

CT_EX 宏集手册

ctex.org

2016/08/27 v2.4.3*

简介

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

CT_EX 宏集支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX、LuaL^AT_EX、upL^AT_EX 等多种不同的编译方式,并为它们提供了统一的界面。主要功能由宏包 ctex 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
		8.1 字号与间距	23
第 2 节 简明教程	3	8.2 中文数字转换	24
2.1 CT _E X 宏集的组成	3	8.3 杂项	24
2.2 CT _E X 宏集的安装和更新	3	第 9 节 LuaL ^A T _E X 下的中文支持方式	24
2.3 使用 CT _E X 文档类	4	9.1 LuaL ^A T _E X 下替代字体的设置	25
2.4 使用 ctex 宏包	4	第 10 节 CT _E X 宏集的配置文	26
第 3 节 宏包选项与 \ctexset 命令	5	10.1 修改宏包默认选项	26
第 4 节 编译方式、编码与中文字库	5	10.2 宏包载入后的配置	27
4.1 编译方式	5	10.3 配置标题中文翻译	27
4.2 中文编码	6	10.4 自定义字体集	27
4.3 中文字库	6	第 11 节 对旧版本的兼容性	27
第 5 节 排版格式设定	8	11.1 CT _E X 0.8a 及以前的版本	27
5.1 文档默认字号	8	11.2 CT _E X 0.9–CT _E X 1.0d	28
5.2 章节标题风格	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	29
5.3 排版方案选项	9	11.4 CT _E X 2.2 之前的版本	30
第 6 节 文档汉化	11	11.5 CT _E X 2.4.1 和 2.4.2	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	33
第 7 节 章节标题格式设置	14	版本历史	136
		代码索引	137

*ctex-kit rev. e653f84.

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](https://github.com/CTeX-org/ctex-kit)²。

最初，Knuth 在设计开发 T_EX 的时候没有考虑到多国语言支持，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。

关于宏集名字的说明

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

CT_EX 宏集是由 [CT_EX 社区](https://github.com/CTeX-org/ctex-kit) 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](https://github.com/CTeX-org/ctex-kit) 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-ja 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`

`\ctexset {<键值列表>}`

New: 2014-03-18

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 $\text{up}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 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 $\text{pdf}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 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 $\text{up}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 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 $\text{up}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 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
$\text{pdf}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$	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
$\text{up}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ + 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 $\text{pdf}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 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 `CTEX` 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 ☆

`CTEX` 文档类是在 `LATEX` 标准文档类之上开发的。因此,除了可以使用 `CTEX` 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,`CTEX` 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 `CTEX` 文档类中依旧有效。例如,设置纸张大小和方向的 `a4paper` 和 `landscape`,设置单双面的 `oneside` 和 `twoside` 等。`CTEX` 会将这些选项传给标准文档类¹²。

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

5.2 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 8 月 27 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 8 月 27 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年八月二十七日”。使用全汉字的日期格式。
`old` 效果为“August 27, 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”。

<code>tablename</code> ★	<code>tablename = <名字></code> 设置表格环境标题名 <code>\tablename</code> 。中文默认为“表”。
<code>abstractname</code> ★	<code>abstractname = <名字></code> 设置摘要 <code>abstract</code> 环境标题名 <code>\abstractname</code> 。中文默认为“摘要”。注意 <code>book</code> 类没有摘要,该选项无效。
<code>indexname</code> ★	<code>indexname = <名字></code> 设置索引标题名 <code>\indexname</code> 。中文默认为“索引”。
<code>appendixname</code> ★	<code>appendixname = <名字></code> 设置附录标题名 <code>\appendixname</code> 。中文默认为“附录”。
<code>bibname</code> ★	<code>bibname = <名字></code> 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”。
<code>proofname</code> ★	<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。
<code>refname</code> ★	<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”。
<code>algorithmname</code> ★	<code>algorithmname = <名字></code> 设置算法环境标题名 <code>\algorithmname</code> 。中文默认为“算法”。
<code>continuation</code> ★	<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 `CTEX` 的文档类或是用 `ctex` 宏包加载了 `heading` 选项时,会设置整个文档的页面格式(`page style`)为 `headings`,即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

同时,`CTEX` 宏包也会对默认的 `headings` 页面格式进行修改,使之调用 `\CTEXthechapter`、`\CTEXthesection` 等宏来正确显示中文的章节编号。

`CTEX` 宏包的默认页面格式设置是经过汉化的 `headings`,其基本效果如本文档所示,只在页眉一侧显示章节编号和标题,另一侧显示页码。

更复杂的页面格式可以通过调用 `fancyhdr`、`titleps` 等宏包来设置。`CTEX` 宏包同时也为这些自定义页面格式的包提供了以下宏供使用:

- `\CTEXthechapter`、`\CTEXthesection` 等章节编号 (见 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`, `fixskip`, `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`, `afterindent` 和 `fixskip` 这三个格式无效。

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}	$\text{\partname}\text{\space}$	原 \partname 为 Part
chapter	{第,章}	$\text{\chaptername}\text{\space}$	原 \chaptername 为 Chapter
section (beamer)	{}	$\text{\sectionname}\text{\space}$	原 \sectionname 为 $\text{\translate{Section}}$
section	同右	{}	
subsection (beamer)	{}	$\text{\subsectionname}\text{\space}$	原 \subsectionname 为 $\text{\translate{Subsection}}$
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 ★
paragraph/runin    ★
subparagraph/runin ★

```

New: 2015-06-27

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

`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

.../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

.../fixskip ★ fixskip = true|false

New: 2016-06-03

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

.../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）。

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/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。

`appendix/name` `name = {(前名字),(后名字)}`
`name = {(前名字)}`

Updated: 2014-03-08

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	<code>scheme = chinese</code>	实际定义	<code>scheme = plain</code>	实际定义
article	<code>\section</code>	<code>{}</code>		<code>{}</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>

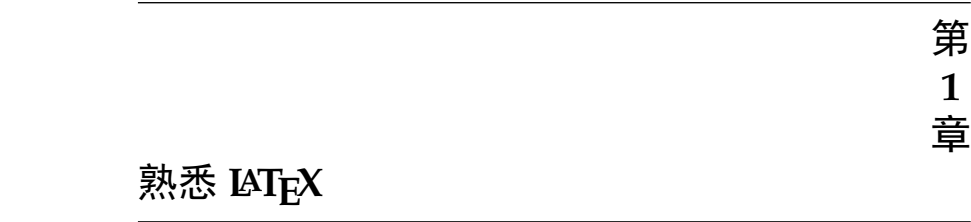
我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 19

```
\ctexset {
  chapter = {
    before skip = 0pt,
    fix skip    = true,
    format      = \Huge\bfseries,
    name format = \rule{\linewidth}{1bp}\par\bigskip\hfill\chaptername box,
    number      = \arabic{chapter},
    after name  = \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 8 节 实用命令

8.1 字号与间距

<code>\zihao</code>
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

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

$\backslash\text{chinese}$	$\backslash\text{chinese}$ $\{(\text{counter})\}$
Updated: 2016-05-01	$\backslash\text{pagenumbering}$ $\{\text{chinese}\}$
	$\backslash\text{chinese}$ 命令与 $\backslash\text{roman}$ 等命令的用法类似,作用在一个 \LaTeX 计数器上,将计数器的值以中文数字的形式输出。

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

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 $\text{T}\text{\TeX}$ 宏,不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 $\langle\text{number}\rangle$ 转为中文数字,最后将结果存储在 $\backslash(\text{macro})$ 里。对 $\backslash(\text{macro})$ 的定义是局部的,将它展开一次就可以得到转换结果。

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

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

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

8.3 杂项

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

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

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

pTeX 引擎的(大部分)功能。它为了兼容 pTeX 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,因而 C_TE_X 禁用了它在 L^AT_EX 格式下的大部分设置,只保留了必要的部分。同时修改了它的字体设置方式,使得相关命令与 xeCJK 宏包大致相同。

20150420 版以后的 LuaTeX-j_a 宏包开始支持竖排,但 C_TE_X 暂不支持竖排。

9.1 LuaTeX 下替代字体的设置

AlternateFont <hr/> New: 2014-04-14	<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/> New: 2014-04-14	<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/> New: 2014-04-14	<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 *family* 的替换字体设置。如果没有给定值,则作用于当前 CJK 字体族。清除与重置操作总是全局的。

第 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-jan 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。

11.5 C_TE_X 2.4.1 和 2.4.2

这两个选项已经被删除, 相应功能由新的选项 fixskip 提供。

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

part/fixbeforeskip
chapter/fixbeforeskip

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。
- indentfirst 宏包, 属于 tools 宏集。
- everyisel 宏包, 属于 ms 宏集。
- ulem 宏包。
- zhnumber 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- CJK 宏集。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包, 它还依赖 oberdiek 宏集。
- ➡ 以上是使用 pdf_LA_TE_X 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
 - graphics 宏集。

- graphics-cfg 宏包。
- graphics-def 宏包。
- environ 宏包, 它还依赖 trimspaces 宏包。
- ➡ 以上是使用 Xe_{La}TeX 编译时的依赖项。
- LuaTeX-jan 宏集, 它还依赖
 - adobemapping 宏包。
 - fontspec 宏包。
 - lualibs 宏包。
 - luaotfload 宏包。
 - luatexbase 宏包, 它还依赖 ctablestack 宏包。
 - oberdiek 宏集。
 - xkeyval 宏包。
- ➡ 以上是使用 Lua_{La}TeX 编译时的依赖项。
- zhmetrics-uptex 宏包。
- ➡ 以上是使用 up_{La}TeX 编译时的依赖项。

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

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

C_TE_X 宏集已被 T_EX Live 和 MiK_TE_X 收录, 若无特别理由, 我们强烈建议用户使用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 C_TE_X 宏集的 **TDS 安装包**,
2. 按目录结构将文件复制到 T_EX 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 T_EX 发行版的 ls-R 数据库以完成安装。

第 13 节 开发人员

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

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

第 14 节 代码实现

1 <@@=ctex>

宏包载入检查。

```

2 <*class|ctex>
3 \tl_const:Nx \c__ctex_version_tl
4 { \cs_if_exist_use:cF { ver@ \@currname . \@current } { 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 , everyisel }

```

14.1 内部函数与变量

```

\l__ctex_tmp_tl 58 \tl_new:N \l__ctex_tmp_tl
\l__ctex_tmp_int 59 \int_new:N \l__ctex_tmp_int
\l__ctex_tmp_dim 60 <|ctexheading>\dim_new:N \l__ctex_tmp_dim
\l__ctex_tmp_box 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 8.)

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 221     GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
UTF8 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 7.)

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 9.)

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 10.)

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

```

307 <!*beamer>

```

sub4section

```

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
\CTEXoptions

过时命令。

```

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}TeX 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   }
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

```

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

```

\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 `\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 lltjdefs.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: }

```

\fontspec_visible_space: 我们不使用 luatexja 对 fontspec 的补丁, 直接处理。

```

728 \cs_set_protected:Npn \fontspec_visible_space:
729 {
730   \font_glyph_if_exist:NnTF \font { "2423 }
731   { \ltjalchar "2423 \exp_stop_f: }
732   { \fontspec_visible_space_fallback: }
733 }

```

(End definition for \fontspec_visible_space:.)

