemma}{关键引理}
4663 \providetranslation{Key Observations}{关键观察}
4664 \providetranslation{key observations}{关键观察}
4665 \providetranslation{Key Observation}{关键观察}
4666 \providetranslation{key observation}{关键观察}
4667 \providetranslation{Lemmas}{引理}
4668 \providetranslation{lemmas}{引理}
4669 \providetranslation{Lemma}{引理}
4670 \providetranslation{lemma}{引理}
4671 \providetranslation{Main Theorems}{主要定理}
4672 \providetranslation{main theorems}{主要定理}
4673 \providetranslation{Main Theorem}{主要定理}
4674 \providetranslation{main theorem}{主要定理}
4675 \providetranslation{Observations}{观察}
4676 \providetranslation{observations}{观察}
4677 \providetranslation{Observation}{观察}
4678 \providetranslation{observation}{观察}
4679 \providetranslation{Problems}{问题}
4680 \providetranslation{problems}{问题}
4681 \providetranslation{Problem}{问题}
4682 \providetranslation{problem}{问题}
4683 \providetranslation{Proofs}{证明}
4684 \providetranslation{proofs}{证明}
4685 \providetranslation{Proof}{证明}
4686 \providetranslation{proof}{证明}
4687 \providetranslation{Proof Sketch}{证明提要}
4688 \providetranslation{Proof sketch}{证明提要}
4689 \providetranslation{proof sketch}{证明提要}
4690 \providetranslation{Proof Sketches}{证明提要}
4691 \providetranslation{Proof sketches}{证明提要}
4692 \providetranslation{proof sketches}{证明提要}
4693 \providetranslation{Sketch of Proof}{证明提要}
4694 \providetranslation{Sketch of Proofs}{证明提要}
4695 \providetranslation{Sketch of proof}{证明提要}
4696 \providetranslation{Sketch of proofs}{证明提要}
4697 \providetranslation{sketch of proof}{证明提要}
4698 \providetranslation{sketch of proofs}{证明提要}
4699 \providetranslation{Propositions}{命题}
4700 \providetranslation{propositions}{命题}
4701 \providetranslation{Proposition}{命题}
4702 \providetranslation{proposition}{命题}
4703 \providetranslation{Remarks}{注}
4704 \providetranslation{remarks}{注}
4705 \providetranslation{Remark}{注}
4706 \providetranslation{remark}{注}
4707 \providetranslation{Solutions}{解}
4708 \providetranslation{solutions}{解}
4709 \providetranslation{Solution}{解}
4710 \providetranslation{solution}{解}
4711 \providetranslation{Theorems}{定理}
4712 \providetranslation{theorems}{定理}
4713 \providetranslation{Theorem}{定理}
4714 \providetranslation{theorem}{定理}

```

```
4715 </theorem>
4716 </dict>
```

14.17 ctexcap 宏包

```
4717 <*ctexcap>
```

ctexcap 是过时宏包。

```
4718 \clist_new:N \l__ctex_ctexcap_options_clist
4719 \clist_set:Nx \l__ctex_ctexcap_options_clist
4720 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
4721 \msg_new:nnn { ctexcap } { deprecated }
4722 {
4723   Package~`ctexcap'~is~deprecated.\\
4724   Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
4725   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4726 }
4727 \msg_warning:nnx { ctexcap } { deprecated }
4728 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }
```

ctexcap 是默认打开 heading 选项的 ctex。

```
4729 \PassOptionsToPackage { heading = true } { ctexcap }
4730 \RequirePackageWithOptions { ctex }
4731 </ctexcap>
```

14.18 ctexhook 宏包

```
4732 <*ctexhook>
```

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```
\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4733 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4734 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
4735 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4736 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
4737 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4738 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4739 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4740 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4741 \cs_set_nopar:Npx \document
4742 {
4743   \CTEX@document@left@hook
4744   \exp_not:o { \document }
4745   \CTEX@document@right@hook
4746 }
4747 \tl_new:N \g__ctex_end_preamble_hook_tl
4748 \tl_new:N \g__ctex_after_end_preamble_hook_tl
```

(End definition for \ctex_at_end_preamble:n and \ctex_after_end_preamble:n.)

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。

```
4749 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4750 {
4751   \@ifpackageloaded {#1}
4752   {#2}
4753   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
4754 }
```

(End definition for \ctex_at_end_package:nn.)

```
\ctex_gadd_hook:Nn 给钩子附加内容。
\ctex_gadd_hook:cn
```

```

4755 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4756 {
4757   \tl_if_exist:NF #1 { \tl_new:N #1 }
4758   \tl_gput_right:Nn #1 {#2}
4759 }
4760 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

(End definition for \ctex_gadd_hook:Nn and \ctex_gadd_hook:cn.)

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

\ctex_package_end_hook:o

```

4761 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
4762 {
4763   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
4764   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
4765 }
4766 \cs_generate_variant:Nn \ctex_package_end_hook:n { o }

```

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

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

```

4767 \tl_put_left:Nn \@popfilename
4768 {
4769   \cs_if_eq:NNT \@currentx \pkgextension
4770   { \ctex_package_end_hook:o { \@currname } }
4771 }

```

4772 </ctexhook>

14.19 ctexpatch 宏包

4773 <*ctexpatch>

```

4774 \cs_if_exist:NF \str_new:N { \RequirePackage { l3str } }

```

\ctex_patch_cmd_once:NnnnTF 只进行第一次匹配进行替换。参数 #2 是宏重建时的 \catcode 设置。

```

4775 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
4776 {
4777   \ctex_patch_boot:NnnnTF \_ctex_patch_cmd:Nnnnnw #1
4778   { once } {#2} { \use_i:nn } { \use_ii:nn }
4779 }

```

(End definition for \ctex_patch_cmd_once:NnnnTF.)

\ctex_patch_cmd_all:NnnnTF 替换所有匹配到的文本。

```

4780 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
4781 {
4782   \ctex_patch_boot:NnnnTF \_ctex_patch_cmd:Nnnnnw #1
4783   { all } {#2} { \use_i:nn } { \use_ii:nn }
4784 }

```

(End definition for \ctex_patch_cmd_all:NnnnTF.)

\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。

```

4785 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
4786 {
4787   \ctex_patch_boot:NnnnTF \_ctex_patch_cmd:Nnnnnw #1
4788   { once }
4789   {
4790     \ExplSyntaxOff
4791     \char_set_catcode_letter:n { 64 }
4792   }

```

```

4793     { }
4794     { \ctex_patch_failure:N #1 }
4795   }
4796   \cs_new_protected:Npn \ctex_patch_failure:N #1
4797   { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
4798   \msg_new:nnn { ctex } { patch-failure }
4799   { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

(End definition for \ctex_patch_cmd:Nnn.)

\ctex_preto_cmd:NnnTF 在宏的原本定义前面增加钩子。

```

4800 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
4801 {
4802   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
4803   { left } {#2} { \use_i:nn } { \use_ii:nn }
4804 }

```

(End definition for \ctex_preto_cmd:NnnTF.)

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```

4805 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
4806 {
4807   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
4808   { right } {#2} { \use_i:nn } { \use_ii:nn }
4809 }

```

(End definition for \ctex_appto_cmd:NnnTF.)

\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。

```

4810 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
4811 {
4812   \tl_set:Nn \__ctex_patch_true:w {#5}
4813   \tl_set:Nn \__ctex_patch_false:w {#6}
4814   \group_begin:
4815     \char_set_catcode_other:n { 35 }
4816     \ctex_parse_name:NN #1 #2 {#3} {#4}
4817   }
4818   \tl_new:N \__ctex_patch_true:w
4819   \tl_new:N \__ctex_patch_false:w

```

(End definition for \ctex_patch_boot:NNnnTF.)

\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。

```

4820 \cs_new_protected:Npn \ctex_parse_name:NN #1#2

```



```

4821 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
4822 \group_begin:
4823 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
4824 {
4825   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
4826   {
4827     \bool_if:nTF
4828     {
4829       \cs_if_exist_p:c { ##3 ~ } ||
4830       \cs_if_exist_p:c { #1##3 }
4831     }
4832     {
4833       \group_begin:
4834       \use:x
4835       {
4836         \group_end:
4837         \__ctex_parse_name:nNNNnN
4838         { \token_get_replacement_spec:N ##2 }
4839         \exp_not:N ##2
4840         \exp_not:c { ##3 ~ }
4841         \exp_not:c { #1##3 }
4842       } {##3} ##1
4843     }
4844     { ##1##2 }
4845   }
4846   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
4847   {
4848     \exp_args:Nc ##6
4849     {
4850       \str_case:nnTF {##1}
4851       {
4852         { \protect ##3 } { }
4853         { \x@protect ##2 \protect ##3 } { }
4854       }
4855       {
4856         \str_if_eq_x:nnTF
4857         { \exp_not:n { #1@protected@ ##3 #1##3 } }
4858         {
4859           \exp_last_unbraced:Nf \__ctex_parse_name:w
4860           \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
4861         }
4862         { #1##5 ~ } { ##5 ~ }
4863       }
4864       {
4865         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
4866         {
4867           { #1@protected@ ##2 ##4 } { }
4868           { #1@ ##4 } { }
4869         }
4870         { #1##5 } {##5}
4871       }
4872     }
4873   }
4874   \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
4875 }
4876 \use:x
4877 {
4878   \__ctex_tmp:w
4879   { \c_backslash_str }
4880   { \c_left_brace_str }
4881   { \tl_to_str:n { testopt } }
4882 }
4883 \group_end:
4884 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

(End definition for \ctex_parse_name:NN.)

```

\l__ctex_prefix_str 分别保存宏的 \meaning 中的前缀、参数文本和替换文本。
\l__ctex_parameter_str
\l__ctex_replacement_str
4885 \str_new:N \l__ctex_prefix_str
4886 \str_new:N \l__ctex_parameter_str
4887 \str_new:N \l__ctex_replacement_str

(End definition for \l__ctex_prefix_str, \l__ctex_parameter_str, and \l__ctex_replacement_str.)

```

```

\ctex_get_macro_meaning:NTF 解构待补丁宏的 \meaning。若命令不是宏，则走向 false 分支。
\__ctex_get_macro_meaning:w
4888 \group_begin:
4889 \cs_set_protected:Npn \__ctex_tmp:w #1
4890 {
4891   \prg_new_protected_conditional:Npnn
4892   \ctex_get_macro_meaning:N ##1 { TF }
4893   {
4894     \exp_after:wN \__ctex_get_macro_meaning:w
4895     \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
4896   }
4897   \cs_new_protected:Npn \__ctex_get_macro_meaning:w
4898   ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
4899   {
4900     \tl_if_empty:nTF { ##4 }
4901     { \prg_return_false: }
4902     {
4903       \str_set:Nn \l__ctex_prefix_str { ##1 }
4904       \str_set:Nn \l__ctex_parameter_str { ##2 }
4905       \str_set:Nn \l__ctex_replacement_str { ##3 }
4906       \prg_return_true:
4907     }
4908   }
4909 }
4910 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
4911 \group_end:

(End definition for \ctex_get_macro_meaning:NTF.)

```

```

\ctex_if_rescanable:NnTF 检查宏是否可以重建。
4912 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
4913 {
4914   \ctex_get_macro_meaning:NTF #1
4915   {
4916     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
4917     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
4918   }
4919   {#4}
4920 }
4921 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

(End definition for \ctex_if_rescanable:NnTF.)

```

```

\__ctex_patch_rebuild:Nn 使用 \tl_rescan:nn 来重新记号化 \meaning 字符串。
4922 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
4923 {
4924   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
4925   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
4926   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
4927   \use:x
4928   {
4929     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
4930     \exp_not:o { \l__ctex_parameter_tl }
4931     { \exp_not:o { \l__ctex_replacement_tl } }
4932   }
4933 }
4934 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3

```

```

4935 {
4936   \str_if_empty:NTF #2
4937     { \tl_clear:N #1 }
4938     { \tl_set_rescan:Nno #1 {#3} {#2} }
4939 }
4940 \tl_new:N \l__ctex_prefix_tl
4941 \tl_new:N \l__ctex_parameter_tl
4942 \tl_new:N \l__ctex_replacement_tl

```

(End definition for __ctex_patch_rebuild:Nn.)

__ctex_patch_cmd:Nnnnw 对宏的替换文本进行字符串替换,然后重建。

```

4943 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
4944 {
4945   \group_end:
4946   \ctex_if_rescanable:NnTF #1 {#3}
4947   {
4948     \use:x
4949     {
4950       \__ctex_patch_replace:nnnTF {#2}
4951       { \tl_to_str:n {#4} }
4952       { \tl_to_str:n {#5} }
4953     }
4954     {
4955       \__ctex_patch_rebuild:Nn #1 {#3}
4956       \__ctex_patch_true:w
4957     }
4958     { \__ctex_patch_false:w }
4959   }
4960   { \__ctex_patch_false:w }
4961 }

```

(End definition for __ctex_patch_cmd:Nnnnw.)

__ctex_patch_replace:nnnTF 替换前先检查原文本是否存在。

```

4962 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
4963 {
4964   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
4965   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
4966 }

```

(End definition for __ctex_patch_replace:nnnTF.)

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

4967 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
4968 {
4969   \group_end:
4970   \ctex_get_macro_meaning:NTF #1
4971   {
4972     \str_if_empty:NTF \l__ctex_parameter_str
4973     { \__ctex_hookto_cmd_parameterless:Nnnnw }
4974     { \__ctex_hookto_cmd_parameter:Nnnnw }
4975     #1 {#2} {#3} {#4}
4976   }
4977   { \__ctex_patch_false:w }
4978 }

```

(End definition for __ctex_hookto_cmd:Nnnnw.)

__ctex_hookto_cmd_parameterless:Nnnnw 如果宏没有参数,可以直接进行附加操作。注意保持宏的前缀。

```

4979 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
4980 {

```

```

4981 \str_if_empty:NF \l__ctex_prefix_str
4982 { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
4983 \tex_edef:D #1
4984 {
4985   \use:c { __ctex_ #2 _hook_aux:nn }
4986   { \exp_not:o {#1} }
4987   { \exp_not:n {#4} }
4988 }
4989 \__ctex_patch_true:w
4990 }
4991 \cs_generate_variant:Nn \tl_rescan:nn { no }
4992 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
4993 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

(End definition for __ctex_hookto_cmd_parameterless:Nnnnw.)

__ctex_hookto_cmd_parameter:Nnnnw 如果宏有参数,需要在字符串中进行附加,然后再重建。

```

4994 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
4995 {
4996   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
4997   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
4998   {
4999     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5000     \__ctex_patch_rebuild:Nn #1 {#3}
5001     \__ctex_patch_true:w
5002   }
5003   { \__ctex_patch_false:w }
5004 }

```

(End definition for __ctex_hookto_cmd_parameter:Nnnnw.)

5005 </ctexpatch>

版本历史

v2.0

General: c5size, cs4size 是过时选项。	36
captiondelimiter 是过时选项。	110
fancyhdr 成为过时选项, 原选项功能总是打开。	39
fntef 成为过时选项, 原选项功能总是打开。	39
hyperref 成为过时选项, 原选项功能总是打开。	39
indent, noindent 是过时选项。	37
nofonts, adobefonts, winfonts 是过时选项。	37
nopunct 是过时选项。	38
nospace 是过时选项。	38
nozhmap 是过时选项。	38
punct 选项可以设置标点格式。	38
ctex 宏包新增 heading 选项。	39
\CTEXindent, \CTEXnoindent 是过时命令。	81
\CTEXsetup, \CTEXoptions 是过时命令。	14, 41
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; CTExfilltwosides 是过时环 境。	74
标题设置新增 pagestyle 选项。	85
调整 \footnotesep 的大小, 以适合行距的变化。	109
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号 选项。	40
将标题汉化功能加入 ctex.sty。	82
将中文字号功能提取到可以独立使用的 ctexsize。 . . .	106
解决 etoolbox 与 breqn 关于 \end 的冲突。	110
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	74
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jan 宏包支持 Lua _{La} TeX。	48
新增 autoindent 选项。	36
新增 fontset 选项。	37
新增 linespread 选项。	36
新增 linestretch 选项。	79
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。 .	39
新增 zhmCJK 支持选项。	38
新增 zihao 选项。	36
新增统一设置接口 \ctexset。	41
应用 \LaTeX 3 重新整理代码。	1
中文字号不再采用近似值。	106
自动检测操作系统, 载入对应的字体配置。	71
\CJK@surr: 解决与 \nouppercase 的冲突。	44

v2.0.1

General: 修复 10pt、11pt 等选项无效的问题。	40
---	----

v2.0.2

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。	105
---	-----

v2.1

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。	84
nameformat 可以接受章节名字为参数。	84
不依赖 ifpdf 宏包。	34
不再设置 hyperref 宏包的 colorlinks 选项。	72
给 pdf _{La} TeX 下的非 UTF8 编码 CJK 字体族加上 CMap。 .	42
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。	84
修复 ctexbook 和 ctexrep 类的中文 part/number 选项 初值为空的错误。	100
\ctex_if_macosx:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	72
hyperref: 补充定义 \hypersetup。	40

v2.2

General: before skip 和 after skip 选项的符号不再有特 殊意义。	84
before skip、after skip 和 indent 选项支持表达式。 . .	86
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/before skip 和 part/after skip 选项有意义。	86
给 enumitem 宏包注册 \chinese 和 \zhnum。	82
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。 .	130
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。	100
删去 etoolbox 与 breqn 的兼容补丁。	110
新的标题格式选项 afterindent。	84
新的标题格式选项 numbering。	84
新的标题格式选项 runin。	84
新增子宏包 ctexpatch 实现给宏打补丁功能。	131

v2.3

General: .value_required: 和 .value_forbidden: 已 过时。	36
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。 .	34
更新 LuaTeX-jan 支持 (20150922.0)。	48
更新 unicode-math 宏包补丁。	48
兼容 titleps 宏包。	95
修复 nameformat 作用域问题。	84
与 \LaTeX 3 (2015/12/20) 同步。	32

v2.4

General: 初步支持 up _{La} TeX。	68
加强 beamer 宏包支持。	1
提供 translator 宏包的中文定理名称翻译。	128
提供 up _{La} TeX 的 NFSS 字体定义。	112
正确更新 CJK 包的 \CJKfamilydefault。	70
正确设置 up _{La} TeX 下字体命令。	69
\ctex_parse_name:NN: 修复宏名解析错误。	132

v2.4.1

General: beamer 不调整默认行距。	105
beamer 不调整默认字体大小。	105
使用 bootfont.bin 判断 Windows XP 以避免权限问 题。	114
随字体更新 up _{La} TeX 的 \xkanjiskip。	77
新的标题格式选项 part/fixbefore skip 和 chapter/fixbefore skip。	85
正确更新 up _{La} TeX 的 \CJKfamilydefault。	70
\ccwd: 正确设置 up _{La} TeX 下的 \ccwd。	75
\chinese: 支持 \pagenumbering。	82
zihao: 不允许无参 zihao 选项。	36

v2.4.2

General: 恢复 luatexja 对 \emshape 和 \eminnershape 的 重定义。	49
\em: 兼容 up _{La} TeX 2016/05/07u00 的定义。	69

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	
_	2965, 2969, 2985, 2989
\#	491, 2886, 2932
.../afterindent	20
.../aftername	18
.../aftername+	18
.../afterskip	21
.../aftertitle	19
.../aftertitle+	19
.../beforeskip	20
.../format	16
.../format+	16
.../indent	21
.../name	15
.../nameformat	16
.../nameformat+	16
.../number	15
.../numberformat	18
.../numberformat+	18
.../titleformat	18
.../titleformat+	18
\/	1631
\\	15, 33, 34, 102, 145, 147, 149, 205, 215, 677, 1122, 1204, 2259, 3330, 3331, 3444, 3445, 3665, 3695, 4723, 4724, 4725, 4799
\{	1133, 1134, 4725
\}	1133, 1134, 4725
10pt	8, <u>40</u>
11pt	8, <u>40</u>
12pt	8, <u>40</u>
A	
\abovedisplayshortskip	3535, 3543, 3554, 3574, 3582, 3593
\abovedisplayskip	3534, 3537, 3542, 3549, 3553, 3560, 3573, 3576, 3581, 3588, 3592, 3599
\abstractname	2243
abstractname	12
\addCJKfontfeature	1204, 1227
\addCJKfontfeatures	<u>59</u> , 1222, 1227
\addcontentsline	2465, 2466, 2469, 2472, 2495, 2496, 2499, 2502, 2576, 2581, 2584, 2587, 2693, 2709
\AddEnumerateCounter	2203, 2205, 2206, 2207
\addpenalty	2660, 2661
\addtocontents	2590, 2591
\addvspace	2428, 2429, 2590, 2591, 2660, 2661, 3001, 3011, 3019, 3029, 3037, 3047
adobefonts	28
\advance	2417, 2418
\algorithmname	2250, 2266
algorithmname	12
AlternateFont	25, <u>61</u>
\appendix	<u>93</u> , 2801, 2802
appendix/name	22, <u>93</u>
appendix/number	22, <u>93</u>
appendix/numbering	21, <u>93</u>
\appendixname	2245, 3315
appendixname	12
\arabic	3220
\AtBeginDocument	579, 1620
\AtBeginDvi	449
\AtBeginShipoutFirst	451, 1831, 1847
\AtBeginUTFCommand	721
\AtEndOfClass	135
\AtEndOfPackage	136
\AtEndUTFCommand	727
autoindent	10, <u>36</u> , <u>78</u>
B	
backslash commands:	
\c_backslash_str	4879
\baselinestretch	2113
\begin	3007, 3025, 3043
\begingroup	2684, 2723, 2753, 2997, 3015, 3033, 4223, 4256, 4263
\belowdisplayshortskip	3536, 3544, 3555, 3575, 3583, 3594
\belowdisplayskip	3537, 3549, 3560, 3576, 3588, 3599
\bfdefault	985, 991, 1317, 1324
\bfseries	2478, 2481, 2509, 2512, 2523, 2536, 2611, 2617, 2635, 3108, 3110, 3114, 3126, 3127, 3130, 3152, 3154, 3159, 3176, 3180, 3201, 3234, 3250, 3263
\bibname	2248, 2251, 2259, 2267, 2273, 2274
bibname	12
bool commands:	
\bool_gset_false:N	265, 270
\bool_gset_true:N	260
\bool_if:cTF	2852, 2862, 2873
\bool_if:NT	1162
\bool_if:NTF	70, 3334
\bool_if:nTF	4827
\bool_new:N	131, 1199
\bool_set_false:N	1174
\bool_set_true:N	1192
box commands:	
\box_ht:N	3615
\box_new:N	61
\box_wd:N	1919, 2928
C	
c5size	28
cap	28
captiondelimiter	29, 110
\catcode	4223, 4224, 4225, 4226, 4230, 4231
CCT	28

- CCTfont 28
- \ccwd 24, 75, 115, 188, 1919, 1923, 1926, 1928, 1965,
1966, 1967, 2064, 2102, 2121, 2122, 2129, 2156, 2157,
2158, 2161, 2163, 2164, 2175, 2186, 2187, 3632, 3633, 3634
- \centering 2506, 2534, 2998,
3016, 3034, 3114, 3125, 3130, 3134, 3159, 3180, 3188, 3222
- \chapter 89, 2558, 3340
- chapter/beforeskip 30
- chapter/fixbeforeskip 20, 85
- chapter/numbering 15
- chapter/pagestyle 21, 85
- \chaptermark 2589
- \chaptername 3149
- char commands:
- \char_generate:n 749
- \char_set_catcode:n 73, 552, 553
- \char_set_catcode_active:N 1352
- \char_set_catcode_letter:n 67, 1623, 2946, 4791
- \char_set_catcode_math_superscript:n 547
- \char_set_catcode_other:N 491, 918, 1335, 2886, 2932
- \char_set_catcode_other:n 546, 4815
- \char_value_catcode:n 73, 552, 553
- CharRange 25, 61
- \Chinese 29, 2199
- \chinese 24, 82, 2197, 2199, 2207, 3101, 3158
- \CJK 1884
- \CJKbold
3745, 3746, 3756, 3757, 3771, 3772, 3774, 3775, 3777,
3778, 3782, 3783, 3785, 3786, 3788, 3789, 3801, 3802,
3804, 3805, 3807, 3808, 3812, 3813, 3815, 3816, 3818, 3819
- \CJKfamily
. 57, 526, 530, 534, 538, 1059, 1081, 1261, 1265, 1269,
1273, 1421, 1642, 4211, 4212, 4213, 4214, 4216, 4217, 4219
- \CJKfamilydefault
... 520, 521, 538, 594, 1255, 1256, 1273, 1281, 1283,
1287, 1294, 1307, 1638, 1647, 1721, 1725, 1733, 1746, 1748
- \CJKfilltwosides 1892
- \CJKfontspec 59, 1217
- \CJKglue 1918, 1932, 1949, 1962
- \CJKhook 46, 609
- \CJKnormal 3738, 3739, 3740, 3741, 3742, 3743, 3744,
3749, 3750, 3751, 3752, 3753, 3754, 3755, 3770, 3773,
3776, 3781, 3784, 3787, 3800, 3803, 3806, 3811, 3814, 3817
- \CJKpunctmapfamily
. 642, 643, 644, 645, 646, 647, 651, 652, 653, 654, 655, 656
- \CJKrmdefault 514,
521, 522, 526, 1129, 1140, 1230, 1256, 1257, 1261, 1283,
1286, 1287, 1635, 1638, 1639, 1729, 3867, 3869, 3870,
3875, 3876, 3877, 3880, 3957, 3960, 3961, 3966, 3967,
3968, 4035, 4038, 4039, 4045, 4046, 4047, 4112, 4115,
4116, 4123, 4124, 4125, 4173, 4176, 4177, 4182, 4183, 4184
- \CJKsfdefault 515, 523,
530, 1130, 1141, 1236, 1258, 1265, 1636, 1640, 1730,
3851, 3857, 3858, 3881, 3958, 4036, 4040, 4113, 4117, 4174
- \CJKtilde 595
- \CJKttdefault 516, 524, 534, 1131, 1142, 1241, 1259,
1269, 1637, 1641, 1731, 3868, 3882, 3959, 4037, 4114, 4175
- clearalternatefont 26, 65
- \cleardoublepage 2440, 2558
- \clearpage 2442, 2558
- clist commands:
- \clist_clear:N 1006, 4560
- \clist_concat:NNN 1028
- \clist_const:Nn 622
- \clist_gput_right:Nn 382, 404, 406
- \clist_gput_right:No 394
- \clist_gset:Nn 1247
- \clist_if_empty:NT 1340, 1599
- \clist_if_empty:NTF 1359
- \clist_map_break:n 2831, 3325
- \clist_map_function:nN 1550, 1564
- \clist_map_function:xN 1516, 1518
- \clist_map_inline:Nn 612, 1452
- \clist_map_inline:nn
. 367, 634, 1557, 1876, 1902, 2826, 3322, 3462, 3497, 4549
- \clist_new:N 138, 1025, 1248, 1394, 4578, 4718
- \clist_put_left:Nn 3391, 3417
- \clist_put_left:Nx 1029
- \clist_put_right:Nn 1191
- \clist_put_right:Nx 1440, 4572
- \clist_set:Nn 1377, 1378
- \clist_set:No 1010
- \clist_set:Nx 4719
- \clist_use:Nn 4728
- \clubpenalty 2752, 2759
- colon commands:
- \c_colon_str 943, 944, 945
- \contentsname 2238
- contentsname 11
- continuation 12
- cs commands:
- \cs:w 3072
- \cs_end: 3072
- \cs_generate_variant:Nn
..... 93, 467, 479, 766, 970, 1093, 1107, 1180, 1200,
1201, 1373, 1410, 1512, 1550, 1558, 4760, 4766, 4884, 4991
- \cs_gset_eq:cc 1167, 1170
- \cs_gset_eq:NN 161, 162, 163,
261, 266, 271, 309, 311, 377, 381, 499, 813, 817, 3397, 3424
- \cs_gset_nopar:Npx 814
- \cs_gset_protected_nopar:cpx 1055, 2780
- \cs_gset_protected_nopar:Npn 754, 2411, 2802
- \cs_gset_protected_nopar:Npx 852, 1414, 1508
- \cs_if_eq:cNF 954
- \cs_if_eq:NNT 4769
- \cs_if_eq:NNTF 4917, 4997
- \cs_if_exist:cF 808, 850, 888
- \cs_if_exist:cTF 3059
- \cs_if_exist:NF 358, 476, 820, 1413, 1505, 1983, 3342, 4774
- \cs_if_exist:NTF 44, 3340
- \cs_if_exist_p:c 4829, 4830