对 unicode-math 的补丁主要是将 unicode-math-table.tex 中的数学符号设置为 lua-texja 中的数学字母。

```

734 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
735 {
736   \group_begin:
737   \cs_set_protected:Npn #1 ##1##2##3
738   { \ltjsetmathletter { ##1 } }
739   #2
740   \group_end:
741 }
742 \ctex_at_end_package:n { unicode-math }
743 {
744   \cs_if_exist:NTF \um_input_math_symbol_table:

```

```

745     {
746       \ctex_ltj_set_math_letter:NN
747       \um_sym:nnn
748       \um_input_math_symbol_table:
749     }
750     {
751       \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
752       \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
753         { \__um_switchto_literal: }
754       \ctex_ltj_set_math_letter:NN
755       \__um_sym:nnn
756       \__um_input_math_symbol_table:
757     }
758   }

```

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。

```

759 \ctex_at_end_package:nn { listings }
760 {
761   \use:x
762   {
763     \exp_not:N \RequirePackage { lltpj-listings }
764     \tl_set:Nn \exp_not:N \lstlistingname
765       { \exp_not:o { \lstlistingname } }
766     \tl_set:Nn \exp_not:N \lstlistlistingname
767       { \exp_not:o { \lstlistlistingname } }
768   }
769 }

```

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。

```

770 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
771   {
772     \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
773     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
774   }
775 \tl_new:N \CJK@family
776 \tl_new:N \l__ctex_ltj_current_font_tl
777 \tl_set:Nn \l__ctex_ltj_current_font_tl
778   { \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。

```

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

```

789   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
790   { \__ctex_ltj_select_font_aux: }

```

```

791 }
792 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
793 {
794   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
795   \cs_gset_nopar:Npx \font@name {#1}
796 }
797 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
798 { \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`。

```

799 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
800 {
801   \exp_after:wN \cs_if_exist:NF \font@name
802   {
803     \group_begin:
804     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
805     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
806     \define@newfont
807     \group_end:
808   }
809 }
810 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

(End definition for \ctex_ltj_pickup_font:.)

```

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

```

%%%%\jfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont#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`。

```

811 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
812 {
813   \get@external@font
814   \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
815   {
816     \tl_set:Nx \external@font
817     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
818   }
819   \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。

```

820   \font@name
821   \lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
822   \use:c { \f@encoding + \f@family }
823   \use:c { \curr@fontshape }
824 }

```

(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。

```
825 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
826 {
827   \ctex_ltj_if_alterate_shape_exist:nF { \curr@fontshape }
828   {
829     \group_begin:
830     \tl_set_eq:NN \CJK@family \f@family
831     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
832     {
833       \cs_gset_protected_nopar:Npx \subst@correction
834       {
835         \cs_new_eq:NN
836         \exp_not:c { \l__ctex_ltj_current_font_tl }
837         \font@name
838       }
839       \group_insert_after:N \group_insert_after:N
840       \group_insert_after:N \subst@correction
841     }
842     \group_end:
843   }
844 }
```

(End definition for `\ctex_ltj_subst_font:`)

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

```
845 \prg_new_conditional:Npnn \ctex_ltj_if_alterate_shape_exist:n #1 { T , F , TF }
846 {
847   \lua_now_x:n { luatexja.jfont.does_alt_set (' \lua_escape_x:n { #1 } ') }
848   \prg_return_true: \else: \prg_return_false: \fi:
849 }
```

(End definition for `\ctex_ltj_if_alterate_shape_exist:nTF`)

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

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

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

```
850 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
851 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }
```

(End definition for `__ctex_ltj_patch_external_font:w`)

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

```

852 \cs_new_protected_nopar:Npn \ctex_ltj_select_altername_font:
853 {
854   \ctex_ltj_if_altername_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
855   {
856     \lua_now_x:n
857     {
858       luatexja.jfont.output_alt_font_cmd
859       ('y', '\lua_escape_x:n { \l__ctex_ltj_current_shape_tl }')
860     }
861     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a ('f@size') }
862   }
863 }
864 \tl_new:N \l__ctex_ltj_current_shape_tl
865 \tl_set:Nn \l__ctex_ltj_current_shape_tl
866 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

(End definition for \ctex_ltj_select_altername_font:.)

```

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

```

867 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
868 {
869   \cs_if_exist:cF { #1/\f@size }
870   {
871     \group_begin:
872     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
873     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
874     \ctex_ltj_pickup_font:
875     \group_end:
876     \__ctex_ltj_pop_fontname:
877   }
878 }

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

879 <@@@=>

```

`\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。

```

880 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
881 {
882   \ltj@@getjfontnumber #1
883   \lua_now_x:n
884   {
885     luatexja.jfont.pickup_alt_font_b
886     ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
887   }
888 }

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

```

14.4.3.4 数学字体族

889 <@@@-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 \LaTeX 2_ϵ 编码名称或者字体命令。 \LaTeX 2_ϵ 字体命令的一般形式是:

`\(encoding)/<family>/<series>/<shape>`

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

```

890 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1

```



```

891 {
892   \lua_now_x:n
893   { \luaescape_x:n {#1}', '[~/]+' ) }
894   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
895 }

```

(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。

```

896 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
897 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
898 \group_begin:
899   \char_set_catcode_other:N M
900   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
901     { \ctex_ltj_if_jfont:nTF {#3} }
902 \group_end:

```

(End definition for \ctex_ltj_if_jfont_math:NTF.)

903 <@@=)

\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。

```

904 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
905 {
906   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
907     { \ctex_ltj_get_and_define_fonts_ja:nN }
908     { \ctex_ltj_get_and_define_fonts_al:nN }
909     {#1} #2
910 }
911 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
912 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
913 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
914 {
915   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
916   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
917   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
918   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
919   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
920   \ctex_ltj_pickup_font:
921   \tl_put_right:Nx \math@fonts
922     {
923       \ltj@setpar@global
924       \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
925       \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
926       \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
927     }
928 }

```

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

929 <@@=-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。

```

930 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
931 {
932   \mode_if_math:T
933   {
934     \math@bgroup
935     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
936     \ctex_ltj_math_group_hook:
937     \ctex_ltj_if_jfont_math:NTF #1
938       { \jfam } { \mathgroup } #2 \scan_stop:
939     \math@egroup
940   }

```

```

941 }
942 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
943 \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-ja 中与标点格式 plain 对应的 JFM 是 mono。
\l__ctex_ltj_jfm_tl
944 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
945 {
946   \str_if_eq:nnTF {#1} { plain }
947     { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
948     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
949 }
950 \tl_new:N \l__ctex_ltj_jfm_tl
951 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
952 \ctex_mono_jfm:o { \l__ctex_punct_tl }
(End definition for \ctex_mono_jfm:n and \l__ctex_ltj_jfm_tl.)

```

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

```

953 \tl_const:Nn \CJK@encoding { LTJY3 }
954 \DeclareFontEncoding { \CJK@encoding } { } { }
955 \use:x
956 {
957   \exp_not:N \DeclareFontSubstitution
958     { \CJK@encoding } { song } { \mddefault } { \updefault }
959 }
960 \lua_now_x:n { luatexja.fonts.add_kyenc_list('\CJK@encoding') }
961 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
962 { \tl_set_eq:NN \g_fonts_spec_encoding_tl \CJK@encoding }
963 \DeclareFontFamily { \CJK@encoding } { song } { }
964 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
965 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
966 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
967 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
968 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
969 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
970 { \CJK@encoding } { song } { \mddefault } { \updefault }
971 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
972 { \CJK@encoding } { song } { \bfdefault } { \updefault }
973 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
974 \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。

```

975 \newfontfeature { CID } { cid = #1 }
976 \newfontfeature { JFM } { jfm = #1 }
977 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

978 \keys_define:nn { fontspec-preparse-external }
979 {
980   NoEmbed .code:n =
981     { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
982 }
983 \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。

```

984 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
985 {
986   \group_begin:
987   \clist_clear:N \l__ctex_ltj_char_range_clist
988   \seq_clear:N \l__ctex_ltj_alternate_seq
989   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
990   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
991   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
992   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
993   {
994     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
995     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
996       {#1} { \l__ctex_ltj_font_options_clist }
997     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
998     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
999     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
1000   }
1001   \group_end:
1002 }
1003 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1004 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1005 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1006 \clist_new:N \l__ctex_ltj_font_options_clist
1007 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1008 {
1009   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1010   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1011 }

```

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

`\g__ctex_ltj_family_name_prop`
`\g__ctex_ltj_family_font_name_prop`
`\g__ctex_ltj_family_font_options_prop`

分别保存 **fontspec** 设置的字体族名、字体名称和字体选项。

```

1012 \prop_new:N \g__ctex_ltj_family_name_prop
1013 \prop_new:N \g__ctex_ltj_family_font_name_prop
1014 \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。

```

1015 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1016 {
1017   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1018   {
1019     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1020     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1021     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1022     {
1023       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1024       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1025       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1026       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1027     }
1028     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1029   }
1030 }
1031 \tl_new:N \l__ctex_ltj_tmp_tl
1032 \msg_new:nnn { ctex } { redefine-family }
1033 { 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。

```

1034 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1035 {
1036   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }
1037   {
1038     \group_begin:
1039     \__ctex_ltj_change_encoding:
1040     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1041     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1042     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1043     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1044     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1045     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1046     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1047     \__ctex_ltj_set_alternate_family:n {#1}
1048     \group_end:
1049   }
1050 }
1051 \tl_new:N \l__ctex_ltj_base_family_tl
1052 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1053 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1054 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1055 {
1056   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1057   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1058   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1059   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1060 }
1061 \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。

```

1062 \NewDocumentCommand \CJKfamily { m }
1063 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1064 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1065 {
1066   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1067   {
1068     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1069     \selectfont
1070   }
1071   { \__ctex_ltj_family_unknown_warning:n {#1} }
1072 }
1073 \tl_new:N \l_ctex_ltj_family_tl
1074 \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。

```

1075 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1076 {
1077   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1078   { \prg_return_true: }
1079   {
1080     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1081     {
1082       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1083       \prg_return_true:
1084     }
1085     { \prg_return_false: }
1086   }
1087 }
1088 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

(End definition for \ctex_ltj_family_if_exist:nTF.)

```

```

\__ctex_ltj_family_unknown_warning:n 1089 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1090 {
1091   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1092   {
1093     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1094     {
1095       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1096       \msg_warning:nnn { ctex } { family-unknown } {#1}
1097     }
1098   }
1099 }
1100 \seq_new:N \g__ctex_ltj_unknown_family_seq
1101 \msg_new:nnn { ctex } { family-unknown }
1102 {
1103   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1104   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1105 }
1106 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1107 {
1108   \str_case_x:nnF {#1}
1109   {
1110     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1111     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1112     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1113   }
1114   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1115   [...] \{...\}
1116 }
1117 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1118 {
1119   \str_case_x:nnF {#1}
1120   {
1121     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1122     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1123     \CJKttdefault { \token_to_str:N \CJKttdefault }
1124   }
1125   {#1}
1126 }

```

(End definition for __ctex_ltj_family_unknown_warning:n.)