- \cs_if_exist_use:c 1077, 1078
- \cs_if_exist_use:cF 4, 791
- \cs_if_exist_use:cT 1527, 4763
- \cs_if_exist_use:cTF 1099
- \cs_if_free:cF 461, 614, 629, 1164, 2845, 2911, 2916
- \cs_if_free:cTF 1316
- \cs_if_free:Nf 471, 2091, 2203, 2920
- \cs_if_free:Nf 2077
- \cs_new:Npn
110, 117, 124, 909, 915, 919, 1002, 1080, 2870, 4874, 4992
- \cs_new_eq:cc 1911
- \cs_new_eq:cN 6, 7, 8, 9, 20, 23, 1897
- \cs_new_eq:Nc 1974
- \cs_new_eq:NN 95, 106, 107, 108, 133, 134,
152, 359, 468, 599, 660, 829, 854, 930, 961, 1227, 1233,
1742, 1898, 1899, 1962, 1973, 2198, 2199, 2200, 2772,
2773, 2774, 2775, 2801, 3077, 3338, 3343, 4554, 4921, 4993
- \cs_new_nopar:Npn
869, 1072, 1125, 1136, 2197, 2909, 2914, 4579
- \cs_new_protected:Npn 62,
747, 752, 1003, 1022, 1338, 1429, 1443, 1594, 1968,
2378, 2922, 3517, 3523, 4555, 4733, 4735, 4749, 4755,
4775, 4780, 4785, 4796, 4800, 4805, 4810, 4820, 4825,
4846, 4897, 4912, 4922, 4934, 4943, 4962, 4967, 4979, 4994
- \cs_new_protected_nopar:cpx 1881
- \cs_new_protected_nopar:Npn
77, 79, 135, 136, 447, 459,
469, 474, 480, 512, 542, 559, 601, 607, 610, 627, 632,
640, 649, 658, 731, 789, 798, 811, 816, 818, 830, 844,
871, 886, 899, 923, 932, 949, 963, 980, 1026, 1034, 1053,
1073, 1083, 1108, 1146, 1158, 1160, 1182, 1184, 1277,
1302, 1311, 1357, 1368, 1395, 1404, 1411, 1417, 1437,
1450, 1469, 1483, 1490, 1502, 1522, 1538, 1556, 1559,
1561, 1574, 1586, 1628, 1660, 1673, 1687, 1719, 1751,
1769, 1784, 1800, 1811, 1815, 1854, 1861, 1865, 1888,
1894, 1915, 1929, 1964, 1998, 2076, 2078, 2103, 2138,
2148, 2153, 2302, 2332, 2409, 2413, 2765, 2818, 2838,
2843, 2848, 2858, 2868, 2887, 3055, 3064, 3069, 3435,
3450, 3482, 3487, 3637, 3641, 4546, 4570, 4737, 4739, 4761
- \cs_new_protected_nopar:Npx 484, 1213
- \cs_set:Npn 729, 730, 1336
- \cs_set_eq:cc 616, 630
- \cs_set_eq:cN 1979
- \cs_set_eq:Nc 2767, 2768, 2769, 2770
- \cs_set_eq:NN 109, 189, 194, 199,
539, 557, 577, 604, 608, 769, 772, 823, 824, 931, 962,
1000, 1059, 1274, 1413, 1421, 1431, 1432, 1433, 1505,
1506, 1507, 1631, 1724, 2214, 2219, 2222, 2227, 2822, 2832
- \cs_set_nopar:cpx 1912
- \cs_set_nopar:Npn 1659, 1695
- \cs_set_nopar:Npx 2860, 4741
- \cs_set_protected:Npn 734, 4823, 4889
- \cs_set_protected_nopar:Npn 603, 773, 1932, 3524
- \cs_set_protected_nopar:Npx 1970
- \cs_to_str:N 54, 1212, 4821
- \cs_undefine:c 1038, 1039, 1043, 1044, 4764
- \cs_undefine:N 1434
- cs4size 28
- \csname 2667, 2668,
2673, 2689, 2690, 2692, 2695, 2696, 2697, 2706, 2707,
2708, 2711, 2712, 2713, 4255, 4257, 4258, 4259, 4264, 4268
- \CTeX 24, 109, 3618, 3621
- ctex commands:
- \ctex_add_cmap:cn 472
- \ctex_add_cmap:n 42, 464, 469
- \ctex_add_cmap:Nn 474, 479
- \ctex_add_to_selectfont:n . 76, 1968, 1989, 2017, 2090
- \ctex_after_end_preamble:n 34, 109, 130, 4735
- \g__ctex_after_end_preamble_hook_tl 4736, 4740, 4748
- \ctex_appto_cmd:NnnTF 132, 495, 1622, 4805
- \ctex_assign_heading_name:nn ... 85, 2338, 2375, 2794
- __ctex_assign_heading_name:nnn 85, 2377, 2378
- \ctex_at_end:n 35, 135, 136, 293, 356, 1962, 3704
- \ctex_at_end_package:nn
130, 131, 450, 498, 767, 778, 1825, 1827, 1845,
2201, 2410, 2878, 2919, 2921, 2943, 2973, 3083, 3620, 4749
- \ctex_at_end_preamble:n
130, 361, 584, 591, 1276, 1648, 4733
- \ctex_auto_ignorespaces: 46, 599, 608, 2027
- \l__ctex_autoindent_tl
35, 132, 188, 193, 198, 2064, 2068, 2071, 2140, 2143
- __ctex_calc_bounds:nn 4574, 4575, 4579
- __ctex_ccglue: 1949, 1962
- \l__ctex_ccglue_skip 1933, 1937,
1940, 1943, 1945, 1953, 1957, 2114, 2125, 2134, 2170, 2178
- \ctex_CJK_input:n 44, 542, 557
- \c__ctex_class_tl
84, 86, 409, 414, 419, 424, 3338, 3344, 3346, 3347, 3349
- __ctex_clear_fntef_color:n ... 1854, 1861, 1865, 1880
- \c__ctex_cmap_encoding_seq 42, 456, 457, 463
- \l__ctex_ctexcap_options_clist 4718, 4719, 4728
- \ctex_declare_math_sizes:nnnn 107, 3482, 3516
- __ctex_def_heading_keys:n 84, 2332, 2401
- \ctex_default_pt:n 34, 113, 117
- __ctex_default_pt:w 119, 124
- \ctex_detected_platform: 71, 1751, 1769, 1784, 3645, 3654
- \l__ctex_encoding_tl 35, 128, 129, 221,
222, 505, 508, 586, 594, 1821, 2053, 2195, 3379, 3406, 3420
- \g__ctex_end_preamble_hook_tl 4734, 4738, 4747
- \c__ctex_engine_file_str 32, 50, 2057
- \c__ctex_engine_str 32, 43, 48, 51, 53
- \ctex_family_cmap:nn 42, 459, 467
- \ctex_family_cmap:xn 468
- \l__ctex_family_default_init_tl
71, 1721, 1741, 1743, 1748
- __ctex_family_default_wrap:n 1724, 1742, 1745
- \ctex_file_input:n 33, 77, 397,
661, 2057, 3349, 3384, 3389, 3394, 3395, 3410, 3415,
3421, 3422, 3526, 3528, 3646, 3650, 3658, 3704, 3829, 3830
- \ctex_file_wrapper:nnn 33, 62, 78, 81, 544
- __ctex_fix_varioref_label:n 99
- \ctex_fix_varioref_label:n 3067, 3069

`\g_ctex_font_size_flag` 36, 152, 161, 162, 163, 377, 381, 403, 3396, 3397, 3423, 3424, 3525
`\c_ctex_font_size_prop` 106, 3437, 3448, 3454, 3492, 3519
`\c_ctex_font_size_seq` 3446, 3449, 3460
`\l_ctex_font_size_tl` 3437, 3438, 3484, 3485, 3519, 3520
`\ctex_fontset_error:n` 110, 3637, 3936, 3970, 4004, 4049, 4081, 4084, 4127, 4186
`\g_ctex_fontset_tl` 225, 1753, 1771, 1796, 1803, 1804, 3643, 3646, 3649, 3650, 3656, 3658, 3678, 3680, 3685, 3702
`\ctex_gadd_hook:cn` 130, 4753
`\ctex_gadd_hook:Nn` 130, 609, 4755, 4760
`__ctex_get_font_sizes:Nn` 107, 3484, 3487
`\ctex_get_macro_meaning:N` 4892
`\ctex_get_macro_meaning:NTF` 134, 4914, 4970
`__ctex_get_macro_meaning:w` 134, 4894, 4897
`\l_ctex_heading_bool` 304, 3334
`\c_ctex_headings_seq` 84, 2292, 2294, 2295, 2296, 2299, 2398, 3066
`__ctex_hookto_cmd:Nnnnw` 135, 4802, 4807, 4967
`__ctex_hookto_cmd_parameter:Nnnnw` .. 136, 4974, 4994
`__ctex_hookto_cmd_parameterless:Nnnnw` 135, 4973, 4979
`\ctex_hypersetup:n` 72, 72, 359, 1811, 1815, 1820, 1823, 1838, 1841, 1844
`\ctex_if_autoindent_touched:F` 35, 133, 189, 194, 199, 3376
`\ctex_if_ccglue_touched:` 1946
`\ctex_if_ccglue_touched:TF` 76, 2109
`\ctex_if_ccglue_touched_p:` 76
`\ctex_if_macosx:TF` 72, 1791, 1792, 1794, 1800
`\ctex_if_preamble:TF` 34, 108, 109, 3673
`\ctex_if_rescanable:NnTF` 134, 4912, 4946
`\ctex_ignorespaces_case:N` .. 46, 601, 2024, 2027, 2030
`__ctex_initial_heading:n` 84, 2302, 2400
`__ctex_left_hook_aux:nn` 4992
`\l_ctex_line_spread_fp` 109, 181, 3398, 3399, 3425, 3426, 3611, 3613
`\l_ctex_line_stretch_tl` 2096, 2101, 2102, 2106
`\ctex_load_fontset:` 110, 3641, 3662, 3681, 3703
`\ctex_load_zhmap:nnnn` 43, 512, 519, 3873, 3964, 4043, 4121, 4180
`\l_ctex_ltj_add_alternate_bool` 1162, 1174, 1192, 1199
`\ctex_ltj_add_font_features:n` 59, 1182, 1200
`\ctex_ltj_add_font_features:nn` 59, 1184, 1201
`\ctex_ltj_add_font_features:x` 1224
`\ctex_ltj_add_font_features:xn` 1183
`__ctex_ltj_alternate_cs:n` 1039, 1042, 1043, 1044, 1077, 1078, 1080, 1165, 1168, 1169, 1171, 1172, 1363, 1400, 1462, 1487, 1527, 1544
`\l_ctex_ltj_alternate_family_tl` 1425, 1428
`\l_ctex_ltj_alternate_options_clist` 1378, 1379, 1390, 1394
`\l_ctex_ltj_alternate_seq` 1007, 1347, 1348, 1356, 1362, 1370, 1618
`\l_ctex_ltj_base_CJKfamily_tl` 1008, 1024, 1075, 1462, 1487
`\l_ctex_ltj_base_family_tl` 65, 1040, 1070, 1076, 1406, 1447, 1506, 1525, 1530, 1531, 1543
`__ctex_ltj_change_encoding:` 55, 980, 1058, 1420
`\l_ctex_ltj_char_range_clist` 1006, 1331, 1340, 1359, 1377, 1389, 1401, 1424, 1446, 1597, 1599
`\ctex_ltj_char_range_key:nn` 67, 1591, 1594
`__ctex_ltj_char_range_parse_feature:w` 67, 1602, 1606
`\g_ctex_ltj_char_range_prop` ... 66, 1454, 1565, 1570
`\l_ctex_ltj_char_range_tl` 1454, 1457, 1563, 1565, 1567, 1569, 1576
`__ctex_ltj_check_family:n` 56, 1034, 1361
`\ctex_ltj_clear_alternate_font:n` 1516, 1522
`\l_ctex_ltj_current_font_tl` 50, 791, 795, 796, 808, 850, 855
`\l_ctex_ltj_current_shape_tl` 873, 878, 883, 884
`\ctex_ltj_declare_alternate_shape:nnnnnn` 64, 1423, 1443, 1445
`\ctex_ltj_declare_char_range:n` 1556, 1558
`__ctex_ltj_declare_char_range:nn` 1557, 1559
`\ctex_ltj_declare_char_range:nn` 66, 1560, 1561
`\ctex_ltj_declare_char_range:x` 1553
`\ctex_ltj_def_char_range_key:n` 67, 1566, 1586
`\g_ctex_ltj_default_features_clist` 1028, 1247, 1248
`\ctex_ltj_ensure_default_family:` ... 60, 1277, 1738
`\ctex_ltj_extract_font:` 51, 823, 830
`__ctex_ltj_family_csname:n` ... 1038, 1055, 1072, 1099
`\g_ctex_ltj_family_font_name_prop` 56, 1013, 1032, 1036, 1110, 1186, 1279, 1291
`\g_ctex_ltj_family_font_options_prop` 56, 1014, 1033, 1189
`\ctex_ltj_family_if_exist:nTF` 57
`\ctex_ltj_family_if_exist:xN` 1094
`\ctex_ltj_family_if_exist:xNF` 1281
`\ctex_ltj_family_if_exist:xNT` 1307
`\ctex_ltj_family_if_exist:xNTF` 1085, 1286, 1304, 1525
`\g_ctex_ltj_family_int` . 1152, 1154, 1439, 1440, 1442
`\g_ctex_ltj_family_name_prop` 56, 1031, 1040, 1062, 1096, 1406
`\l_ctex_ltj_family_tl` 1087, 1092, 1149, 1150, 1165, 1169, 1172, 1183, 1519, 1520
`__ctex_ltj_family_unknown_warning:n` 57, 1090, 1108, 1535
`\l_ctex_ltj_font_options_clist` 1010, 1015, 1016, 1017, 1025, 1061, 1190, 1191, 1194, 1397, 1398, 1402
`\ctex_ltj_fontspec:nn` 58, 1146, 1159
`__ctex_ltj_fontspec:nnn` 1160, 1180
`__ctex_ltj_fontspec:xnn` 1153
`\ctex_ltj_fontspec:xx` 1158, 1193, 1219
`\g_ctex_ltj_fontspec_family_tl` 1060, 1063, 1064, 1065, 1071, 1076, 1101
`\g_ctex_ltj_fontspec_prop` 1148, 1176, 1181
`\ctex_ltj_get_and_define_fonts:nN` 54, 923, 931
`\ctex_ltj_get_and_define_fonts_al:nN` 927, 930
`\ctex_ltj_get_and_define_fonts_ja:nN` 926, 932
`__ctex_ltj_gset_family_cs:nn` 56, 1018, 1053