```

\ctex_ltj_fontspec:nn 1127 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1128 {
1129   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1130   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1131   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1132   {
1133     \int_gincr:N \g__ctex_ltj_family_int
1134     \__ctex_ltj_fontspec:xnn
1135     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1136     {#1} {#2}
1137   }
1138 }
1139 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1140 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1141 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1142 {
1143   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1144   {
1145     \cs_if_free:cF
1146     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1147     {
1148       \cs_gset_eq:cc
1149       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1150       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1151       \cs_gset_eq:cc
1152       { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1153       { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }

```

```

1154     }
1155     \bool_set_false:N \l__ctex_ltj_add_alterate_bool
1156   }
1157   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1158   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1159   \ctex_ltj_switch_family:n {#1}
1160 }
1161 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1162 \prop_new:N \g__ctex_ltj_fontspec_prop
(End definition for \ctex_ltj_fontspec:nn.)

```

```

\ctex_ltj_add_font_features:n 1163 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1164 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1165 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1166 {
1167   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1168   {#1} \l__ctex_ltj_tmp_tl
1169   {
1170     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1171     {#1} \l__ctex_ltj_font_options_clist
1172     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1173     \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1174     \ctex_ltj_fontspec:xx
1175     { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1176     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1177   }
1178   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1179 }
1180 \bool_new:N \l__ctex_ltj_add_alterate_bool
1181 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1182 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1183 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1184 {
1185   \token_to_str:N \addCJKfontfeature (s)~ignored.\
1186   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1187 }

```

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

```

\setCJKfamilyfont 1188 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
\newCJKfontfamily 1189 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
\CJKfontspec 1190 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
\addCJKfontfeatures 1191 {
1192   \tl_set:Nx \l__ctex_ltj_tmp_tl
1193   { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1194   \cs_new_protected_nopar:Npx #2
1195   { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1196   \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1197 }
1198 \NewDocumentCommand \CJKfontspec { O { } m }
1199 {
1200   \ctex_ltj_fontspec:xx {#1} {#2}
1201   \tex_ignorespaces:D
1202 }
1203 \NewDocumentCommand \addCJKfontfeatures { m }
1204 {
1205   \ctex_ltj_add_font_features:x {#1}
1206   \tex_ignorespaces:D
1207 }
1208 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

(End definition for \setCJKfamilyfont and others.)

```

\setCJKmainfont 1209 \NewDocumentCommand \setCJKmainfont { O { } m }
\setCJKsansfont 1210 {
\setCJKmonofont 1211   \ctex_ltj_set_family:xxx { \CJKrmdefault } {#1} {#2}
\setCJKmathfont 1212   \normalfont
\defaultCJKfontfeatures 1213 }

```

```

1214 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1215 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1216 {
1217   \ctex_ljtj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1218   \normalfont
1219 }
1220 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1221 {
1222   \ctex_ljtj_set_family:xxx { \CJKttdefault } {#1} {#2}
1223   \normalfont
1224 }
1225 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1226 { \ctex_ljtj_set_family:xxx { \c__ctex_ljtj_math_tl } {#1} {#2} }
1227 \NewDocumentCommand \defaultCJKfontfeatures { m }
1228 { \clist_gset:Nn \g__ctex_ljtj_default_features_clist {#1} }
1229 \clist_new:N \g__ctex_ljtj_default_features_clist
1230 \@onlypreamble \setCJKmainfont
1231 \@onlypreamble \setCJKsansfont
1232 \@onlypreamble \setCJKmonofont
1233 \@onlypreamble \setCJKmathfont
1234 \@onlypreamble \setCJKromanfont
1235 \@onlypreamble \defaultCJKfontfeatures

```

(End definition for \setCJKmainfont and others.)

```

1236 \tl_if_exist:NF \CJKfamilydefault
1237 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1238 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1239 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1240 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1241 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1242 { \CJKfamily { \CJKrmdefault } }
1243 { }
1244 { \ctex_patch_failure:N \rmfamily }
1245 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1246 { \CJKfamily { \CJKsfdefault } }
1247 { }
1248 { \ctex_patch_failure:N \sffamily }
1249 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1250 { \CJKfamily { \CJKttdefault } }
1251 { }
1252 { \ctex_patch_failure:N \ttfamily }
1253 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1254 { \CJKfamily { \CJKfamilydefault } }
1255 { \cs_set_eq:NN \reset@font \normalfont }
1256 { \ctex_patch_failure:N \normalfont }

```

在导言区末尾更新 \CJKfamilydefault。

```

1257 \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。

```

1258 \cs_new_protected_nopar:Npn \ctex_ljtj_ensure_default_family:
1259 {
1260   \prop_if_empty:NF \g__ctex_ljtj_family_font_name_prop
1261   {
1262     \ctex_ljtj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ljtj_tmp_tl
1263     {
1264       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1265       { \use:n }
1266       {
1267         \ctex_ljtj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ljtj_tmp_tl
1268         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1269         { \use:n }
1270       }
1271     }
1272     \prop_map_inline:Nn \g__ctex_ljtj_family_font_name_prop

```



```

1273         {
1274             \prop_map_break:n
1275             { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1276         }
1277     }
1278 }
1279 \normalfont
1280 \ctex_ljtj_update_mathfont:
1281 }
1282 }

```

(End definition for \ctex_ljtj_ensure_default_family:.)

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

```

1283 \cs_new_protected_nopar:Npn \ctex_ljtj_update_mathfont:
1284 {
1285     \ctex_ljtj_family_if_exist:xNTF { \c__ctex_ljtj_math_tl } \l__ctex_ljtj_tmp_tl
1286     { \ctex_ljtj_update_mathfont:n { \l__ctex_ljtj_tmp_tl } }
1287     {
1288         \ctex_ljtj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ljtj_tmp_tl
1289         { \ctex_ljtj_update_mathfont:n { \l__ctex_ljtj_tmp_tl } }
1290     }
1291 }
1292 \cs_new_protected_nopar:Npn \ctex_ljtj_update_mathfont:n #1
1293 {
1294     \tl_const:Nx \c__ctex_ljtj_math_family_tl {#1}
1295     \DeclareSymbolFont { \c__ctex_ljtj_math_tl } { \CJK@encoding }
1296     { \c__ctex_ljtj_math_family_tl } { \mddefault } { \updefault }
1297     \cs_if_free:cTF
1298     { \CJK@encoding/\c__ctex_ljtj_math_family_tl/\bfdefault/\updefault }
1299     {
1300         \SetSymbolFont { \c__ctex_ljtj_math_tl } { bold } { \CJK@encoding }
1301         { \c__ctex_ljtj_math_family_tl } { \mddefault } { \updefault }
1302     }
1303     {
1304         \SetSymbolFont { \c__ctex_ljtj_math_tl } { bold } { \CJK@encoding }
1305         { \c__ctex_ljtj_math_family_tl } { \bfdefault } { \updefault }
1306     }
1307 }

```

(End definition for \ctex_ljtj_update_mathfont:.)

14.4.3.6 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1308 \keys_define:nn { ctex_ljtj / fontspec }
1309 {
1310     AlternateFont .code:n = \ctex_ljtj_set_alternate_seq:n {#1} ,
1311     AlternateFont .value_required:n = true ,
1312     CharRange .clist_set:N = \l__ctex_ljtj_char_range_clist ,
1313     CharRange .value_required:n = true
1314 }

```

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

\ctex_ljtj_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。

```

1315 \group_begin:
1316 \char_set_catcode_other:N \|
1317 \cs_set:Npn \__ctex_ljtj_tmp:w #1
1318 {
1319     \cs_new_protected:Npn \ctex_ljtj_set_alternate_seq:n ##1
1320     {
1321         \clist_if_empty:NT \l__ctex_ljtj_char_range_clist

```

```

1322         {
1323             \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1324             \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1325             \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1326             \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1327             { ! \tl_if_blank_p:n { ####1 } }
1328             \seq_concat:NNN \l__ctex_ltj_alternate_seq
1329             \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1330         }
1331     }
1332 }
1333 \char_set_catcode_active:N \l
1334 \__ctex_ltj_tmp:w { || }
1335 \group_end:
1336 \seq_new:N \l__ctex_ltj_tmp_seq
1337 \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。

```

1338 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1339 {
1340     \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1341     {
1342         \__ctex_ltj_check_family:n {#1}
1343         \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1344         { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1345         #3
1346     }
1347     { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1348 }

```

(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 保存由 AlternateFont 设置的替代字体序列。

```

1349 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1350 {
1351     \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1352     { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1353 }
1354 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1355 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1356 { m m m +0{ } m u{ \q_stop } }
1357 {
1358     \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1359     \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1360     \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1361     \tl_if_blank:nTF {#5}
1362     { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1363     {
1364         \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1365         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1366     }
1367     \use:x
1368     {
1369         \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1370         { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1371         { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1372         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1373     }
1374 }
1375 \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。

```

1376 \cs_new_protected_nopar:Npn \ctex_ljtj_set_alternate_family:nn #1#2
1377 {
1378   \__ctex_ljtj_update_family_uid:N \l__ctex_ljtj_font_options_clist
1379   \__ctex_ljtj_use_global_options:N \l__ctex_ljtj_font_options_clist
1380   \ctex_ljtj_set_alternate_family:coonn
1381   { \__ctex_ljtj_alternate_cs:n {#1} }
1382   { \l__ctex_ljtj_char_range_clist }
1383   { \l__ctex_ljtj_font_options_clist } {#2} {#1}
1384 }
1385 \cs_new_protected_nopar:Npn \ctex_ljtj_set_alternate_family:Nnnnn #1#2#3#4#5
1386 {
1387   \prop_get:NnNT \g__ctex_ljtj_family_name_prop {#5} \l__ctex_ljtj_base_family_tl
1388   { \ctex_ljtj_set_alternate_family:nnn {#2} {#3} {#4} }
1389   \ctex_ljtj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1390 }
1391 \cs_generate_variant:Nn \ctex_ljtj_set_alternate_family:Nnnnn { coo }

(End definition for \ctex_ljtj_set_alternate_family:nn)

```

\ctex_ljtj_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。

```

1392 \cs_new_protected_nopar:Npn \ctex_ljtj_save_alternate_family:Nnnn #1#2#3#4
1393 {
1394   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1395   \cs_gset_protected_nopar:Npx #1
1396   { \exp_not:o { #1 \ctex_ljtj_set_alternate_family:nnn {#2} {#3} {#4} } }
1397 }

(End definition for \ctex_ljtj_save_alternate_family:Nnnn)

```

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

```

1398 \cs_new_protected_nopar:Npn \ctex_ljtj_set_alternate_family:nnn #1#2#3
1399 {
1400   \group_begin:
1401   \__ctex_ljtj_change_encoding:
1402   \cs_set_eq:NN \CJKfamily \use_none:n
1403   \ctex_ljtj_swap_cs:NN
1404   \DeclareFontShape@ \ctex_ljtj_declare_alternate_shape:nnnnnn
1405   \tl_set:Nn \l__ctex_ljtj_char_range_clist {#1}
1406   \fontspec_set_family:Nnn \l__ctex_ljtj_alternate_family_tl {#2} {#3}
1407   \group_end:
1408 }
1409 \tl_new:N \l__ctex_ljtj_alternate_family_tl

(End definition for \ctex_ljtj_set_alternate_family:nnn)

```

\ctex_ljtj_swap_cs:NN 交换两个控制序列的意义。

```

1410 \cs_new_protected:Npn \ctex_ljtj_swap_cs:NN #1#2
1411 {
1412   \cs_set_eq:NN \__ctex_ljtj_tmp:w #1
1413   \cs_set_eq:NN #1 #2
1414   \cs_set_eq:NN #2 \__ctex_ljtj_tmp:w
1415   \cs_undefine:N \__ctex_ljtj_tmp:w
1416 }

(End definition for \ctex_ljtj_swap_cs:NN)

```

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

```

1417 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1418 \cs_new_protected_nopar:Npn \__ctex_ljtj_update_family_uid:N #1
1419 {

```

```

1420 \int_gincr:N \g__ctex_ltj_family_int
1421 \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1422 }
1423 \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>。

```

1424 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1425 {
1426   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1427   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1428   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1429   { \l_fontspec_family_tl } {#3} {#4}
1430 }
(End definition for \ctex_ltj_declare_alter_nate_shape:nnnnnn.)

```

\ctex_ltj_set_alter_nate_shape:Nnnnnnn 与 LuaTeX-jā 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。

```

1431 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1432 {
1433   \clist_map_inline:Nn #1
1434   {
1435     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1436     {
1437       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1438       \l__ctex_ltj_char_range_tl
1439     }
1440     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1441   }
1442   \__ctex_ltj_save_alter_nate_shape:cnn
1443   { \__ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1444   { luatexja.jfont.clear_alt_font_latex }
1445   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1446 }
(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-jā 一样使用 - 作为区间的分隔符。LuaTeX-jā 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。

```

1447 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1448 { m m > { \SplitArgument { \c_one } { -> } } m }
1449 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1450 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1451 {
1452   \ctex_ltj_set_alter_nate_shape:n
1453   {
1454     \IfNoValueTF {#4}
1455     { \int_eval:n {#3} , \int_eval:n {#3} , }
1456     {
1457       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1458       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1459     }
1460     '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1461     '\lua_escape_x:n { \CJK@encoding/#1 }'
1462   }
1463 }
1464 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:n #1
1465 {
1466   \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }

```

```

1467 \__ctex_ltj_save_altername_shape:cnn
1468 { \__ctex_ltj_altername_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1469 { luatexja.jfont.set_alt_font_latex } {#1}
1470 }

```

(End definition for \ctex_ltj_set_altername_shape:nnn.)

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

```

1471 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:nnN #1#2#3
1472 {
1473   \tl_map_inline:Nn #3
1474   {
1475     \ctex_ltj_set_altername_shape:n
1476     {
1477       ##1 ,
1478       '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1479       '\lua_escape_x:n { \CJK@encoding/#1 }'
1480     }
1481   }
1482 }

```

(End definition for \ctex_ltj_set_altername_shape:nnN.)

__ctex_ltj_save_altername_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。

```

1483 \cs_new_protected_nopar:Npn \__ctex_ltj_save_altername_shape:Nnn #1#2#3
1484 {
1485   \group_begin:
1486   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1487   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1488   \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1489   \cs_gset_protected_nopar:Npx #1
1490   { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1491   \group_end:
1492 }
1493 \cs_generate_variant:Nn \__ctex_ltj_save_altername_shape:Nnn { c }

```

(End definition for __ctex_ltj_save_altername_shape:Nnn.)

clearalternamefont
resetalternamefont

清除和重置操作总是全局的。

```

1494 \keys_define:nn { ctex }
1495 {
1496   clearalternamefont .code:n =
1497   { \clist_map_function:xN {#1} \ctex_ltj_clear_altername_font:n } ,
1498   resetalternamefont .code:n =
1499   { \clist_map_function:xN {#1} \ctex_ltj_reset_altername_font:n } ,
1500   clearalternamefont .default:n = \l_ctex_ltj_family_tl ,
1501   resetalternamefont .default:n = \l_ctex_ltj_family_tl
1502 }
1503 \cs_new_protected_nopar:Npn \ctex_ltj_clear_altername_font:n #1
1504 {
1505   \group_begin:
1506   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1507   {
1508     \cs_if_exist_use:cT { \__ctex_ltj_altername_cs:n { clear / #1 } }
1509     {
1510       \prop_gput:Nno \g__ctex_ltj_reset_altername_prop
1511       {#1} { \l__ctex_ltj_base_family_tl }
1512       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1513       \selectfont
1514     }
1515   }
1516   { \__ctex_ltj_family_unknown_warning:n {#1} }

```

```

1517 \group_end:
1518 }
1519 \cs_new_protected_nopar:Npn \ctex_ljt_reset_alternate_font:n #1
1520 {
1521 \group_begin:
1522 \prop_gpop:NnNT \g__ctex_ljt_reset_alternate_prop {#1} \CJK@family
1523 {
1524 \tl_set_eq:NN \l__ctex_ljt_base_family_tl \CJK@family
1525 \use:c { \__ctex_ljt_alternate_cs:n { reset / #1 } }
1526 \selectfont
1527 }
1528 \group_end:
1529 }
1530 \prop_new:N \g__ctex_ljt_reset_alternate_prop
1531 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalternatfont` and `resetalternatfont`. These functions are documented on page 26.)

declarecharrange 预先声明字符范围。

```

1532 \keys_define:nn { ctex }
1533 {
1534 declarecharrange .code:n = \ctex_ljt_declare_char_range:x {#1} ,
1535 declarecharrange .value_required:n = true
1536 }
1537 \cs_new_protected_nopar:Npn \ctex_ljt_declare_char_range:n #1
1538 { \clist_map_inline:nn {#1} { \__ctex_ljt_declare_char_range:nn ##1 } }
1539 \cs_generate_variant:Nn \ctex_ljt_declare_char_range:n { x }
1540 \cs_new_protected_nopar:Npn \__ctex_ljt_declare_char_range:nn #1#2
1541 { \use:x { \ctex_ljt_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

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

```

\ctex_ljt_declare_char_range:nn #1 是名字,#2 是范围。
\g__ctex_ljt_char_range_prop
1542 \cs_new_protected_nopar:Npn \ctex_ljt_declare_char_range:nn #1#2
1543 {
1544 \tl_clear:N \l__ctex_ljt_char_range_tl
1545 \clist_map_function:nN {#2} \ctex_ljt_save_char_range:n
1546 \prop_gput:Nno \g__ctex_ljt_char_range_prop {#1} { \l__ctex_ljt_char_range_tl }
1547 \ctex_ljt_def_char_range_key:n {#1}
1548 \tl_clear:N \l__ctex_ljt_char_range_tl
1549 }
1550 \tl_new:N \l__ctex_ljt_char_range_tl
1551 \prop_new:N \g__ctex_ljt_char_range_prop

```

(End definition for `\ctex_ljt_declare_char_range:nn` and `\g__ctex_ljt_char_range_prop`.)

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

```

1552 \NewDocumentCommand \ctex_ljt_save_char_range:n
1553 { > { \SplitArgument { \c_one } { -> } } m }
1554 { \ctex_ljt_save_char_range:nn #1 }
1555 \cs_new_protected_nopar:Npn \ctex_ljt_save_char_range:nn #1#2
1556 {
1557 \tl_put_right:Nx \l__ctex_ljt_char_range_tl
1558 { {
1559 \IfNoValueTF {#2}
1560 { \int_eval:n {#1} , \int_eval:n {#1} }
1561 {
1562 \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1563 \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1564 }
1565 } }
1566 }

```

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

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

```

1567 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1568 {
1569   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1570   {
1571     \keys_define:nn { ctex_ltj / fontspec }
1572     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1573   }
1574 }

(End definition for \ctex_ltj_def_char_range_key:n)

```

`\ctex_ltj_char_range_key:nn` 如果字符范围键没有值,则只设置的这个字符范围内的替代字体。

```

1575 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1576 {
1577   \tl_if_blank:nTF {#2}
1578   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1579   {
1580     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1581     {
1582       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1583       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1584     }
1585   }
1586 }

(End definition for \ctex_ltj_char_range_key:nn)

```

`__ctex_ltj_char_range_parse_feature:w` 可以使用加方括号的方式,通过文件名来调用字体。这容易与字体选项混淆。例如,需要将 `[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]`。

```

1587 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1588 { +o o u { \q_stop } }
1589 {
1590   \exp_args:NNf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1591   {
1592     \IfNoValueTF {#1} { {#3} }
1593     {
1594       \IfNoValueTF {#2}
1595       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] {#3} } }
1596       { [ { #1 } ] { [ #2 ] } }
1597     }
1598   }
1599   \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1600 }

(End definition for \__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
}

```

```

    end
  }

1601 \AtBeginDocument
1602 {
1603   \ctex_appto_cmd:NnnTF \verbatim@font
1604   { \char_set_catcode_letter:n { 64 } }
1605   { \CTEX@verbatim@font@hook }
1606   { }
1607   { \ctex_patch_failure:N \verbatim@font }
1608 }
1609 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1610 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

1611 <@@= >

```

`\@@italiccorr` \LaTeX 的倾斜校正也要重新定义。

```

1612 \cs_set_eq:NN \@@italiccorr \/

(End definition for \@@italiccorr.)

1613 <@@=ctex >

1614 </luatex >

```

14.4.4 ctex-engine-uptex.def

```

1615 <*uptex|aptex >

```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```

1616 \tl_set:Nn \CJKrmdefault { rm }
1617 \tl_set:Nn \CJKsfdefault { sf }
1618 \tl_set:Nn \CJKttdefault { tt }
1619 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1620 \tl_put_left:Nn \rmfamily { \kanjifamily { \CJKrmdefault } }
1621 \tl_put_left:Nn \sffamily { \kanjifamily { \CJKsfdefault } }
1622 \tl_put_left:Nn \ttfamily { \kanjifamily { \CJKttdefault } }
1623 \NewDocumentCommand \CJKfamily { m }
1624 {
1625   \kanjifamily { #1 }
1626   \selectfont
1627 }
1628 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```