- \ctex_ljt_if_alter_nate_shape_exist:n 864
- \ctex_ljt_if_alter_nate_shape_exist:nF 846
- \ctex_ljt_if_alter_nate_shape_exist:nT 833, 873
- \ctex_ljt_if_alter_nate_shape_exist:nTF 52
- \ctex_ljt_if_jfont:nTF 53, 55, 909, 920, 925
- \ctex_ljt_if_jfont_math:NTF 53, 915, 956
- _ctex_ljt_if_jfont_math:w 916, 919
- \l_ctex_ljt_jfm_tl ... 55, 966, 967, 969, 984, 986, 1029
- \c_ctex_ljt_math_fam_int 992, 993
- \c_ctex_ljt_math_family_tl
 - 1313, 1315, 1317, 1320, 1324
- \ctex_ljt_math_group_hook: 54, 773, 955, 961
- \c_ctex_ljt_math_tl
 - 987, 988, 990, 992, 1245, 1304, 1314, 1319, 1323
- _ctex_ljt_msg_def_family_map:n 1123, 1125
- _ctex_ljt_msg_family_map:n 1052, 1122, 1136
- _ctex_ljt_noembed_wrap:n 1000, 1002
- _ctex_ljt_patch_external_font:w 52, 836, 869
- \ctex_ljt_pickup_font:
 - 50, 804, 818, 829, 893, 935, 937, 939
- _ctex_ljt_pop_fontname: 807, 816, 895
- _ctex_ljt_push_fontname:n 803, 811, 892
- \ctex_ljt_reset_alter_nate_font:n 1518, 1538
- \g_ctex_ljt_reset_alter_nate_prop
 - 1045, 1529, 1541, 1549
- \ctex_ljt_save_alter_nate_family:Nnnn
 - 63, 1388, 1408, 1411
- \ctex_ljt_save_alter_nate_seq:cn 1363
- \ctex_ljt_save_alter_nate_seq:Nn 62, 1368, 1373
- \ctex_ljt_save_alter_nate_seq:Nnnwn 62
- \ctex_ljt_save_alter_nate_seq:Nnnwnw 1371, 1374
- _ctex_ljt_save_alter_nate_shape:cn 1461, 1486
- _ctex_ljt_save_alter_nate_shape:Nnn 65, 1502, 1512
- \ctex_ljt_save_char_range:n 67, 1564, 1571
- \ctex_ljt_save_char_range:nn 1573, 1574
- _ctex_ljt_save_fontname:w 813, 817
- \ctex_ljt_select_alter_nate_font: 52, 871, 1992
- \ctex_ljt_select_font: 50, 789, 1991
- _ctex_ljt_select_font_aux: 50, 792, 798, 809
- \ctex_ljt_set_alter_nate_family:coon 1399
- _ctex_ljt_set_alter_nate_family:n 1066, 1073
- \ctex_ljt_set_alter_nate_family:nn .. 63, 1366, 1395
- \ctex_ljt_set_alter_nate_family:nnF .. 62, 1011, 1357
- \ctex_ljt_set_alter_nate_family:nnn
 - 63, 1407, 1415, 1417
- \ctex_ljt_set_alter_nate_family:Nnnnn ... 1404, 1410
- \ctex_ljt_set_alter_nate_seq:n 62, 1329, 1338
- \ctex_ljt_set_alter_nate_shape:n ... 1471, 1483, 1494
- \ctex_ljt_set_alter_nate_shape:nnN .. 65, 1456, 1490
- \ctex_ljt_set_alter_nate_shape:nnn .. 64, 1459, 1466
- \ctex_ljt_set_alter_nate_shape:nnnn 1468, 1469
- \ctex_ljt_set_alter_nate_shape:Nnnnnnn 64, 1446, 1450
- \ctex_ljt_set_family:nnn 55, 1003, 1023, 1177
- \ctex_ljt_set_family:xxx
 - 1022, 1208, 1215, 1230, 1236, 1241, 1245
- \ctex_ljt_subst_font: 51, 824, 844
- \ctex_ljt_swap_cs:NN 63, 1422, 1429
- \ctex_ljt_switch_family:n 1083, 1093, 1178, 1214
- \ctex_ljt_switch_family:x 1082, 1150
- _ctex_ljt_tmp:w 1336, 1353, 1431, 1433, 1434
- \l_ctex_ljt_tmp_seq 1344, 1345, 1348, 1355
- \l_ctex_ljt_tmp_tl .. 1009, 1010, 1036, 1047, 1050,
 - 1187, 1195, 1211, 1214, 1215, 1304, 1305, 1307, 1308,
 - 1342, 1343, 1344, 1381, 1383, 1384, 1391, 1601, 1609, 1618
- \l_ctex_ljt_tmpa_tl 1281, 1286
- _ctex_ljt_um_char:Nn 742, 747
- _ctex_ljt_um_char_aux:Nn 752, 766
- _ctex_ljt_um_char_aux:Nx 749
- _ctex_ljt_um_define_math_chars: 731, 769
- \g_ctex_ljt_unknown_family_seq ... 1112, 1114, 1119
- _ctex_ljt_update_family_uid:N 64, 1016, 1397, 1437
- \ctex_ljt_update_mathfont: 61, 1299, 1302
- \ctex_ljt_update_mathfont:n 1305, 1308, 1311
- _ctex_ljt_use_global_options:N
 - 1017, 1026, 1379, 1398
- \ctex_ljt_use_math_group:Nn 54, 772, 949, 962
- \c_ctex_macosx_file_str 1802, 1806
- \ctex_make_spa:nn 127, 4546, 4554
- \ctex_mono_jfm:n 55, 963, 970
- \ctex_mono_jfm:o 971, 2047
- \l_ctex_msyh_suffix_tl
 - 3833, 3834, 3838, 3854, 3856, 3890
- \ctex_package_end_hook:n 131, 4761, 4766
- \ctex_package_end_hook:o 131, 4770
- \l_ctex_parameter_str ... 134, 4886, 4904, 4925, 4972
- \l_ctex_parameter_tl 4925, 4930, 4941
- \ctex_parse_name:NN 132, 4816, 4820
- \ctex_parse_name:NNn 4825, 4884
- _ctex_parse_name:nNNnnN 4837, 4846
- \ctex_parse_name:NNx 4821
- _ctex_parse_name:w 4859, 4865, 4874
- \ctex_patch_boot:NNnnTF
 - 132, 4777, 4782, 4787, 4802, 4807, 4810
- \ctex_patch_cmd:Nnn
 - 131, 1655, 1697, 2889, 2895, 2901, 2954, 2958,
 - 2964, 2968, 2976, 2979, 2984, 2988, 3632, 3633, 3634, 4785
- _ctex_patch_cmd:Nnnnnw . 135, 4777, 4782, 4787, 4943
- \ctex_patch_cmd_all:NnnnTF 131, 4780
- \ctex_patch_cmd_once:NnnnTF 131, 1649, 4775
- \ctex_patch_failure:N 501, 528, 532, 536, 540,
 - 1263, 1267, 1271, 1275, 1626, 1981, 2939, 2949, 4794, 4796
- _ctex_patch_false:w 4813, 4819, 4958, 4960, 4977, 5003
- _ctex_patch_rebuild:Nn
 - 134, 4916, 4922, 4955, 4996, 5000
- _ctex_patch_replace:nnnTF 135, 4950, 4962
- _ctex_patch_rescan:NNn 4924, 4925, 4926, 4934
- _ctex_patch_true:w 4812, 4818, 4956, 4989, 5001
- \ctex_plane_to_utfxvibe:Nn 44, 559, 577
- \l_ctex_prefix_str .. 134, 4885, 4903, 4924, 4981, 4982
- \l_ctex_prefix_tl 4924, 4929, 4940
- \ctex_preto_cmd:NnnTF 132, 525, 529,
 - 533, 537, 1260, 1264, 1268, 1272, 1975, 2936, 2945, 4800

- \l_ctex_punct_bounds_clist ... 4560, 4566, 4572, 4578
- \c_ctex_punct_family_clist 612, 622
- \l_ctex_punct_font 4558, 4559
- \ctex_punct_map_bfseries:nn
..... 47, 632, 3858, 3870, 3876, 3961,
3967, 4039, 4040, 4046, 4116, 4117, 4118, 4124, 4177, 4183
- \ctex_punct_map_family:nn ... 46, 627, 3851, 3857,
3867, 3868, 3875, 3957, 3958, 3959, 3966, 4035, 4036,
4037, 4045, 4112, 4113, 4114, 4123, 4173, 4174, 4175, 4182
- \ctex_punct_map_itshape:nn 47, 649,
3869, 3877, 3960, 3968, 4038, 4047, 4115, 4125, 4176, 4184
- \ctex_punct_map_series:nnn 636, 637, 640
- \c_ctex_punct_seq 4537, 4538, 4561
- \ctex_punct_set:n 46, 610,
3846, 3874, 3956, 3965, 4034, 4044, 4111, 4122, 4172, 4181
- \ctex_punct_space:nn 47, 658, 660
- \l_ctex_punct_tl
..... 280, 581, 582, 669, 971, 2044, 2045, 2046, 2047
- _ctex_rebuild_cmd:w ... 4916, 4917, 4921, 4996, 4997
- \l_ctex_replacement_str
..... 134, 4887, 4905, 4926, 4964, 4965, 4999
- \l_ctex_replacement_tl 4926, 4931, 4942
- _ctex_right_hook_aux:nn 4993
- _ctex_save_bounds:n 4562, 4570
- _ctex_save_cmap:Nn 476, 480
- _ctex_save_font_size:nn 106, 3450, 3481
- _ctex_save_selectfont: 1974, 1979
- \ctex_scheme_input:n 33, 79, 93
- \ctex_scheme_input:o 3085, 3351
- \l_ctex_scheme_tl 315, 3085, 3351
- \g_ctex_section_depth_flag
..... 34, 95, 309, 311, 3273, 3289, 3305
- \c_ctex_section_headings_seq
..... 83, 2288, 2289, 2294, 2777, 2836
- \ctex_select_size:.. 2065, 2072, 2075, 2076, 2097, 2151
- \ctex_set_default_ccwd:Nn ... 34, 110, 198, 2071, 2096
- \ctex_set_font_size:Nnn 107, 3517, 3532,
3540, 3551, 3562, 3563, 3564, 3565, 3566, 3567, 3568,
3571, 3579, 3590, 3601, 3602, 3603, 3604, 3605, 3606, 3607
- _ctex_set_font_size:nnNn 3520, 3523
- \ctex_set_ignorespaces: 46, 603, 605, 607, 609
- \ctex_set_upfamily:nnn
..... 69, 1660, 1672, 3893, 3894, 3895, 3896, 3897,
3898, 3900, 3980, 3981, 3982, 3983, 4059, 4060, 4061,
4062, 4136, 4137, 4138, 4139, 4140, 4142, 4195, 4196, 4197
- \ctex_set_upfonts:nnnnnn
..... 69, 1687, 1694, 3888, 3976, 4055, 4132, 4191
- \ctex_set_upmap:nnn 69,
1673, 1686, 1689, 1690, 1691, 1692, 3899, 3901, 4141, 4143
- \ctex_set_zhmap:n 42, 447, 453, 517, 1675
- \g_ctex_spa_iow 4548, 4551, 4553, 4563
- \c_ctex_std_class_tl 103, 3325, 3337, 3338, 3404, 3427
- \g_ctex_std_options_clist
..... 35, 138, 382, 394, 404, 406, 410, 415, 420, 425
- \ctex_titleps_hook: 95, 2887, 2920, 2921
- _ctex_titlesec_format:n 2832, 2840, 2843
- _ctex_titlesec_format:Nn 2822, 2846, 2848
- _ctex_titlesec_format_explicit:Nn 2823, 2858
- \ctex_titlesec_hook: 94, 2818, 2882
- _ctex_titlesec_hook:n 2832, 2836, 2838
- _ctex_titlesec_spacing:Nn 2841, 2868
- _ctex_titlesec_spacing:nnnnnn 2869, 2870
- _ctex_tmp:w 4823, 4878, 4889, 4910
- \l_ctex_tmp_box 33, 61, 1918, 1919, 2924, 2928
- \l_ctex_tmp_dim 33, 60,
2106, 2107, 2112, 2115, 2121, 2160, 2162, 2164, 2173, 2175
- \l_ctex_tmp_int 33, 59, 562,
564, 565, 567, 571, 573, 2120, 2122, 2123, 2129, 2130,
2162, 2163, 2164, 2167, 2168, 2173, 2175, 2776, 2779, 2783
- \l_ctex_tmp_tl . 33, 58, 366, 373, 388, 389, 482, 483,
561, 563, 2334, 2387, 2396, 2406, 3403, 3404, 3492, 3493
- \ctex_update_ccglue: 75, 1929, 2116, 2135, 2179
- \ctex_update_ccwd: 75, 1915, 2110, 2187
- \ctex_update_default_family:
..... 45, 70, 584, 1276, 1648, 1719
- \ctex_update_em_unit: 76, 1964, 2105, 2155
- \ctex_update_parindent: 80, 2083, 2138, 2180
- \ctex_update_size: 79, 2077, 2078, 2091
- \ctex_update_stretch: 79, 2082, 2103
- \ctex_update_xkanjiskip: 77, 1998, 2017
- \ctex_update_ziju: 80, 2085, 2153
- \c_ctex_upper_case_file_str 1790, 1798
- \ctex_varioref_hook: 99, 3064, 3075, 3083
- \c_ctex_version_tl 3, 6, 7, 8, 9, 20, 23
- _ctex_write_family:nn 4550, 4555
- \l_ctex_xkanjiskip_skip
..... 77, 2001, 2002, 2004, 2005, 2006, 2013, 2014
- \l_ctex_xkanjiskip_tl 2004, 2009, 2010
- \c_ctex_zero_tl 2080, 2087, 2089
- \ctex_zhmap_case:nnn
..... 35, 134, 261, 266, 271, 3844, 3938, 4006, 4099, 4163
- \g_ctex_zhmCJK_bool ... 35, 131, 260, 265, 270, 504, 585
- \ctex_zihao:n 106, 3434, 3435
- \ctex_ziju:n 2147, 2148
- \l_ctex_ziju_dim 2156, 2157, 2161, 2172, 2174, 2178, 2182
- \l_ctex_ziju_tl 2080, 2088, 2089, 2150, 2156
- \ctexbibname 2260, 2280
- \CTEXcounter 2200
- \CTEXdigits 24, 82, 2212
- \CTEXfilltwosides 29, 1888, 1898
- \CTEXindent 29, 81, 2183, 2185
- \CTEXnoindent 29, 81, 2189, 2191
- \CTEXnumber 24, 82, 2210
- \CTEXoptions 29, 41, 439, 441
- \ctexset 5, 41, 430, 436, 442
- \CTEXsetfont 29, 79, 2075
- \CTEXsetup 29, 41, 433, 435
- \CTEXsout 29
- \ctexspadef 47, 660, 4565
- \CTEXthechapter 16, 2575, 2578, 2966, 2986, 2987
- \CTEXtheparagraph 16
- \CTEXthepart 16, 2466, 2496