在导言区末尾更新 `\CJKfamilydefault`。

```

1629 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

`\em` 取消 \upLaTeX 对 `\em` 使用 `\mcfamily`、`\gtfamily` 命令的重定义, 恢复 $\text{\LaTeX}2_{\epsilon}$ 对 `\em` 的原始定义。如果用户已经重定义了 `\em`, 则新定义保持不变。 $\text{\upLaTeX} 2016/05/07$ 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。

```

1630 \ctex_patch_cmd_once:NnnTF \em
1631 { \ExplSyntaxOff }
1632 { \eminnershape \else \gtfamily \itshape }
1633 { \eminnershape \else \itshape }
1634 { }
1635 {
1636   \ctex_patch_cmd:Nnn \em
1637   { \mcfamily \upshape \else \gtfamily \itshape }
1638   { \eminnershape \else \itshape }
1639 }
1640 \cs_set_nopar:Npn \eminnershape { \upshape }

(End definition for \em.)

```


`\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。

```

1641 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1642 {
1643   \DeclareKanjiFamily{JY2}{#1}{}
1644   \DeclareKanjiFamily{JT2}{#1}{}
1645   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1646   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1647   \tl_if_empty:nF { #3 }
1648   {
1649     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1650     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1651   }
1652 }
1653 \@onlypreamble \ctex_set_upfamily:nnn
(End definition for \ctex_set_upfamily:nnn.)

```

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

```

1654 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1655 {
1656   \ctex_set_zhmap:n
1657   {
1658     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1659     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1660     \tl_if_empty:nF { #3 }
1661     {
1662       \special{ pdf:mapline~ #1b-h~ UniGB-UTF16-H~ #3 }
1663       \special{ pdf:mapline~ #1b-v~ UniGB-UTF16-V~ #3 }
1664     }
1665   }
1666 }
1667 \@onlypreamble \ctex_set_upmap:nnn
(End definition for \ctex_set_upmap:nnn.)

```

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

```

1668 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1669 {
1670   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1671   \ctex_set_upmap:nnn { upserifit } { #3 } {}
1672   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1673   \ctex_set_upmap:nnn { upmono } { #6 } {}
1674 }
1675 \@onlypreamble \ctex_set_upfonts:nnnnnn
(End definition for \ctex_set_upfonts:nnnnnn.)

```

`\@EverySelectfont@Init` `everyysel` 宏包 (2011/10/28) 未考虑 `upLATEX` 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。

```

1676 \cs_set_nopar:Npn \@EverySelectfont@Init
1677 {
1678   \ctex_patch_cmd:Nnn \selectfont
1679   { \enc@update }
1680   {
1681     \@EverySelectfont@EveryHook
1682     \@EverySelectfont@AtNextHook
1683     \gdef \@EverySelectfont@AtNextHook { }
1684     \enc@update

```

```

1685     }
1686     \@ifpackageloaded { tracefmt }
1687     {
1688         \EverySelectfont
1689         {
1690             \ifnum \tracingfonts > \tw@
1691                 \@font@info { Switching~ to~ \font@name }
1692             \fi
1693         }
1694     }
1695     { }
1696     \let \@EverySelectfont@Init \undefined
1697 }

```

(End definition for \@EverySelectfont@Init.)

1698 </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。

```

1699 <*pdf|luatex|uptex|aptex>
1700 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1701 {
1702     \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1703     {
1704         \group_begin:
1705         \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1706         \tl_gset:Nx \CJKfamilydefault
1707         {
1708             \str_case:onF { \familydefault }
1709             {
1710                 { \rmdefault } { \exp_not:N \CJKrmdefault }
1711                 { \sfdefault } { \exp_not:N \CJKsfdefault }
1712                 { \ttdefault } { \exp_not:N \CJKttdefault }
1713             }
1714             { \CJKfamilydefault }
1715         }
1716         \group_end:
1717     }

```

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

```

1718 <*luatex>
1719     \ctex_ljtj_ensure_default_family:
1720 </luatex>
1721 }

```

(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。

```

1722 \tl_new:N \l__ctex_family_default_init_tl
1723 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1724 \tl_set:Nx \l__ctex_family_default_init_tl
1725 {
1726     \exp_not:N \__ctex_family_default_wrap:n
1727     { \exp_not:o { \CJKfamilydefault } }
1728 }
1729 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1730 </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 LuaTeX 下直接用调用 `os.name` 来判断。

```

1731 <*luatex>
1732 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1733 {
1734   \tl_gset:Nx \g__ctex_fontset_tl
1735   {
1736     \lua_now_x:n
1737     {
1738       if ~ os.name == 'windows' then ~
1739         tex.sprint ( 'windows' )
1740       elseif ~ os.name == 'macosx' then ~
1741         tex.sprint ( 'mac' )
1742       else ~
1743         tex.sprint ( 'fandol' )
1744       end
1745     }
1746   }
1747 }
1748 </luatex>

```

ApTeX 可以使用 `\ngostype` 来判断。

```

1749 <*aptex>
1750 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1751 {
1752   \tl_gset:Nx \g__ctex_fontset_tl
1753   {
1754     \str_case:onF { \ngostype }
1755     {
1756       { Win32 } { windows }
1757       { Win64 } { windows }
1758       { Darwin } { mac }
1759     }
1760     { fandol }
1761   }
1762 }
1763 </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!

```

1764 <*xetex|pdfTeX|uptex>
1765 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1766 {
1767   \file_if_exist:nTF { /dev/null }
1768   {
1769     \file_if_exist:nTF { nul: }
1770     {
1771       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1772       { \ctex_if_macosx:TF { mac } { windows } }
1773       { \ctex_if_macosx:TF { mac } { fandol } }
1774     }
1775     { \ctex_if_macosx:TF { mac } { fandol } }
1776   }
1777   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1778 }
1779 \str_const:Nx \c__ctex_upper_case_file_str
1780 { \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。

```

1781 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2

```

```

1782 {
1783   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1784     { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1785     { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1786 }
1787 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1788 </xetex|pdfetex|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。

```

1789 <*class|ctex>
1790 \ifpackageloaded { hyperref }
1791 {
1792   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1793     { \hypersetup {#1} }
1794 }
1795 {
1796   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1797     { \PassOptionsToPackage {#1} { hyperref } }
1798 }
1799 </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。

```

1800 <*pdfetex>
1801 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1802 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1803 {
1804   \ctex_hypersetup:n { CJKbookmarks = true }
1805   \sys_if_output_pdf:TF
1806     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1807     {
1808       \ctex_at_end_package:nn { hyperref }
1809       {
1810         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1811         {
1812           \AtBeginShipoutFirst
1813             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1814         }
1815         { \RequirePackage { xCJK2uni } }
1816       }
1817     }
1818 }
1819 { \ctex_hypersetup:n { pdfencoding = unicode } }
1820 </pdfetex>

```

在 `XYTeX` 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 `hxdetex.def` 中设置了 `\Hy@unicodetrue`, 从而书签总是会被 `\HyPsd@ConvertToUnicode`

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

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@unicodedefalse
          }{%
            \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 XDVIPDFMX 的警告¹⁸:

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

X_ƎT_ƎX 的维护者 Khaled Hosny 已经注意到了这个问题¹⁹。需要注意的是, `hxetex.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_T_ƎX 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 X_ƎT_ƎX 下更保险的 `pdfencoding` 选项来设置。为了与 X_ƎT_ƎX 下的行为一致(使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 LuaT_ƎX 下也启用这个选项。

```
1821 <*xetex|luatex>
1822 \ctex_hypersetup:n { pdfencoding = unicode }
1823 </xetex|luatex>
```

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

```
1824 <*uptex|aptex>
1825 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1826 \ctex_at_end_package:nn { hyperref }
1827 {
1828   \AtBeginShipoutFirst{ \special{ pdf:tounicode~ UTF8-UTF16 } }
1829 }
1830 </uptex|aptex>

1831 <*pdfƎtex|xetex|luatex|uptex|aptex>
```

14.4.8 CJKfntef、xeCJKfntef 相关设置

C_T_ƎX 宏集对 pdfT_ƎX 与 X_ƎT_ƎX 引擎, 分别载入 CJKfntef 或 xeCJKfntef 宏包, 并关闭宏包默认的彩色等多余格式。

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

```
1832 <*pdfƎtex>
1833 \RequirePackage { CJKfntef }
1834 \normalem
```

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

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

```

1835 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1836 { \tl_clear:c { CJK#1color } }
1837 </pdfTeX>
1838 <*/xetex>
1839 \RequirePackage { xeCJKfntef }
1840 \ifpackagealater { xeCJKfntef } { 2014/11/04 }
1841 {
1842   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1843   { \xeCJKsetup { #1 / format = { } } }
1844 }
1845 {
1846   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1847   { \tl_clear:c { CJK#1color } }
1848 }
1849 </xetex>
1850 <*/luatex|uptex|aptex>
1851 \msg_new:nnn { ctex } { fntef-not-available }
1852 <luatex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1853 <uptex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
1854 <aptex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1855 \msg_warning:nn { ctex } { fntef-not-available }
1856 </luatex|uptex|aptex>
1857 \clist_map_inline:nn
1858 { underdot , underline , underdblline , underwave , sout , xout }
1859 <*/pdfTeX|xetex>
1860 {
1861   \__ctex_clear_fntef_color:n {#1}
1862   \cs_new_protected_nopar:cpx { CTEX#1 }
1863   {
1864     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1865     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1866     \exp_not:c { CJK#1 }
1867   }
1868 }
1869 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1870 {
1871   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1872   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1873   \CJKfilltwosides
1874 }
1875 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1876 </pdfTeX|xetex>
1877 <*/luatex|uptex|aptex>
1878 { \cs_new_eq:cN { CTEX#1 } \use:n }
1879 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1880 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1881 </luatex|uptex|aptex>
1882 <*/pdfTeX>
1883 \clist_map_inline:nn
1884 {
1885   underdotbasesep , underdotsep , underlinebasesep ,
1886   underlinesep , underdbllinesep , underdbllinebasesep ,
1887   underwavebasesep , underwavesep , southeight ,
1888   underdotcolor , underwavecolor , underlinecolor ,
1889   underdbllinecolor , soutcolor , xoutcolor
1890 }
1891 {
1892   \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1893   \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1894 }
1895 </pdfTeX>

```

14.4.9 \ccwd 的更新

```

\ctex_update_ccwd: 1896 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1897 <*/pdfTeX|xetex>

```

```

1898 {
1899   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1900   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1901 }
1902 </pdf|xetex>
1903 <*luatex>
1904 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1905 </luatex>
1906 <*uptex|aptex>
1907 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1908 </uptex|aptex>
1909 \dim_new:N \ccwd
(End definition for \ctex_update_ccwd: and \ccwd.)