`\CTEXthesection` 16, 2956, 2970, 2978, 2990
`\CTEXthesubparagraph` 16
`\CTEXthesubsection` 16, 2960, 2981
`\CTEXthesubsubsection` 16
`\CTEXunderdblline` 29
`\CTEXunderdot` 29
`\CTEXunderline` 29
`\CTEXunderwave` 29
`\CTEXxout` 29
`\CurrentOption` 394

D

`declarecharrange` 25, 66
`\DeclareFontEncoding` 973
`\DeclareFontFamily`
 ... 43, 495, 501, 982, 3722, 3723, 3724, 3725, 3726, 3727
`\DeclareFontShape`
 ... 983, 985, 1664, 1665, 1668, 1669, 3738, 3739,
 3740, 3741, 3742, 3743, 3744, 3745, 3746, 3749, 3750,
 3751, 3752, 3753, 3754, 3755, 3756, 3757, 3761, 3762,
 3763, 3764, 3765, 3766, 3770, 3771, 3772, 3773, 3774,
 3775, 3776, 3777, 3778, 3781, 3782, 3783, 3784, 3785,
 3786, 3787, 3788, 3789, 3793, 3794, 3795, 3796, 3800,
 3801, 3802, 3803, 3804, 3805, 3806, 3807, 3808, 3811,
 3812, 3813, 3814, 3815, 3816, 3817, 3818, 3819, 3823, 3824
`\DeclareFontSubstitution` 976
`\DeclareKanjifamily`
 ... 1662, 1663, 3729, 3730, 3731, 3732, 3733, 3734
`\DeclareMathSizes` 3485
`\DeclareSymbolFont` 988, 1314
`\def` 2461, 2491, 2518,
 2532, 2542, 2569, 2598, 2624, 2643, 2666, 2669, 2702,
 2719, 2732, 2733, 2737, 3545, 3556, 3584, 3595, 4229, 4265
`\defaultCJKfontfeatures` 60, 1246, 1254
`\defbeamertemplate` 2996, 3014, 3032
`\defbeamertemplatealias` 3050, 3051, 3052
dim commands:
 `\dim_abs:n` 2174
 `\dim_add:Nn` 2157
 `\dim_compare:nNnF` 2142, 2165
 `\dim_compare:nNnTF` 112, 2107, 2158
 `\dim_eval:n` 2127, 2173
 `\dim_max:nn` 2927
 `\dim_min:nn` 2174
 `\dim_new:N` 60, 1928, 2182
 `\dim_set:Nn` 1919, 1965,
 1966, 1967, 2106, 2112, 2143, 2156, 2160, 2187, 2925, 3615
 `\dim_sub:Nn` 2164
 `\dim_to_decimal:n` 34, 3456, 3457, 3494
 `\dim_to_decimal_in_unit:nn` 4585
 `\dim_use:N` 120, 3615
 `\dim_zero:N` 2193
`\dimexpr` 2476, 2521, 2606, 2632, 3002, 3020, 3038
`\directlua` 707, 708
`\document` 4741, 4744

E

`\edef` 4248, 4249
`\else` 1651, 1652, 1656, 1657, 2432, 2441,
 2449, 2467, 2471, 2497, 2501, 2558, 2565, 2579, 2583,
 2586, 2594, 2654, 2659, 2672, 2676, 2694, 2701, 2710,
 2731, 2745, 2758, 4260, 4261, 4307, 4346, 4377, 4436, 4499
else commands:
 `\else:` 507, 867, 913, 1950, 3619
`\em` 69, 1649, 1655
`\eminnershape` 730, 1651, 1652, 1657, 1659
`\emshape` 729
`\end` 3010, 3028, 3046, 4630
`\endCJKfilltwosides` 1894
`\endcsname` 2667, 2668, 2673,
 2689, 2690, 2692, 2695, 2696, 2697, 2706, 2707, 2708,
 2711, 2712, 2713, 4255, 4257, 4258, 4259, 4262, 4264, 4268
`\endCTEXfilltwosides` 1894, 1899
`\endgroup` 2691, 2730, 2753, 3012, 3030, 3048, 4248, 4256, 4271
`\endinput` 4253
`\endlinechar` 4227, 4228
etex commands:
 `\etex_dimexpr:D` 120, 2121
 `\etex_glueexpr:D` 2113
`\everypar` 2658, 2748, 2760
`\EverySelectfont` 1707, 1983, 1986
exp commands:
 `\exp_after:wN`
 119, 820, 836, 838, 913, 916, 2869, 3438, 3485, 3520, 4894
 `\exp_args:cc` 3060
 `\exp_args:Nc` 2841, 2846, 4848
 `\exp_args:Nnc` 99
 `\exp_args:NNf` 1609
 `\exp_args:NNNo` 3614
 `\exp_args:Nno` 3072
 `\exp_args:No` 563, 1799, 2865, 4562, 4910
 `\exp_args:Nx` 293, 591, 665, 2147, 3434, 3613
 `\exp_last_unbraced:Nf` 4859
 `\exp_not:c` 855, 1883,
 1885, 1912, 2308, 2309, 2310, 2315, 2317, 2319, 2320,
 2323, 2324, 2327, 2339, 2340, 2341, 2342, 2343, 2344,
 2345, 2346, 2347, 2348, 2349, 2351, 2352, 2355, 2357,
 2359, 2361, 2363, 2365, 2784, 2785, 2786, 2787, 4840, 4841
 `\exp_not:N` 124, 125, 486,
 593, 594, 595, 782, 783, 785, 891, 976, 1062, 1063, 1064,
 1065, 1388, 1509, 1729, 1730, 1731, 1745, 1884, 2318,
 2782, 2787, 2853, 2854, 2863, 2864, 3454, 3656, 4839, 4929
 `\exp_not:n` 69, 1059, 1060, 1724, 2872, 4857, 4987
 `\exp_not:o` 388,
 784, 786, 1061, 1194, 1195, 1389, 1390, 1391, 1415,
 1509, 1746, 1971, 2406, 3061, 4744, 4929, 4930, 4931, 4986
 `\exp_not:v` 4720
`\expandafter`
 4230, 4231, 4255, 4256, 4257, 4258, 4259, 4264, 4268, 4269
`\ExplSyntaxOff` 66, 72, 495, 525, 529, 533, 537, 689,
 1260, 1264, 1268, 1272, 1650, 1975, 2936, 2995, 4591, 4790
`\ExplSyntaxOn` 71, 718, 3053, 4530

F

false commands:

\c_false_bool 2775
 \familydefault 1727
 fancyhdr 28, 40
 \fangsong 4213
 \fi 1711, 2420,
 2426, 2434, 2443, 2451, 2456, 2470, 2473, 2480, 2500,
 2503, 2511, 2551, 2552, 2555, 2558, 2567, 2582, 2585,
 2588, 2597, 2603, 2613, 2614, 2615, 2629, 2644, 2650,
 2656, 2662, 2679, 2680, 2698, 2699, 2714, 2715, 2717,
 2735, 2761, 2762, 2956, 2960, 2966, 2970, 2978, 2981,
 2986, 2987, 2990, 3006, 3024, 3042, 4260, 4261, 4270, 4523

fi commands:

\fi: 407, 541, 578, 597, 867, 913, 1950, 2961, 2971, 3529, 3619
 \figurename 2241
 figurename 11

file commands:

\g_file_current_name_tl 681, 1799
 \file_if_exist:nF 3835, 3837
 \file_if_exist:nTF 48, 1786, 1788, 1790, 1802, 3649, 3828
 \file_if_exist_input:nF 86
 \file_input:n 78, 87, 89, 517, 550

five commands:

\c_five 3457
 fntef 29, 40
 \font 3722, 3723, 3724, 3725, 3726, 3727
 fontset 7, 37, 111
 \fontsize 3438

fontspec commands:

\g_fontspec_encoding_tl 981
 \l_fontspec_family_tl 1448
 __fontspec_namewrap:n 55, 1000
 \fontspec_set_family:Nnn 1060, 1425
 \footnotesep 109, 3615
 \footnotesize 3551, 3590, 3614

four commands:

\c_four 311, 571, 573

fp commands:

\fp_compare:nNnF 3611
 \fp_compare:nNnT 3398, 3425
 \fp_eval:n 2150, 4581
 \fp_set:Nn 3399, 3426
 \fp_use:N 2087, 3613

G

GBK 6, 37
 \gdef 1702, 2806, 2807, 2808, 2809, 2812, 2813, 2814, 2815
 \global 2561, 2747, 2750
 \globaljfont 838
 \glueexpr 2429, 2453, 2487, 2528,
 2544, 2600, 2622, 2626, 2640, 2661, 2686, 2703, 2725,
 2732, 2733, 2743, 2757, 3001, 3011, 3019, 3029, 3037, 3047

group commands:

\group_begin:
 490, 723, 733, 800, 822, 848, 890, 917, 1005,

1057, 1334, 1419, 1504, 1524, 1540, 1723, 2314, 2322,
 2385, 2885, 2931, 3614, 4557, 4738, 4814, 4822, 4833, 4888
 \group_end: 494,
 503, 727, 745, 805, 826, 861, 894, 921, 1020, 1067,
 1354, 1426, 1510, 1536, 1547, 1735, 2325, 2329, 2405,
 2908, 2935, 3614, 4568, 4738, 4836, 4883, 4911, 4945, 4969
 \group_insert_after:N 858, 859
 \gtfamily 1651, 1656

H

hbox commands:

\hbox_set:Nn 1918, 2924
 heading 9, 39
 \heiti 4212
 \hskip ... 2686, 2703, 2725, 2732, 2733, 2756, 2757, 2977, 2980
 \hspace 2465, 2466, 2495, 2496, 2578
 \Huge 2512, 2536, 2617, 2635, 3127, 3154, 3568, 3607
 \huge 2481,
 2509, 2523, 2611, 3110, 3126, 3130, 3152, 3159, 3567, 3606
 hyperref 29, 40
 \hypersetup 358, 359, 1812
 \hyphenchar 3722, 3723, 3724, 3725, 3726, 3727

I

if commands:

\if_bool:N 504, 585
 \if_case:w 403, 3525
 \if_cs_exist:N 558
 \if_false: 107
 \if_meaning:w 1949
 \if_true: 106
 \ifctexpdf 34, 106, 107
 \ifdim 2416, 2648, 2682, 2721, 2739
 \ifincsname 3619
 \IfNoValueF 437, 443
 \IfNoValueTF 1212, 1473, 1578, 1611, 1613, 2381
 \ifnum 1709,
 2462, 2477, 2492, 2508, 2570, 2608, 2670, 2694, 2710, 4260
 \ifodd 2430, 2454,
 2463, 2477, 2493, 2508, 2563, 2572, 2601, 2610, 2627,
 2652, 2673, 2683, 2695, 2711, 2722, 2740, 2956, 2960,
 2966, 2970, 2978, 2981, 2986, 2987, 2990, 3004, 3022, 3040
 \ifx 4258, 4259, 4264
 \ifzhmappdf 4276, 4343, 4374, 4405, 4472
 \ignorespaces 2763
 \immediate 4267
 indent 28, 37
 \indexname 2244
 indexname 12
 \input 4529, 4594
 \insertcontinuationtext 2253, 2262, 2269
 \insertpart 3008, 3009
 \insertromanpartnumber 2999, 3097
 \insertsection 3026, 3027
 \insertsectionnumber 3017, 3189
 \insertsubsection 3044, 3045
 \insertsubsectionnumber 3035, 3217

int commands:

<code>\int_compare:nNnF</code>	2167, 3396, 3423
<code>\int_compare:nNnTF</code>	564, 2123, 3273, 3289, 3305
<code>\int_const:Nn</code>	992
<code>\int_div_truncate:nn</code>	79, 571
<code>\int_eval:n</code>	1474, 1476, 1477, 1579, 1581, 1582
<code>\int_from_hex:n</code>	563
<code>\int_gincr:N</code>	1152, 1439
<code>\int_incr:N</code>	2779
<code>\int_mod:nn</code>	573
<code>\int_new:N</code>	59, 1442
<code>\int_set:Nn</code>	554, 562, 2120, 2162
<code>\int_set_eq:NN</code>	548
<code>\int_sub:Nn</code>	567, 2168
<code>\int_to_Hex:n</code>	565, 570, 572
<code>\int_use:N</code>	487, 554, 905, 1154, 1440, 2783, 4562
<code>\int_zero:N</code>	2776

<code>\interlinepenalty</code>	2474, 2505, 2519, 2533, 2616, 2634, 2687, 2688, 2726, 2727
--------------------------------------	--

iow commands:

<code>\iow_close:N</code>	4551
<code>\iow_indent:n</code>	4725
<code>\iow_new:N</code>	4553
<code>\iow_now:Nx</code>	4563
<code>\iow_open:Nn</code>	4548
<code>\itemsep</code>	3548, 3559, 3587, 3598
<code>\itshape</code>	729, 1651, 1652, 1656, 1657

J

<code>\jfam</code>	957, 993
--------------------------	----------

K

<code>\kaishu</code>	4214
<code>\kanjifamily</code>	1639, 1640, 1641, 1644
<code>\kanjifamilydefault</code>	1647

kernel commands:

<code>\l_kernel_expl_bool</code>	70
--	----

keys commands:

<code>\l_keys_choice_tl</code>	294
<code>\keys_define:nn</code>	155, 388, 391, 997, 1327, 1436, 1513, 1551, 1590, 2019, 2040, 2058, 2092, 2215, 2236, 2264, 2274, 2277, 2280, 2406, 2790, 3622, 3669
<code>\keys_if_exist:nnF</code>	1588
<code>\l_keys_key_tl</code>	141, 143, 145
<code>\keys_set:nn</code>	27, 171, 177, 233, 242, 251, 277, 287, 294, 301, 331, 337, 430, 437, 443, 3088, 3142, 3166, 3197, 3231, 3247, 3260, 3275, 3283, 3291, 3299, 3306, 3307, 3309, 3375, 3377, 3711
<code>\keys_set_known:nn</code>	3354
<code>\keys_set_known:nnN</code>	1009