```

\ctex_update_ccglue: 更新字间距。

```

1910 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1911 <*pdf|xetex>
1912 {
1913   \cs_set_protected_nopar:Npn \CJKglue
1914   { \skip_horizontal:N \l__ctex_ccglue_skip }
1915 }
1916 </pdf|xetex>
1917 <*xetex>
1918 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1919 </xetex>
1920 <*luatex>
1921 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1922 </luatex>
1923 <*uptex|aptex>
1924 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1925 </uptex|aptex>
1926 \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

```

1927 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1928 {
1929 <*pdf|xetex>
1930   \if_meaning:w \CJKglue \__ctex_ccglue:
1931   \prg_return_false: \else: \prg_return_true: \fi:
1932 </pdf|xetex>
1933 <*luatex>
1934   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1935   { \prg_return_false: } { \prg_return_true: }
1936 </luatex>
1937 <*uptex|aptex>
1938   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1939   { \prg_return_false: } { \prg_return_true: }
1940 </uptex|aptex>
1941 }

```

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

```

1942 <*pdf|xetex>
1943 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1944 </pdf|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 pdf_{TEX} 和 X_YTEX 下, 直接使用 \f@size\p@ 作为汉字的宽度, 这应该对大多数汉字字体都成立, 但不适用于诸如“方正兰亭黑长”之类的特殊字体。在 X_YTEX 可以用 \fontcharwd 来改进。而在 pdf_{TEX}

下,若使用 `zhmetrics` 技术,所有的汉字共享同一个 `.tfm`,`\fontcharwd` 也就没有意义。在 `LuaTeX` 下,`LuaTeX-j` 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`。

```

1945 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1946 <pdf|xetex> { \dim_set:Nn \ccwd { \f@size \p@ } }
1947 <luatex> { \dim_set:Nn \ccwd { \zw } }
1948 <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-j` 的字体设置在导言区也可用,我们还需要在这里手工修改 `\selectfont`。`everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`,所以会给出一个警告。为了消除这个警告,在它检查之前,还原本来定义。

```

1949 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1950 {
1951   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1952   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1953 }
1954 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1955 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1956 \ctex_preto_cmd:NnnTF \selectfont { \ExplSyntaxOff }
1957 { \CTEX@selectfont@hook }
1958 {
1959   \tl_put_left:Nn \@EverySelectfont@Init
1960   { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1961 }
1962 { \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。

```

1963 <*pdf|tex>
1964 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1965 </pdf|tex>
1966 <*xetex|luatex|uptex|aptex>
1967 \EverySelectfont { \CTEX@selectfont@hook }
1968 </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。

```

1969 <*luatex>
1970 \ctex_add_to_selectfont:n
1971 {
1972   \ctex_ltj_select_font:
1973   \ctex_ltj_select_alternate_font:
1974 }
1975 \tl_set:Nn \CJK@family { song } \selectfont
1976 \tl_clear:N \CJK@family
1977 </luatex>

```

`\ctex_update_xkanjiskip:` `\l__ctex_xkanjiskip_skip` `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。

```

1978 <*luatex|uptex|aptex>

```



```

1979 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
1980 {
1981   \skip_if_eq:nnT
1982   <luatex> { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
1983   <uptex|aptex> { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
1984   {
1985     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
1986     <luatex> \ltjsetxkanjiskip \l__ctex_xkanjiskip_skip
1987     <uptex|aptex> \skip_set_eq:NN \ptex_xkanjiskip:D \l__ctex_xkanjiskip_skip
1988   }
1989 }
1990 \tl_new:N \l__ctex_xkanjiskip_tl
1991 \tl_set:Nn \l__ctex_xkanjiskip_tl
1992 <luatex> { .25\zw plus 1pt minus 1pt }
1993 <uptex|aptex> { .25\zw plus 1pt minus 1pt }
1994 \skip_new:N \l__ctex_xkanjiskip_skip
1995 \skip_set:Nn \l__ctex_xkanjiskip_skip
1996 <luatex> { \ltjgetparameter { xkanjiskip } }
1997 <uptex|aptex> { \ptex_xkanjiskip:D }

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

1998 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
1999 </luatex|uptex|aptex>

```

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

```

2000 \keys_define:nn { ctex }
2001 {
2002   <*pdf|xtex>
2003     space .choice: ,
2004     space / true .code:n =
2005     <pdf|xtex> { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
2006     <xtex> { \xeCJKsetup { CJKspace = true } } ,
2007     space / auto .code:n =
2008     <pdf|xtex> { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
2009     <xtex> { \xeCJKsetup { CJKspace = false } } ,
2010     space / false .code:n =
2011     <pdf|xtex> { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
2012     <xtex> { \xeCJKsetup { CJKspace = false } } ,
2013     space .default:n = { true } ,
2014     space .initial:n = { auto }
2015   </pdf|xtex>
2016   <*luatex|uptex|aptex>
2017     space .code:n =
2018     { \msg_warning:nn { ctex } { invalid-option } }
2019   </luatex|uptex|aptex>
2020 }

```

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

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-jaf 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。

```

2021 \keys_define:nn { ctex }
2022 {
2023   punct .code:n =
2024   {
2025     \tl_set:Nx \l__ctex_punct_tl { #1 }
2026     <pdf|xtex> \punctstyle { \l__ctex_punct_tl }
2027     <xtex> \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2028     <luatex> \ctex_mono_jfm:o { \l__ctex_punct_tl }
2029     <uptex|aptex> \msg_warning:nn { ctex } { invalid-option }
2030   } ,
2031   punct .default:n = { quanjiao } ,
2032 }

```

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

X_YLaTeX、LuaLaTeX 和 upLaTeX 总是使用 UTF8 编码。

```
2033 <*xetex|luatex|uptex|aptex>
2034 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2035 </xetex|luatex|uptex|aptex>

2036 </pdfTeX|xetex|luatex|uptex|aptex>
2037 <*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。

```
2038 \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。

```
2039 \keys_define:nn { ctex }
2040 {
2041   autoindent .choice: ,
2042   autoindent .default:n = { true } ,
2043   autoindent / true .code:n =
2044   {
2045     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2046     \ctex_select_size:
2047   } ,
2048   autoindent / false .code:n =
2049   { \tl_clear:N \l__ctex_autoindent_tl } ,
2050   autoindent / unknown .code:n =
2051   {
2052     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2053     \ctex_select_size:
2054   }
2055 }
```

(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。

```
2056 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2057 \cs_new_protected_nopar:Npn \ctex_select_size:
2058 { \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。

```
2059 \cs_new_protected_nopar:Npn \ctex_update_size:
2060 {
2061   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2062   {
2063     \ctex_update_stretch:
2064     \ctex_update_parindent:
2065   }
2066   { \ctex_update_ziju: }
2067 }
2068 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2069 \tl_new:N \l__ctex_ziju_tl
2070 \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。

```
2071 \ctex_add_to_selectfont:n
2072 { \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`。

```
2073 \keys_define:nn { ctex }
2074 {
2075   linestretch .code:n =
2076   {
2077     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2078     \ctex_select_size:
2079   } ,
2080   linestretch .value_required:n = true
2081 }
2082 \tl_new:N \l__ctex_line_stretch_tl
2083 \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 `everysel` 的钩子位于 `\size@update` 之前, `\baselineskip` 还未更新, 不能直接使用它。

```
2084 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2085 {
2086   \ctex_update_em_unit:
2087   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2088   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2089   {
2090     \ctex_if_ccglue_touched:TF
2091     { \ctex_update_ccwd: }
2092     {
2093       \dim_set:Nn \l__ctex_tmp_dim
2094       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2095       \skip_set:Nn \l__ctex_ccglue_skip
2096       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2097       \ctex_update_ccglue:
2098     }
2099   }
2100   {
2101     \int_set:Nn \l__ctex_tmp_int
2102     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2103     \tex_divide:D \l__ctex_tmp_int \ccwd
2104     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2105     {
2106       \skip_set:Nn \l__ctex_ccglue_skip
2107       {
2108         \c_zero_dim plus \dim_eval:n
2109         {
2110           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2111           \l__ctex_tmp_int
2112         }
2113       }
2114     }
2115   }
```

```

2115         { \skip_zero:N \l__ctex_ccglue_skip }
2116         \ctex_update_ccglue:
2117     }
2118 }

```

(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。

```

2119 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2120 {
2121     \tl_if_empty:NF \l__ctex_autoindent_tl
2122     {
2123         \dim_compare:nNnF \parindent = \c_zero_dim
2124         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2125     }
2126 }

```

(End definition for \ctex_update_parindent:.)

\ziju 若参数为 0, 则恢复正常间距。

```

2127 \NewDocumentCommand \ziju { m }
2128 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2129 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2130 {
2131     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2132     \ctex_select_size:
2133 }

```

(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。

```

2134 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2135 {
2136     \ctex_update_em_unit:
2137     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2138     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2139     \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。

```

2140     {
2141         \dim_set:Nn \l__ctex_tmp_dim
2142         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2143         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2144         \tex_divide:D \l__ctex_tmp_int \ccwd
2145         \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。

```

2146         \dim_compare:nNnF \parindent = \c_zero_dim
2147         {
2148             \int_compare:nNnF \l__ctex_tmp_int < \c_three
2149             { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2150         }
2151         \skip_set:Nn \l__ctex_ccglue_skip
2152         {
2153             \l__ctex_ziju_dim
2154             plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2155             minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2156             { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2157         }
2158     }

```

```

2159     { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2160     \ctex_update_ccglue:

```

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

```

2161     \ctex_update_parindent:
2162   }
2163   \dim_new:N \l__ctex_ziju_dim
(End definition for \ctex_update_ziju:.)

```

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

```

2164 \NewDocumentCommand \CTEXindent { }
2165 {
2166   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2167   { \parindent is~ set~ to~ 2\ccwd. }
2168   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2169 }
2170 \NewDocumentCommand \CTEXnoindent { }
2171 {
2172   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2173   { \parindent is~ set~ to~ 0pt. }
2174   \dim_zero:N \parindent
2175 }
(End definition for \CTEXindent and \CTEXnoindent. These functions are documented on page 29.)

```

14.6 中文数字与日期

```

2176 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2177 \RequirePackage { zhnumber }

```

$\backslash\text{chinese}$

```

2178 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2179 \cs_new_eq:NN \@chinese \@zhnum
2180 \cs_new_eq:NN \Chinese \chinese
2181 \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 $\backslash\text{chinese}$ 、 $\backslash\text{zhnum}$ 和 $\backslash\text{zhdig}$ 。

```

2182 \ctex_at_end_package:nn { enumitem }
2183 {
2184   \cs_if_free:NF \AddEnumerateCounter
2185   {
2186     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2187     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2188     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2189   }
2190 }

```

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

```

2191 \NewDocumentCommand \CTEXnumber { m m }
2192 { \protected@edef #1 { \zhnumber {#2} } }
2193 \NewDocumentCommand \CTEXdigits { m m }
2194 { \protected@edef #1 { \zhdigits {#2} } }
(End definition for \CTEXnumber and \CTEXdigits. These functions are documented on page 24.)

```

`today`

```

2195 \cs_set_eq:NN \CTEX@todayold \today
2196 \keys_define:nn { ctex }
2197 {
2198   today .choice: ,
2199   today / old .code:n =
2200   { \cs_set_eq:NN \today \CTEX@todayold } ,

```

```

2201   today / small   .code:n =
2202   {
2203       \cs_set_eq:NN \today \zhtoday
2204       \zhnumsetup { time = Arabic }
2205   } ,
2206   today / big     .code:n =
2207   {
2208       \cs_set_eq:NN \today \zhtoday
2209       \zhnumsetup { time = Chinese }
2210   } ,
2211   today / unknown .code:n =
2212   { \msg_error:nxx { ctex } { today-undef } {#1} }
2213 }
2214 \msg_new:nnnn { ctex } { today-undef }
2215 { Today~format~`#1'~is~undefined. }
2216 { 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

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

```

2217 \tl_if_exist:NF \proofname
2218 {
2219     \tl_new:N \proofname
2220     \tl_set:Nn \proofname { Proof }
2221 }

```

(End definition for `\proofname`.)

```

2222 \keys_define:nn { ctex }
2223 {
2224     contentsname .tl_set:N = \contentsname ,
2225     listfigurename .tl_set:N = \listfigurename ,
2226     listtablename .tl_set:N = \listtablename ,
2227     figurename .tl_set:N = \figurename ,
2228     tablename .tl_set:N = \tablename ,
2229     abstractname .tl_set:N = \abstractname ,
2230     indexname .tl_set:N = \indexname ,
2231     appendixname .tl_set:N = \appendixname ,
2232     proofname .tl_set:N = \proofname ,
2233     <article> bibname .tl_set:N = \refname
2234     <book|report> bibname .tl_set:N = \bibname
2235     <*beamer>
2236     algorithmname .tl_set:N = \algorithmname ,
2237     bibname .tl_set:N = \bibname ,
2238     refname .tl_set:N = \refname ,
2239     continuation .tl_set:N = \insertcontinuationtext
2240     </beamer>
2241 }

```

```

2242 <*ctex>
2243 \msg_new:nnn { ctex } { ctexbibname }
2244 {
2245     Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.\\
2246     The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2247 }
2248 \tl_if_exist:NTF \insertcontinuationtext
2249 {
2250     \keys_define:nn { ctex }
2251     {
2252         algorithmname .tl_set:N = \algorithmname ,
2253         bibname .tl_set:N = \bibname ,
2254         refname .tl_set:N = \refname ,
2255         continuation .tl_set:N = \insertcontinuationtext
2256     }

```

```

2257 }
2258 {
2259   \tl_if_exist:NTF \bibname
2260   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2261   {
2262     \tl_if_exist:NTF \refname
2263     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2264     {
2265       \msg_warning:nn { ctex } { ctexbibname }
2266       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2267     }
2268   }
2269 }
2270 </ctex>

2271 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```
2272 <*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。

```

2273 <*article|book|report>
2274 \seq_new:N \c__ctex_section_headings_seq
2275 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2276 { section , subsection , subsubsection , paragraph , subparagraph }
2277 </article|book|report>

(End definition for \c__ctex_section_headings_seq.)

```

```

\c__ctex_headings_seq 2278 \seq_new:N \c__ctex_headings_seq
2279 <*article|book|report>
2280 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
2281 <book|report>\seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2282 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2283 </article|book|report>
2284 <*beamer>
2285 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2286 { part , section , subsection }
2287 </beamer>

(End definition for \c__ctex_headings_seq.)

```

```

\__ctex_initial_heading:n 2288 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2289 {
2290   \tl_new:c { CTEX@pre#1 }
2291   \tl_new:c { CTEX@post#1 }
2292   \tl_const:cx { CTEXthe#1 }
2293   {
2294     \exp_not:c { CTEX@pre#1 }
2295     \exp_not:c { CTEX@the#1 }
2296     \exp_not:c { CTEX@post#1 }
2297   }
2298   \tl_const:cx { CTEX@#1name }
2299   {
2300     \group_begin:
2301     \exp_not:c { CTEX@#1nameformat }
2302     {
2303       \exp_not:c { CTEX@pre#1 }
2304       \exp_not:N \tl_if_empty:NTF

```

```

2305         \exp_not:c { CTEX@#1@numberformat }
2306         { \exp_not:c { CTEX@the#1 } }
2307         {
2308             \group_begin:
2309             \exp_not:c { CTEX@#1@numberformat }
2310             \exp_not:c { CTEX@the#1 }
2311             \group_end:
2312         }
2313         \exp_not:c { CTEX@post#1 }
2314     }
2315 \group_end:
2316 }
2317 }

```

(End definition for _ctex_initial_heading:n)

```

\_ctex_def_heading_keys:n 2318 \cs_new_protected_nopar:Npn \_ctex_def_heading_keys:n #1
2319 {
2320     \tl_put_right:Nx \l__ctex_tmp_tl
2321     {
2322         #1 .meta:nn = { ctex / #1 } { #####1 } ,
2323         #1 / name .code:n =
2324         { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2325         #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2326         #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2327         #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2328         #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2329         #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2330         #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2331         #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2332         #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
2333         #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2334         #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2335         #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2336         <*article|book|report>
2337         #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2338         #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2339         #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2340         </article|book|report>
2341         #1 / format+ .code:n =
2342         { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2343         #1 / nameformat+ .code:n =
2344         { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2345         #1 / numberformat+ .code:n =
2346         { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2347         #1 / aftername+ .code:n =
2348         { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2349         #1 / titleformat+ .code:n =
2350         { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2351         #1 / aftertitle+ .code:n =
2352         { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2353         #1 / numbering .initial:n = true ,
2354         #1 / beforeskip .initial:n = \c_zero_skip ,
2355         #1 / afterskip .initial:n = \c_zero_skip ,
2356         #1 / indent .initial:n = \c_zero_dim ,
2357         #1 / beforeskip .value_required:n = true ,
2358         #1 / afterskip .value_required:n = true ,
2359         #1 / indent .value_required:n = true ,
2360     }
2361 }

```

(End definition for _ctex_def_heading_keys:n)

ctex_assign_heading_name:nn name 的值是一个至多两个元素的逗号分隔列表。由于 L^AT_EX3 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。

```

2362 \NewDocumentCommand \ctex_assign_heading_name:nn
2363   { m > { \SplitArgument { \c_one } { , } } +m }
2364   { \__ctex_assign_heading_name:nnn {#1} #2 }
2365 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2366   {
2367     \tl_set:cn { CTEX@pre#1 } {#2}
2368     \IfNoValueTF {#3}
2369       { \tl_clear:c { CTEX@post#1 } }
2370       { \tl_set:cn { CTEX@post#1 } {#3} }
2371   }

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

```

`part/pagestyle`
`chapter/pagestyle`

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2372 \group_begin:
2373 <*book|report>
2374 \tl_set:Nn \l__ctex_tmp_tl
2375   {
2376     part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2377     chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2378   }
2379 </book|report>
2380 <*article|beamer>
2381 \tl_clear:N \l__ctex_tmp_tl
2382 </article|beamer>

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

```

定义标题键值选项。

```

2383 \seq_map_inline:Nn \c__ctex_headings_seq
2384   {
2385     \__ctex_initial_heading:n {#1}
2386     \__ctex_def_heading_keys:n {#1}
2387   }
2388 \use:x
2389   {
2390     \group_end:
2391     \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2392   }

2393 <*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。

```

2394 \cs_new_protected_nopar:Npn \CTEX@makeheadinganchor { }
2395 \ctex_at_end_package:nn { hyperref }
2396   { \cs_gset_protected_nopar:Npn \CTEX@makeheadinganchor { \phantomsection } }

(End definition for \CTEX@makeheadinganchor.)

```

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

```

2397 <*book|report>
2398 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2399   {
2400     \CTEX@fixheadingskip
2401     \dim_compare:nNf \tex_pagegoal:D < \c_max_dim
2402       { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2403   }
2404 </book|report>

```

(End definition for \CTEX@fixtopskip.)

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

```
2405 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2406 {
2407   \par
2408   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2409   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2410 }
2411 \skip_new:N \l__ctex_heading_skip
2412 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2413 { \skip_set:Nn \l__ctex_heading_skip }
2414 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip
```

(End definition for \CTEX@fixheadingskip.)

14.8.2.1 part 的标题

2415 <@@=>

```
\part 2416 <*article>
2417 \renewcommand\part{%
2418   \if@noskipsec \leavevmode \fi
2419   \par
2420   % \addvspace{4ex}%
2421   \CTEX@setheadingskip \CTEX@part@before skip
2422   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2423   \addvspace \CTEX@headingskip
2424   \ifodd \CTEX@part@afterindent
2425     \@afterindenttrue
2426   \else
2427     \@afterindentfalse
2428   \fi
2429   \secdef\@part\@spart}
2430 </article>
2431 <*book|report>
2432 \renewcommand\part{%
2433   \if@openright
2434     \cleardoublepage
2435   \else
2436     \clearpage
2437   \fi
2438   % \thispagestyle{plain}%
2439   \thispagestyle{\CTEX@part@pagestyle}%
2440   \if@twocolumn
2441     \onecolumn
2442     \@tempwatrue
2443   \else
2444     \@tempwafalse
2445   \fi
2446   % \null\vfil
2447   \CTEX@setheadingskip \CTEX@part@before skip
2448   \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2449   \vspace*{\CTEX@headingskip}%
2450   \secdef\@part\@spart}
2451 </book|report>
```

(End definition for \part.)

```
\@part 2452 <*article>
2453 \def\@part[#1]#2{%
2454   \ifnum \c@secnumdepth >\m@ne
2455     \ifodd \CTEX@part@numbering
```

```

2456     \refstepcounter{part}%
2457 %     \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2458     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2459     \else
2460         \CTEX@makeheadinganchor
2461         \addcontentsline{toc}{part}{#1}%
2462     \fi
2463 \else
2464     \addcontentsline{toc}{part}{#1}%
2465 \fi
2466 {\interlinepenalty \@M
2467 % \normalfont \parindent \z@ \raggedright
2468 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2469 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2470 %     \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2471     \CTEX@partname \CTEX@part@aftername
2472 \fi \fi
2473 % \huge\bfseries #2%
2474 \CTEX@part@titleformat{#2}%
2475 \markboth{}{}%
2476 \CTEX@part@aftertitle}%
2477 \nobreak
2478 % \vskip 3ex
2479 \CTEX@setheadingskip \CTEX@part@afterskip
2480 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2481 \vskip \CTEX@headingskip
2482 \@afterheading}
2483 </article>
2484 <*book|report>
2485 \def\@part[#1]#2{%
2486     \ifnum \c@secnumdepth >-2\relax
2487         \ifodd \CTEX@part@numbering
2488             \refstepcounter{part}%
2489 %             \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2490             \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2491         \else
2492             \CTEX@makeheadinganchor
2493             \addcontentsline{toc}{part}{#1}%
2494         \fi
2495     \else
2496         \addcontentsline{toc}{part}{#1}%
2497     \fi
2498     \markboth{}{}%
2499     {\interlinepenalty \@M
2500 % \normalfont \centering
2501 \normalfont \CTEX@part@format
2502 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2503 %     \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2504     \CTEX@partname \CTEX@part@aftername
2505 \fi \fi
2506 % \Huge\bfseries #2\par}%
2507 \CTEX@part@titleformat{#2}%
2508 \CTEX@part@aftertitle}%
2509 \@endpart}
2510 </book|report>
(End definition for \@part.)

\@spart 2511 <*article>
2512 \def\@spart#1{%
2513     {\interlinepenalty \@M
2514 % \normalfont \parindent \z@ \raggedright
2515 \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2516 \CTEX@part@format
2517 % \huge \bfseries #1\par}%
2518 \CTEX@part@titleformat{#1}%
2519 \CTEX@part@aftertitle}%
2520 \nobreak