L

<code>\labelformat</code>	3072
<code>\languagealias</code>	3383, 3388, 3409, 3414
<code>\LARGE</code>	3566, 3605
<code>\Large</code>	2478, 3108, 3114, 3176, 3180, 3565, 3604
<code>\large</code>	3201, 3564, 3603

<code>\lastbox</code>	2751
<code>\leavevmode</code>	2426, 2644

left commands:

<code>\c_left_brace_str</code>	4880
<code>\leftmargin</code>	3545, 3556, 3584, 3595
<code>\leftmargini</code>	3545, 3556, 3584, 3595
<code>\let</code>	1715, 2671, 2678, 3538, 3577, 4251, 4257
<code>\linespread</code>	3613
<code>linespread</code>	10, 36
<code>linestretch</code>	10, 79
<code>\linewidth</code>	2121, 2129, 2161
<code>\lishu</code>	4216
<code>\listfigurename</code>	2239
<code>listfigurename</code>	11
<code>\listtablename</code>	2240
<code>listtablename</code>	11
<code>\LoadClass</code>	411, 416, 421, 426
<code>\long</code>	4265
<code>\lstlistingname</code>	783, 784
<code>\lstlistlistingname</code>	785, 786
<code>\ltjdefcharrange</code>	690, 691, 692, 694, 698, 699, 702, 705
<code>LTJFONTUID</code>	64
<code>\ltjgetparameter</code>	1923, 1953, 2001, 2015
<code>\ltjsetkanjiskip</code>	1940
<code>\ltjsetmathletter</code>	750
<code>\ltjsetparameter</code>	706, 712, 1629
<code>\ltjsetxkanjiskip</code>	2005

lua commands:

<code>\lua_escape_x:n</code>	866, 878, 905, 912, 1464, 1479, 1480, 1497, 1498, 1507
<code>\lua_now_x:n</code>	724, 760, 840, 866, 870, 875, 880, 902, 911, 979, 1485, 1509, 1755

M

<code>\MAKESPA</code>	4554, 4596
-----------------------------	------------

mark commands:

<code>\q_mark</code>	4895, 4898
<code>\markboth</code>	2483, 2504
<code>\mathalpha</code>	738
<code>\mathbin</code>	738
<code>\mathfence</code>	739
<code>\mathgroup</code>	957
<code>\mathop</code>	739
<code>\mathord</code>	738
<code>\mathpunct</code>	739
<code>\mathrel</code>	738

max commands:

<code>\c_max_dim</code>	2107
<code>\maxdimen</code>	2416
<code>\mcfamily</code>	1656
<code>\mddefault</code>	977, 983, 989, 1315, 1320

minus commands:

<code>\c_minus_one</code>	152, 548, 3396, 3423
---------------------------------	----------------------

mode commands:

<code>\mode_if_math:T</code>	951
<code>\mode_if_math:TF</code>	756

msg commands:

`\msg_critical:nnn` 3638
`\msg_critical:nnx` 53, 681
`\msg_error:nn` 3689
`\msg_error:nnn` .. 19, 22, 38, 41, 672, 685, 688, 3439, 3521
`\msg_error:nnx` 2231
`\msg_error:nnxx` 3655, 3684
`\msg_fatal:nn` 4535
`\msg_new:nn` 4533
`\msg_new:nnn` 140, 142, 144, 146, 148, 675,
1051, 1120, 1202, 1870, 2257, 3328, 3639, 3698, 4721, 4798
`\msg_new:nnnn` 12, 30, 45, 99, 2233, 3441, 3663, 3692
`\msg_warning:nn` 1197, 1874, 2037, 2048, 2279
`\msg_warning:nnn` 169, 175, 203, 213, 228, 237,
246, 275, 285, 299, 329, 335, 343, 348, 354, 1115, 3626, 3676
`\msg_warning:nnnn` 435, 441, 1883, 1890, 2185, 2191
`\msg_warning:nnx` 3347, 4727, 4797
`\msg_warning:nnxx` 1047

N

nan commands:

`\c_nan_fp` 182
`\newCJKfontfamily` 59, 1209
`\NewDocumentCommand` ... 430, 433, 439, 1081, 1207, 1209,
1217, 1222, 1228, 1234, 1239, 1244, 1246, 1374, 1466,
1571, 1606, 1642, 2075, 2146, 2183, 2189, 2210, 2212,
2375, 3433, 3618, 4211, 4212, 4213, 4214, 4216, 4217, 4219
`\newfontfeature` 994, 995, 996
`\newpage` 2545, 2550
`\ngostype` 44, 1773
`\nobreak` 2478,
2485, 2526, 2611, 2617, 2620, 2635, 2638, 2741, 3109, 3153
`\nobreakspace` 2478, 2509
`nocap` 28
`\noexpand` 4249, 4251, 4252, 4253
`nofonts` 28
`noindent` 28
`\nointerlineskip` 2419
`nopunct` 28
`\normalem` 1853
`\normalfont` 537, 539, 540, 1231,
1237, 1242, 1272, 1274, 1275, 1298, 2475, 2476, 2506,
2507, 2520, 2521, 2534, 2535, 2605, 2606, 2631, 2632, 2787
`\normalsize` 3234, 3250, 3263, 3532, 3571, 3609, 3617
`nospace` 28
`nozhmap` 28
`\null` 2452, 2548
`\numberline` 96,
2577, 2578, 2696, 2697, 2712, 2713, 2936, 2939, 2945, 2949

O

one commands:

`\c_one` 162, 1467, 1572, 2175, 2376, 4574
`\onecolumn` 2447

or commands:

`\or:` 405, 3527

P

`\pagegoal` 2416
`\pagestyle` 3374
`\par` 2415, 2427, 2478, 2509, 2512, 2523,
2536, 2611, 2617, 2635, 2645, 2741, 3000, 3001, 3008,
3011, 3018, 3019, 3026, 3029, 3036, 3037, 3044, 3047,
3090, 3109, 3120, 3137, 3145, 3153, 3192, 3194, 3225, 3227
`paragraph/afterskip` 30
`paragraph/beforeskip` 30
`paragraph/numbering` 15
`paragraph/runin` 20
`\parindent`
2142, 2143, 2165, 2186, 2187, 2192, 2193, 2475, 2476,
2520, 2521, 2605, 2606, 2631, 2632, 3002, 3020, 3038, 3307
`\parsep` 3547, 3548, 3558, 3559, 3586, 3587, 3597, 3598
`\parskip` 2418
`\part` 86, 2425, 2438
`part/beforeskip` 30
`part/fixbeforeskip` 20, 85
`part/numbering` 15
`part/pagestyle` 21, 85
`\partname` 2478, 2509, 2999, 3092
`\PassOptionsToClass` 410, 415, 420, 425
`\PassOptionsToPackage` 505, 1816, 2195, 4729
`\pdfmapline` 4277,
4278, 4279, 4280, 4281, 4282, 4283, 4285, 4286, 4287,
4288, 4289, 4290, 4291, 4293, 4294, 4295, 4296, 4297,
4298, 4300, 4301, 4302, 4303, 4304, 4305, 4406, 4407,
4408, 4409, 4410, 4411, 4412, 4414, 4415, 4416, 4417,
4418, 4419, 4420, 4422, 4423, 4424, 4425, 4426, 4427,
4429, 4430, 4431, 4432, 4433, 4434, 4473, 4474, 4475,
4476, 4477, 4478, 4480, 4481, 4482, 4483, 4484, 4485,
4487, 4488, 4489, 4490, 4491, 4493, 4494, 4495, 4496, 4497
`\pdfoutput` 4260
`\pdfstringdefDisableCommands` 3621

pdftex commands:

`\pdftex_pdffontattr:D` 486
`\pdftex_pdflastobj:D` 487
`\pdftex_pdfobj:D` 483
`\phantomsection` 2411

prg commands:

`\prg_do_nothing:`
..... 961, 1413, 1505, 1899, 1973, 2024, 2773, 4921
`\prg_new_conditional:Npnn` 864, 1946
`\prg_new_protected_conditional:Npnn` 1094, 4891
`\prg_return_false:` ... 867, 1104, 1950, 1954, 1958, 4901
`\prg_return_true:` 867, 1097, 1102, 1950, 1954, 1958, 4906
`\primitive` 4630
`\ProcessKeysOptions` 400
`\proofname` 2246
`proofname` 12

prop commands:

`\prop_get:NnN` 1189
`\prop_get:NnNT` 1406
`\prop_get:NnNTF` 1107, 1148, 1186, 1454, 3437, 3492, 3519
`\prop_get:NxNTF` 1096

`\prop_gpop:NnNT` 1036, 1040, 1541
`\prop_gput:Nnn` 1013, 1176, 3454
`\prop_gput:Nno` 1014, 1062, 1529, 1565
`\prop_gremove:Nn` 1045
`\prop_if_empty:Nf` 1110, 1279
`\prop_map_break:n` 1293
`\prop_map_inline:Nn` 1291
`\prop_new:N` 1031, 1032, 1033, 1181, 1549, 1570, 3448
`\protect` .. 2577, 2578, 2590, 2591, 2696, 2697, 2712, 2713,
 2890, 2892, 2896, 2898, 2902, 2904, 2912, 2917, 4852, 4853
`\ProvidesFile` 4265, 4273, 4340, 4371, 4402, 4469
`\providetranslation`
 4635, 4636, 4637, 4638, 4639, 4640, 4641,
 4642, 4643, 4644, 4645, 4646, 4647, 4648, 4649, 4650,
 4651, 4652, 4653, 4654, 4655, 4656, 4657, 4658, 4659,
 4660, 4661, 4662, 4663, 4664, 4665, 4666, 4667, 4668,
 4669, 4670, 4671, 4672, 4673, 4674, 4675, 4676, 4677,
 4678, 4679, 4680, 4681, 4682, 4683, 4684, 4685, 4686,
 4687, 4688, 4689, 4690, 4691, 4692, 4693, 4694, 4695,
 4696, 4697, 4698, 4699, 4700, 4701, 4702, 4703, 4704,
 4705, 4706, 4707, 4708, 4709, 4710, 4711, 4712, 4713, 4714
ptex commands:
`\ptex_kanjiskip:D` 1926, 1943, 1957
`\ptex_xkanjiskip:D` 2002, 2006, 2016
`punct` 9, 28, 38, 78
`\punctstyle` 582, 2045

Q

`\quad` . 2667, 2955, 2956, 2959, 2960, 2966, 2970, 2978, 2981,
 2986, 2987, 2990, 3115, 3160, 3170, 3202, 3235, 3251, 3264
`\quotation` 110, 3634

R

`\raggedright` 2475, 2520, 2605, 2631, 3107, 3151
`\refname` 2247, 2252, 2259, 2268, 2276, 2277
`refname` 12
`\refstepcounter` 99, 2464, 2494, 2573, 2674, 3077, 3078
`\relax` 2429, 2453, 2476, 2487, 2492, 2508, 2521, 2528, 2544,
 2600, 2606, 2622, 2626, 2632, 2640, 2646, 2661, 2675,
 2681, 2686, 2703, 2720, 2725, 2732, 2733, 2738, 2743,
 2755, 2757, 2977, 2980, 3001, 3002, 3011, 3019, 3020,
 3029, 3037, 3038, 3047, 4223, 4227, 4230, 4258, 4259, 4264
`\renewcommand` 2425, 2438, 2558
`\RenewDocumentCommand` 3078
`\RequirePackage` 27,
 56, 57, 350, 361, 506, 509, 510, 511, 664, 683, 686,
 720, 728, 782, 1825, 1834, 1852, 1858, 2196, 3428, 4774
`\RequirePackageWithOptions` 4730
`resetalternatfont` 26, 65
reverse commands:
`\reverse_if:N` 585
`\rmdefault` 1729
`\rmfamily` 525, 528, 1260, 1263, 1639

S

scan commands:

`\scan_stop:`
 ... 65, 120, 838, 957, 1506, 1507, 2113, 2121, 4558, 4740
`scheme` 9, 39
`\scriptsize` 3562, 3601
`\secdef` 2435, 2458, 2568
`section/afterskip` 30
`section/beforeskip` 30
`section/numbering` 15
`section/runin` 20
`\sectionname` 3017, 3186
`\selectfont`
 1088, 1532, 1545, 1645, 1697, 1975, 1981, 1994, 2077, 3438

seq commands:

`\seq_clear:N` 1007
`\seq_concat:NNN` 1347
`\seq_gput_left:Nn` 2295, 2296
`\seq_gput_right:Nn` 1114, 3460
`\seq_gset_eq:NN` 2294
`\seq_gset_from_clist:Nn` 2289, 2299, 4538
`\seq_if_empty:Nf` 1362
`\seq_if_in:NnF` 1112
`\seq_if_in:NnT` 463
`\seq_map_function:NN` 2836
`\seq_map_inline:Nn` 1370, 2398, 2777, 3066, 4561
`\seq_new:N` .. 456, 1119, 1355, 1356, 2288, 2292, 3449, 4537
`\seq_put_right:No` 1618
`\seq_set_filter:NNn` 1345
`\seq_set_from_clist:Nn` 457
`\seq_set_split:NnV` 1344
`\seq_use:Nnnn` 3446
`\setbox` 2751
`\setCJKfamilyfont` 59, 1133,
 1207, 3855, 3861, 3862, 3863, 3864, 3865, 3866, 3909,
 3910, 3918, 3919, 3921, 3922, 3923, 3924, 3926, 3948,
 3950, 3952, 3954, 3993, 3994, 3995, 3996, 4020, 4025,
 4030, 4032, 4070, 4072, 4074, 4075, 4089, 4090, 4091,
 4092, 4105, 4106, 4107, 4108, 4109, 4110, 4151, 4152,
 4153, 4154, 4155, 4156, 4169, 4170, 4171, 4204, 4205, 4206
`\setCJKmainfont` .. 60, 1129, 1228, 1233, 1249, 3847, 3905,
 3913, 3940, 3986, 4008, 4065, 4086, 4101, 4146, 4165, 4200
`\setCJKmathfont` 60, 1244, 1252
`\setCJKmonofont` 60, 1131, 1239, 1251, 3860, 3908,
 3917, 3947, 3992, 4019, 4069, 4088, 4104, 4150, 4168, 4203
`\setCJKromanfont` 1233, 1253
`\setCJKsansfont` .. 60, 1130, 1234, 1250, 3850, 3854, 3907,
 3915, 3946, 3991, 4014, 4068, 4087, 4103, 4148, 4167, 4202
`\SetSymbolFont` 990, 1319, 1323
`\sfdefault` 1730
`\sffamily` 529, 532, 1264, 1267, 1640
six commands:
`\c_six` 3457
skip commands:
`\skip_horizontal:N` 1933, 1937
`\skip_if_eq:nnT` 2000

\skip_if_eq:nnTF	1953, 1957	subsection/numbering	15
\skip_new:N	1945, 2013	subsection/runin	20
\skip_set:Nn 1923, 1926, 2004, 2014, 2114, 2125, 2170, 2178		\subsectionname	3035, 3216
\skip_set_eq:NN	1943, 2006	subsubsection/afterskip	30
\skip_zero:N	2134	subsubsection/beforeskip	30
\small	3540, 3579	subsubsection/numbering	15
\songti	4211	subsubsection/runin	20
\space	2574, 2611, 3092, 3149, 3186, 3216, 3315	sys commands:	
space	10, 28, 38, 77	\c_sys_engine_str	44
space commands:		\sys_if_engine_pdftex:TF	
\c_space_tl	487 130, 3842, 3933, 4001, 4080, 4097, 4161	
\special	1677, 1678, 1681,	\sys_if_engine_uptex:TF	
1682, 1832, 1847, 4308, 4309, 4310, 4311, 4312, 4313,	 3886, 3974, 4053, 4083, 4130, 4189	
4314, 4316, 4317, 4318, 4319, 4320, 4321, 4322, 4324,		\sys_if_engine_xetex:F	4531
4325, 4326, 4327, 4328, 4329, 4331, 4332, 4333, 4334,		\sys_if_engine_xetex:TF	4149
4335, 4336, 4347, 4348, 4349, 4350, 4351, 4353, 4354,		\sys_if_output_pdf:TF	105, 492, 1824, 3935, 4003
4355, 4356, 4357, 4359, 4360, 4361, 4362, 4364, 4365,			
4366, 4367, 4378, 4379, 4380, 4381, 4382, 4384, 4385,			
4386, 4387, 4388, 4390, 4391, 4392, 4393, 4395, 4396,			
4397, 4398, 4437, 4438, 4439, 4440, 4441, 4442, 4443,			
4445, 4446, 4447, 4448, 4449, 4450, 4451, 4453, 4454,			
4455, 4456, 4457, 4458, 4460, 4461, 4462, 4463, 4464,			
4465, 4500, 4501, 4502, 4503, 4504, 4506, 4507, 4508,			
4509, 4510, 4512, 4513, 4514, 4515, 4517, 4518, 4519, 4520			
\SplitArgument	1467, 1572, 2376		
stop commands:			
\q_stop	120, 125, 916,		
919, 1371, 1375, 1602, 1607, 4860, 4865, 4874, 4895, 4898			
str commands:			
\str_case:nnTF	4850		
\str_case:onF	1727, 1773		
\str_case:onTF	4865		
\str_case_x:nnF	1127, 1138		
\str_const:Nn	1806		
\str_const:Nx	43, 50, 1798		
\str_if_empty:NF	4981		
\str_if_empty:NTF	4936, 4972		
\str_if_eq:nnF	3695		
\str_if_eq:nnTF	965		
\str_if_eq:onF	586, 3702		
\str_if_eq:onTF . 508, 1821, 1829, 3379, 3406, 3420, 3678			
\str_if_eq_x:nnF	581		
\str_if_eq_x:nnTF	1283, 3675, 4856		
\str_lower_case:n	482		
\str_new:N	4774, 4885, 4886, 4887		
\str_set:Nn	4903, 4904, 4905		
\str_upper_case:n	1799		
\strutbox	3615		
sub3section	9, 39		
sub4section	9, 39		
subparagraph/afterskip	30		
subparagraph/beforeskip	30		
subparagraph/numbering	15		
subparagraph/runin	20		
subsection/afterskip	30		
subsection/beforeskip	30		

\@minus	3172, 3204, 3237, 3252, 3265, 3534, 3536, 3542, 3544, 3546, 3547, 3553, 3555, 3557, 3558, 3573, 3575, 3581, 3583, 3585, 3586, 3592, 3594, 3596, 3597
\@namedef	2890, 2892, 2896, 2898, 2902, 2904, 2912, 2917
\@nameuse	2890, 2892, 2902, 2904, 2912
\@ne	2873
\@nil	891
\@nobreakfalse	2746
\@noskipsecfalse	2750
\@noskipsectrue	2747
\@onlypreamble	34, 453, 519, 1249, 1250, 1251, 1252, 1253, 1254, 1672, 1686, 1694, 3662
\@part	87, 2435, 2458, 2461, 2491
\@pkgextension	6, 7, 8, 9, 20, 23, 682, 4769
\@plus	3121, 3122, 3172, 3173, 3204, 3205, 3237, 3238, 3252, 3265, 3278, 3294, 3534, 3535, 3536, 3542, 3543, 3544, 3546, 3547, 3553, 3554, 3555, 3557, 3558, 3573, 3574, 3575, 3581, 3582, 3583, 3585, 3586, 3592, 3593, 3594, 3596, 3597
\@popfilename	131, 4767
\@schapter	2568
\@secntformat	91, 2666, 2675
\@secpenalty	2660, 2661
\@sect	91, 2665, 2669
\@setfontsize	3524
\@spart	88, 2435, 2458, 2518, 2532
\@ssect	92, 2664, 2719
\@startsection	90, 92, 94, 100, 2643, 2782
\@svsec	2671, 2675, 2678, 2686, 2704, 2705
\@svsechd	2702, 2732, 2733, 2753
\@tempdima	2925, 2927
\@tempskipa	2417, 2418, 2453, 2457, 2600, 2604, 2626, 2630, 2646, 2648, 2649, 2660, 2681, 2682, 2720, 2721, 2738, 2739, 2742, 2755, 2756
\@tempswafalse	2450
\@tempswatrue	2448
\@topnewpage	2593
\@topnum	2561
\@xsect	92, 2718, 2736, 2737
\@zhdig	2206
\@zhnum	2198, 2205
\abstractname	12
\addCJKfontfeature	65
\AfterEndPreamble	130
\algorithmname	12
\appendixname	12, 22, 22, 22
\AtBeginDocument	34, 45
\AtBeginDvi	42
\AtBeginShipoutFirst	42
\AtEndOfClass	35
\AtEndOfPackage	35
\AtEndOfPackageFile*	130
\AtEndPreamble	130
\baselineskip	10, 11, 77, 79, 79
\bfseries	47
\bibname	12
\c@chapter	3316
\c@secnumdepth	2462, 2477, 2492, 2508, 2570, 2608, 2670, 2694, 2710
\c@section	3311
\caption	29
\catcode	33, 33, 44, 44, 121, 131
\ccwd	10, 10, 11, 24, 24, 29, 29, 34, 75, 76, 76, 79, 79, 79, 80
\chapter	9, 15, 20, 21, 22, 22, 86, 104
\chaptermark	13
\chaptername	15
\char	49
\CheckCommand	76
\Chinese	29, 29
\chinese	24, 24, 29, 29, 82
\CJK@ignorespaces	46, 599
\CJK@encoding	55, 55, 56, 64, 797, 801, 885, 972, 973, 977, 979, 981, 982, 983, 985, 989, 991, 1314, 1317, 1319, 1323, 1464, 1479, 1480, 1497, 1498
\CJK@envStart	45, 45, 45, 45, 593
\CJK@family	50, 50, 792, 794, 797, 802, 849, 885, 1085, 1531, 1541, 1543, 1994, 1995
\CJK@ignorespaces	46, 46, 604, 608
\CJK@input	44, 45, 557
\CJK@loadBinding	44, 45, 589
\CJK@loadEncoding	44
\CJK@makeActive	45, 588
\CJK@plane	44, 77, 471, 472, 482, 1983
\CJK@surr	44, 44, 44, 558, 577
\CJK@upperReset	45
\CJKfamily	56
\CJKfamilydefault	45, 45, 60, 60, 69, 70, 70, 70, 71, 71
\CJKglue	79, 79, 109
\CJKhook	46, 46
\CJKrmdefault	43
\CJKsymbol	77
\CJKunderdotbasesep	29
\CJKunderline	29
\clearpage	45
\contentsname	11, 11
\CS	51, 51
\csname	109
\CTEX@afterindent	2652, 2769, 2774
\CTEX@aftertitle	2729, 2734, 2768, 2773
\CTEX@appendix@number	2795, 2807, 2813
\CTEX@appendix@numbering	2796, 2809, 2815
\CTEX@chapter@afterindent	2563
\CTEX@chapter@aftername	2612
\CTEX@chapter@afterskip	2622, 2640
\CTEX@chapter@aftertitle	2619, 2637
\CTEX@chapter@beforeskip	2600, 2626
\CTEX@chapter@fixbeforeskip	2392, 2601, 2627
\CTEX@chapter@format	2607, 2633
\CTEX@chapter@indent	2606, 2632
\CTEX@chapter@numbering	2572, 2610, 2815, 2966, 2986, 2987
\CTEX@chapter@pagestyle	2390, 2560

<code>\CTEX@chapter@titleformat</code>	2618, 2636	<code>\CTEX@toc@width@n</code>	2922, 2937, 2947
<code>\CTEX@chaptername</code>	2612	<code>\CTEX@todayold</code>	2214, 2219
<code>\CTEX@document@left@hook</code>	4737, 4743	<code>\CTEX@update@sectionformat@n</code>	92, 2651, 2765
<code>\CTEX@document@right@hook</code>	4739, 4745	<code>\CTEX@verbatim@font@hook</code>	1624, 1628
<code>\CTEX@Family@CMap</code>	468, 496, 499	<code>\ctex@zhmap@endinput</code>	4525
<code>\CTEX@fixbeforeskip</code>	86, 2413, 2455, 2602, 2628	<code>\CTEX@digits</code>	24, 24
<code>\CTEX@makeheadinganchor</code>	86, 2409, 2411, 2468, 2498, 2580, 2677	<code>\CTEX@number</code>	24, 24, 24, 24, 24
<code>\CTEX@part@afterindent</code>	2430	<code>\CTEX@options</code>	29, 29
<code>\CTEX@part@aftername</code>	2479, 2510, 3005	<code>\ctexset</code> . 5, 5, 5, 5, 5, 5, 5, 7, 11, 14, 14, 25, 26, 29, 29, 111	
<code>\CTEX@part@afterskip</code>	2487, 2528, 2544, 3011	<code>\CTEX@setup</code>	29
<code>\CTEX@part@aftertitle</code> ...	2484, 2514, 2525, 2538, 3009	<code>\CTEX@thechapter</code>	12, 13, 13, 16
<code>\CTEX@part@beforeskip</code>	2429, 2453, 3001	<code>\CTEX@thesection</code>	12, 13, 16, 97
<code>\CTEX@part@fixbeforeskip</code>	2391, 2454	<code>\CTEX@underdotbasesep</code>	29
<code>\CTEX@part@format</code>	2476, 2507, 2522, 2535, 3003	<code>\CTEX@underline</code>	29
<code>\CTEX@part@indent</code>	2476, 2521, 3002	<code>\curr@fontshape</code>	803, 833, 842, 846, 892
<code>\CTEX@part@numbering</code> ...	2463, 2477, 2493, 2508, 3004	<code>\DeclareAlternateKanjiFont</code>	64
<code>\CTEX@part@pagestyle</code>	2389, 2445	<code>\DeclareFontEncoding</code>	53
<code>\CTEX@part@titleformat</code> ..	2482, 2513, 2524, 2537, 3009	<code>\DeclareFontFamily</code>	42, 51
<code>\CTEX@partname</code>	2479, 2510, 3005	<code>\DeclareFontShape</code>	51
<code>\CTEX@postappendix</code>	2800, 2808, 2814	<code>\DeclareFontShape@</code>	64, 1423
<code>\CTEX@postchapter</code>	2814	<code>\DeclareRobustCommand</code>	132
<code>\CTEX@postsection</code>	2808	<code>\define@newfont</code>	50, 825
<code>\CTEX@preappendix</code>	2799, 2806, 2812	<code>\do@subst@correction</code>	50, 51, 824
<code>\CTEX@prechapter</code>	2812	<code>\em</code>	69, 69, 69
<code>\CTEX@presection</code>	2806	<code>\em@innershape</code>	49
<code>\CTEX@runin</code>	2683, 2722, 2740, 2770, 2775	<code>\em@shape</code>	49
<code>\CTEX@save@appendix</code>	2801, 2804	<code>\enc@update</code>	1698, 1703
<code>\CTEX@save@refstepcounter</code>	3077, 3080	<code>\escapechar</code>	54
<code>\CTEX@section@aftername</code>	3023	<code>\EverySelectfont</code>	76
<code>\CTEX@section@afterskip</code>	3029	<code>\external@font</code>	835, 836, 838
<code>\CTEX@section@aftertitle</code>	3027	<code>\extract@font</code>	50, 823
<code>\CTEX@section@beforeskip</code>	3019	<code>\f@baselineskip</code>	2113
<code>\CTEX@section@format</code>	3021	<code>\f@encoding</code>	801, 841, 954
<code>\CTEX@section@indent</code>	3020	<code>\f@family</code>	50, 802, 841, 849
<code>\CTEX@section@numbering</code>	2809, 2956, 2970, 2978, 2990, 3022	<code>\f@series</code>	797, 885
<code>\CTEX@section@titleformat</code>	3027	<code>\f@shape</code>	797, 885
<code>\CTEX@sectionname</code>	3023	<code>\f@size</code>	797, 803, 880, 888, 891, 892, 1919, 1965, 2928
<code>\CTEX@selectfont@hook</code>	76, 1970, 1971, 1973, 1976, 1983, 1986	<code>\fangsong</code>	8, 8
<code>\CTEX@setcurrentlabel@n</code>	3055, 3081	<code>\figurename</code>	11
<code>\CTEX@subsection@aftername</code>	3041	<code>\font</code>	51, 51
<code>\CTEX@subsection@afterskip</code>	3047	<code>\font@name</code>	50, 51, 806, 813, 814, 817, 820, 838, 839, 856, 934, 935, 936, 937, 938, 945, 1710
<code>\CTEX@subsection@aftertitle</code>	3045	<code>\fontcharwd</code>	76, 76
<code>\CTEX@subsection@beforeskip</code>	3037	<code>\footnotesep</code>	109, 109, 109
<code>\CTEX@subsection@format</code>	3039	<code>\footnotesize</code>	109
<code>\CTEX@subsection@indent</code>	3038	<code>\get@external@font</code>	832
<code>\CTEX@subsection@numbering</code>	2960, 2981, 3040	<code>\getanddefine@fonts</code>	54, 930, 931
<code>\CTEX@subsection@titleformat</code>	3045	<code>\globaljfont</code>	51
<code>\CTEX@subsectionname</code>	3041	<code>\hbox</code>	86
<code>\CTEX@thechapter</code>	2813	<code>\heiti</code>	8
<code>\CTEX@thesection</code>	2807	<code>\hrule</code>	86
<code>\CTEX@titleformat@n</code>	92, 2728, 2734, 2767, 2772	<code>\Hy@driver</code>	1829
<code>\CTEX@titlepslabel@clear</code>	96, 2899, 2914	<code>\Hy@unicodetrue</code>	73
<code>\CTEX@titlepslabel@set</code>	96, 2893, 2905, 2909	<code>\HyPsd@ConvertToUnicode</code>	73
		<code>\HyPsd@LoadUnicode</code>	73
		<code>\HyPsd@pdfencoding</code>	73, 73

\HyPsd@ToBigChars	73	\pagestyle	9
\if@mainmatter	2571, 2609, 2986, 3342, 3343	\paragraph	9, 9, 9, 9, 9, 20, 34, 34
\if@nobreak	2657	\parindent	10, 29, 79, 81, 109
\if@noskipsec	2426, 2644, 2749	\parskip	86, 86, 86
\if@openright	2439, 2547, 2558	\part	14, 20, 20, 21, 86
\if@tempswa	2553	\partname	15
\if@twocolumn	2446, 2592	\partpage	14, 97
\if@twoside	2546, 2957, 2967	\PassOptionsToClass	41
\iffalse	53	\PassOptionsToPackage	72
\ifin@	53, 913	\pdfmapline	115, 118
\iftrue	53	\pdfstringdef	73
\indexname	12	\pickup@font	50
\insertcontinuationtext	12	\pickup@jfont	829
\itdefault	51	\ProcessKeysOptions	27
\itshape	47	\proofname	12
\kaishu	8	\protected	49, 96
\kanjiskip	77	\protected@edef	2211, 2213, 2675, 3057
\labelformat	99, 99	\ProvidesFile	122
\leftmark	13	\ps@fancy	97, 2976, 2979, 2984, 2988
\linewidth	79, 79	\ps@headings	97, 2954, 2958, 2964, 2968
\lishu	8, 8	\refname	12, 12
\listfigurename	11	\refstepcounter	99
\listtablename	11	\relax	44, 79
\ltj@@does@alt@set	52	\reset@font	539, 1274
\ltj@@getjfontnumber	53, 901	\rightmark	13
\ltj@@set@stackfont	943, 944, 945	\rmfamily	47
\ltj@allalchar	725, 761	\roman	24
\ltj@curjfnt	51, 77	\scantokens	132
\ltj@pickup@altfont@auxy	53, 886	\scriptfont@name	937, 944
\ltj@pickup@altfont@copy	53, 899	\section	9, 9, 14, 15, 20, 20, 21, 21, 22, 22, 30, 83
\ltj@setpar@global	942	\section*	15
\ltj@tempcntc	53, 905	\sectionmark	13, 13
\ltjalchar	49	\sectionname	15
\m@ne	2462, 2477, 2570, 2608, 3722, 3723, 3724, 3725, 3726, 3727	\sectionpage	14, 98
\markboth	13	\selectfont	52, 70, 76, 76, 76, 77, 79
\markright	13	\setCJKfamilyfont	25, 25, 25
\math@bgroup	953	\setCJKmainfont	25
\math@egroup	958	\sf@size	936
\math@fonts	940	\sffamily	47
\mathbb	109	\size@update	79, 79, 2077, 2091
\maxdimen	11, 79	\sldefault	51
\meaning	132, 134, 134, 134	\songti	7
\newcommand	132	\special	7, 72
\newrobustcmd	132	\split@name	891
\newtitlemark	96, 96	\SplitArgument	85
\nfss@catcodes	55	\ssf@size	938
\ngostype	71	\strutbox	109
\normalsize	8, 8, 105	\subparagraph	9, 9, 9, 9, 20, 34
\nouppercase	44	\subsectionname	15
\p@	1919, 1965, 2509, 2590, 2591, 2599, 2611, 2621, 2625, 2639, 2928, 3120, 3146, 3147, 3153, 3534, 3535, 3536, 3542, 3543, 3544, 3546, 3547, 3553, 3554, 3555, 3557, 3558, 3573, 3574, 3575, 3581, 3582, 3583, 3585, 3586, 3592, 3593, 3594, 3596, 3597	\subsectionpage	14, 98
\pagenumbering	24	\subst@correction	852, 859
		\tablename	12
		\textfont@name	935, 943
		\tf@size	934
		\thechapter	13, 16, 22
		\thesection	13, 15, 16, 22, 97

- \titleformat 94
- \titlespacing 94
- \today 11, 11
- \trans@languagepath 3391, 3417
- \ttfamily 47
- \ttl@extract 94
- \ttl@setsubmark 95, 2895, 2901
- \ttl@settopmark 95, 2889, 2920
- \ttlh@hang 2854, 2864
- \ttlh@runin 2853, 2863
- \tw@ 1709
- \uppercase 44, 44
- \use@mathgroup 54, 772, 962
- \usepackage 28, 29
- \verbatim@font 1622, 1626
- \wrong@fontshape 50
- \x@protect 4853
- \xdef 96
- \XeTeXglyphbounds 126
- \xkanjiskip 77, 77, 77, 77
- \yahei 8, 8
- \youyuan 8, 8
- \z@ . 712, 715, 2475, 2520, 2561, 2605, 2631, 2648, 2682,
2721, 2739, 2751, 2873, 3535, 3543, 3554, 3574, 3582, 3593
- \zhdig 82
- \zhdigits 24
- \zhnum 82
- \zhnumber 24, 24, 24, 24
- \zihao 23, 23
- \ziju 24
- \zw 76
- tex commands:
- \tex_def:D 4929
- \tex_divide:D 79, 2122, 2163
- \tex_edef:D 4983
- \tex_endlinechar:D 548, 554
- \tex_font:D 486, 4558
- \tex_iftrue:D 3343
- \tex_ignorespaces:D
..... 1082, 1220, 1225, 2030, 2147, 3434, 4740
- \tex_immediate:D 483
- \the 4227, 4230, 4231, 4250, 4254
- \thechapter 2574, 2577, 2611, 2965, 2985, 3150
- \theparagraph 3249
- \thepart 2465, 2478, 2495, 2509, 3094
- \thesection 2955, 2969, 2977, 2989, 3169
- \thesubparagraph 3262
- \thesubsection 2959, 2980, 3200
- \thesubsubsection 3233
- \thispagestyle 2444, 2445, 2549, 2559, 2560
- three commands:
- \c_three 309, 2167, 3289, 4575
- \tiny 3563, 3602
- tl commands:
- \tl_clear:c 1855, 1866, 2382
- \tl_clear:N
.. 193, 366, 389, 1563, 1567, 1995, 2068, 2396, 3489, 4937
- \tl_const:cn 659, 682
- \tl_const:cx 2306, 2312
- \tl_const:Nn 409, 414, 419,
424, 521, 972, 987, 1256, 1257, 1258, 1259, 3325, 3344, 3346
- \tl_const:Nx 3, 1313, 2087
- \tl_gput_right:cn 464
- \tl_gput_right:Nn 4734, 4736, 4758
- \tl_gset:Nn 1287, 1796, 1803, 1804
- \tl_gset:Nx 565, 568, 934, 936, 938, 1725, 1753, 1771, 3680
- \tl_gset_eq:NN 1064, 1748
- \tl_gset_rescan:Nnn 1294
- \tl_head:N 3493
- \tl_if_blank:nTF 1380, 1476, 1477, 1581, 1582, 1596, 1614
- \tl_if_blank_p:n 1346
- \tl_if_empty:cT 3071
- \tl_if_empty:NF 792, 2140
- \tl_if_empty:nF 1666, 1679
- \tl_if_empty:NTF 2318, 3643
- \tl_if_empty:nTF 4900
- \tl_if_eq:NNT 1721
- \tl_if_eq:NNTF 2080, 3404
- \tl_if_exist:NF
..... 520, 522, 523, 524, 1255, 1257, 1258, 1259, 4757
- \tl_if_exist:NT 3427
- \tl_if_exist:NTF 84, 2262, 2273, 2276, 3337
- \tl_if_in:nnT 736
- \tl_if_in:NnTF 4964
- \tl_map_inline:Nn 1492
- \tl_map_inline:nn 3490
- \tl_new:c 2304, 2305
- \tl_new:N 58, 128,
132, 522, 523, 524, 794, 795, 883, 969, 1024, 1050, 1070,
1071, 1092, 1428, 1569, 1741, 2009, 2088, 2101, 2799,
2800, 3833, 4747, 4748, 4757, 4818, 4819, 4940, 4941, 4942
- \tl_put_left:Nn 1639, 1640, 1641, 1978, 4767
- \tl_put_right:Nn
..... 373, 1609, 2355, 2357, 2359, 2361, 2363, 2365
- \tl_put_right:Nx 940, 1576, 2334, 3493, 3494
- \tl_replace_all:Nnn 1343, 1384
- \tl_rescan:nn 134, 4991
- \tl_rescan:no 4982
- \tl_set:cn 2380, 2383
- \tl_set:Nn 114, 115, 188, 221,
222, 514, 515, 516, 783, 785, 796, 884, 966, 967, 1008,
1075, 1087, 1342, 1381, 1383, 1424, 1597, 1601, 1635,
1636, 1637, 1638, 1647, 1994, 2010, 2053, 2064, 2102,
2387, 3403, 3621, 3834, 3838, 3880, 3881, 3882, 4812, 4813
- \tl_set:Nx
.. 129, 482, 561, 835, 1211, 1743, 2044, 2150, 2850, 2869
- \tl_set_eq:NN
801, 802, 849, 935, 937, 981, 1076, 1101, 1531, 1543, 2089
- \tl_set_rescan:Nno 4938
- \tl_tail:N 2855
- \tl_tail:n 2865

`\tl_to_str:n` 125, 126, 4881, 4910, 4951, 4952
`\tl_trim_spaces:n` 1560
`\today` 2214, 2219, 2222, 2227
`today` 11, 82
token commands:
`\token_get_replacement_spec:N` 4838, 4860
`\token_to_meaning:N` 4895
`\token_to_str:N`
 .. 54, 916, 925, 934, 936, 938, 1129, 1130, 1131, 1133,
 1140, 1141, 1142, 1204, 2259, 2260, 3442, 4565, 4725, 4797
`\toks` 4227, 4228, 4230, 4231, 4250, 4254
`\topsep` 3546, 3557, 3585, 3596
`\topskip` 2417
`\tracingfonts` 1709
`\TrimSpaces` 433
true commands:
`\c_true_bool` 2774
`\ttdefault` 1731
`\ttfamily` 533, 536, 1268, 1271, 1641
two commands:
`\c_two` 95, 163, 377, 381, 2168, 2928, 3273, 3305
`\c_two_hundred_fifty_six` 564, 567
`\twocolumn` 2554
`\typeout` 2574, 2575

U

um commands:
`__um_define_math_chars:` 769
`__um_input_math_symbol_table:` 744
`__um_switchto_literal:` 774
`__um_sym:nnn` 734
`\undefined` 1715, 4252
`\unless` 2683, 2722, 2740
`\unskip` 2754
`\updefault` ... 977, 983, 985, 989, 991, 1315, 1317, 1320, 1324
`\upshape` 730, 1656, 1659
use commands:
`\use:c` 803, 841,
 842, 892, 934, 936, 938, 992, 1042, 1544, 4965, 4985, 4999
`\use:n` 133, 1284, 1288, 1742, 1897, 2772, 2933, 3335
`\use:nn` 4993
`\use:x` 64, 122, 388, 780, 891, 974, 1023,
 1159, 1386, 1560, 2403, 3452, 3652, 4834, 4876, 4927, 4948
`\use_i:nn` 108, 913, 4778, 4783, 4803, 4808
`\use_i:nnn` 261
`\use_ii:nn` 109, 913, 4778, 4783, 4803, 4808
`\use_ii:nnn` 134, 266
`\use_iii:nnn` 271
`\use_none:n` .. 189, 194, 199, 1059, 1287, 1421, 1898, 2200
`\use_none:nn` 499
`\usebeamerfont` 2999, 3017, 3035, 3136, 3191, 3224

`\usebeamerfont` 2999, 3008,
 3017, 3026, 3035, 3044, 3135, 3138, 3190, 3193, 3223, 3226
`\uselanguage` 3382, 3387, 3408, 3413
`\usepackage` 4725
`UTF8` 6, 37

V

`\verse` 110, 3632, 3633
`\vfil` 2452, 2543
`\vskip` 2486, 2487,
 2509, 2527, 2528, 2544, 2611, 2621, 2622, 2639, 2640,
 2742, 2743, 3000, 3018, 3036, 3120, 3137, 3153, 3192, 3225
`\vspace` 2457, 2599, 2604, 2625, 2630

W

`winfonts` 28
`\write` 4267

X

`\x` 4229, 4232,
 4233, 4234, 4235, 4236, 4237, 4238, 4239, 4240, 4241,
 4242, 4243, 4244, 4245, 4246, 4247, 4248, 4255, 4265, 4269
`\xdef` 4268
`\xeCJKsetup` 665, 1862, 1937, 2025, 2028, 2031, 2046
xetex commands:
`\xetex_charglyph:D` 4562
`\xetex_glyphbounds:D` 4586

Y

`\yahei` 4219
`\youyuan` 4217

Z

zero commands:

`\c_zero` 161, 2123, 3397, 3424
`\c_zero_dim` ... 2115, 2127, 2142, 2158, 2165, 2369, 3306
`\c_zero_fp` 2087, 3398, 3425, 3611
`\c_zero_skip` 2367, 2368
`\zhdig` 2206
`\zhdigits` 24, 2213
`zhmap` 7, 28, 38
`\zhnum` 2205
zhnum commands:
`\zhnum_counter:n` 2197
`\zhnumber` 24, 2211
`\zhnumsetup` 2223, 2228
`\zhtoday` 2222, 2227
`\zihao` 23, 106, 3433, 3442
`zihao` 8, 36
`\ziju` 24, 80, 2146
`\zw` 713, 1923, 1966, 2011