```

```

2521 %      \vskip 3ex
2522      \CTEX@setheadingskip \CTEX@part@afterskip
2523      \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2524      \vskip \CTEX@headingskip
2525      \@afterheading}
2526 </article>
2527 <*book|report>
2528 \def\@spart#1{%
2529     {\interlinepenalty \@M
2530     %      \normalfont \centering
2531     \normalfont \CTEX@part@format
2532     %      \Huge \bfseries #1\par}%
2533     \CTEX@part@titleformat{#1}%
2534     \CTEX@part@aftertitle}%
2535     \@endpart}
2536 </book|report>
(End definition for \@spart.)

\@endpart 2537 <*book|report>
2538 \def\@endpart{%
2539     %      \vfil
2540     \CTEX@setheadingskip \CTEX@part@afterskip
2541     \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2542     \vskip \CTEX@headingskip
2543     \newpage
2544     \if@twoside
2545     \if@openright
2546     \null
2547     \thispagestyle{empty}%
2548     \newpage
2549     \fi
2550     \fi
2551     \if@tempswa
2552     \twocolumn
2553     \fi}
2554 </book|report>
(End definition for \@endpart.)

```

14.8.2.2 chapter 的标题

```

2555 <*book|report>

\chapter 2556 \renewcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
2557 %      \thispagestyle{plain}%
2558      \thispagestyle{\CTEX@chapter@pagestyle}%
2559      \global\@topnum\z@
2560 %      \@afterindentfalse
2561      \ifodd \CTEX@chapter@afterindent
2562      \@afterindenttrue
2563      \else
2564      \@afterindentfalse
2565      \fi
2566      \secdef\@chapter\@schapter}
(End definition for \chapter.)

\@chapter 2567 \def\@chapter[#1]#2{%
2568     \ifnum \c@secnumdepth >\m@ne
2569     <book> \if@mainmatter
2570     \ifodd \CTEX@chapter@numbering
2571     \refstepcounter{chapter}%
2572     %      \typeout{\@chapapp\space\thechapter.}%
2573     \typeout{\CTEXthechapter}%
2574     \addcontentsline{toc}{chapter}
2575     %      {\protect\numberline{\thechapter}#1}%
2576     {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2577     \else

```

```

2578         \CTEX@makeheadinganchor
2579         \addcontentsline{toc}{chapter}{#1}%
2580     \fi
2581     <book>     \else
2582     <book>         \addcontentsline{toc}{chapter}{#1}%
2583     <book>     \fi
2584     \else
2585         \addcontentsline{toc}{chapter}{#1}%
2586     \fi
2587     \chaptermark{#1}%
2588     \addtocontents{lof}{\protect\addvspace{10\p@}}%
2589     \addtocontents{lot}{\protect\addvspace{10\p@}}%
2590     \if@twocolumn
2591         \@topnewpage[\@makechapterhead{#2}]%
2592     \else
2593         \@makechapterhead{#2}%
2594     \afterheading
2595 \fi}

```

(End definition for \@chapter.)

```

\@makechapterhead 2596 \def\@makechapterhead#1{%
2597 % \vspace*{50\p@}%
2598 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2599 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2600 \vspace*{\CTEX@headingskip}%
2601 % {\normalfont \parindent \z@ \raggedright
2602 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2603 \CTEX@chapter@format
2604 \ifnum \c@secnumdepth >\m@ne
2605 <book>     \ifmainmatter
2606             \ifodd \CTEX@chapter@numbering
2607 %         \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2608             \CTEX@chaptername \CTEX@chapter@aftername
2609     \fi
2610 <book>     \fi
2611     \fi
2612     \interlinepenalty\@M
2613 % \Huge \bfseries #1\par\nobreak
2614 \CTEX@chapter@titleformat{#1}%
2615 \CTEX@chapter@aftertitle
2616 \nobreak
2617 % \vskip 40\p@
2618 \CTEX@setheadingskip \CTEX@chapter@afterskip
2619 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2620 \vskip \CTEX@headingskip
2621 }}

```

(End definition for \@makechapterhead.)

```

\@makeschapterhead 2622 \def\@makeschapterhead#1{%
2623 % \vspace*{50\p@}%
2624 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2625 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2626 \vspace*{\CTEX@headingskip}%
2627 % {\normalfont \parindent \z@ \raggedright
2628 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2629 \CTEX@chapter@format
2630 \interlinepenalty\@M
2631 % \Huge \bfseries #1\par\nobreak
2632 \CTEX@chapter@titleformat{#1}
2633 \CTEX@chapter@aftertitle
2634 \nobreak
2635 % \vskip 40\p@
2636 \CTEX@setheadingskip \CTEX@chapter@afterskip
2637 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2638 \vskip \CTEX@headingskip
2639 }}

```

(End definition for \makeschapterhead.)

2640 </book|report>

14.8.2.3 section 类的标题

```
\@startsection 2641 \def\@startsection#1#2#3#4#5#6{%
2642   \if@noskipsec \leavevmode \fi
2643   \par
2644   % \@tempskipa #4\relax
2645   % \@afterindenttrue
2646   % \ifdim \@tempskipa <\z@
2647   %   \@tempskipa -\@tempskipa \@afterindentfalse
2648   % \fi
2649   \CTEX@update@sectionformat@n{#1}%
2650   \ifodd \CTEX@afterindent
2651     \@afterindenttrue
2652   \else
2653     \@afterindentfalse
2654   \fi
2655   \if@nobreak
2656     \everypar{}%
2657   \else
2658     %   \addpenalty\@secpenalty\addvspace\@tempskipa
2659     \addpenalty\@secpenalty
2660     \CTEX@setheadingskip{#4}%
2661     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2662     \addvspace \CTEX@headingskip
2663   \fi
2664   \@ifstar
2665     {\@ssect{#3}{#4}{#5}{#6}}%
2666     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
(End definition for \@startsection.)

\@secntformat 2667 \def\@secntformat#1{%
2668   % \csname the#1\endcsname\quad}%
2669   \csname CTEX@#1name\endcsname \csname CTEX@#1aftername\endcsname}
(End definition for \@secntformat.)

\@sect 2670 \def\@sect#1#2#3#4#5#6[#7]#8{%
2671   \ifnum #2>\c@secnumdepth
2672     \let\@svsec\@empty
2673   \else
2674     \ifodd \csname CTEX@#1numbering\endcsname
2675       \refstepcounter{#1}%
2676       \protected@edef\@svsec{\@secntformat{#1}\relax}%
2677     \else
2678       \CTEX@makeheadinganchor
2679       \let\@svsec\@empty
2680     \fi
2681   \fi
2682   % \@tempskipa #5\relax
2683   % \ifdim \@tempskipa>\z@
2684   \unless \ifodd \CTEX@runin
2685     \begingroup
2686       #6{%
2687         \@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2688         % \interlinepenalty \@M #8\@par}%
2689         \interlinepenalty \@M
2690         \csname CTEX@#1titleformat\endcsname{#8}%
2691         \csname CTEX@#1aftertitle\endcsname}%
2692     \endgroup
2693     \csname #1mark\endcsname{#7}%
2694     \addcontentsline{toc}{#1}{%
2695       \ifnum #2>\c@secnumdepth \else
2696         \ifodd \csname CTEX@#1numbering\endcsname
```

```

2697 %          \protect\numberline{\csname the#1\endcsname}%
2698          \protect\numberline{\csname CTEXthe#1\endcsname}%
2699          \fi
2700          \fi
2701          #7}}%
2702      \else
2703          \def\@svsechd{%
2704              #6{\hskip\glueexpr #3\relax
2705              \@svsec #8}}%
2706          \@svsec
2707          \csname CTEX@#1@titleformat\endcsname{#8}%
2708          \csname CTEX@#1@aftertitle\endcsname}%
2709          \csname #1mark\endcsname{#7}%
2710          \addcontentsline{toc}{#1}{%
2711              \ifnum #2>\c@secnumdepth \else
2712                  \ifodd \csname CTEX@#1@numbering\endcsname
2713                      \protect\numberline{\csname the#1\endcsname}%
2714                      \protect\numberline{\csname CTEXthe#1\endcsname}%
2715                  \fi
2716              \fi
2717              #7}}}%
2718      \fi
2719      \@xsect{#5}}
(End definition for \@sect.)

\@ssect 2720 \def\@ssect#1#2#3#4#5{%
2721 % \@tempskipa #3\relax
2722 % \ifdim \@tempskipa>\z@
2723 \unless \ifodd \CTEX@runin
2724     \begingroup
2725         #4{%
2726             \@hangfrom{\hskip\glueexpr #1\relax}%
2727             % \interlinepenalty \@M #5\@par}%
2728             \interlinepenalty \@M
2729             \CTEX@titleformatn{#5}%
2730             \CTEX@aftertitle}%
2731     \endgroup
2732 \else
2733 % \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2734 \def\@svsechd{#4{\hskip\glueexpr #1\relax
2735                 \CTEX@titleformatn{#5}\CTEX@aftertitle}}%
2736 \fi
2737 \@xsect{#3}}
(End definition for \@ssect.)

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

```

```

2759 %      \hskip -\@tempskipa
2760      \hskip\glueexpr #1\relax
2761      \else
2762      \clubpenalty \@clubpenalty
2763      \everypar{}\%
2764      \fi}%
2765 \fi
2766 \ignorespaces}
(End definition for \@xsect.)

2767 <@@=ctex>

```

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

```

2768 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2769 {
2770   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2771   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2772   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2773   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2774   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2775 }
2776 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2777 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2778 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2779 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2780 \cs_new_eq:NN \CTEX@runin \c_false_bool
(End definition for \CTEX@update@sectionformat@n.)

2781 \int_zero:N \l__ctex_tmp_int
2782 \seq_map_inline:Nn \c__ctex_section_headings_seq
2783 {
2784   \int_incr:N \l__ctex_tmp_int
2785   \cs_gset_protected_nopar:cpx {#1}
2786   {
2787     \exp_not:N \@startsection {#1}
2788     { \int_use:N \l__ctex_tmp_int }
2789     { \exp_not:c { CTEX@#1@indent } }
2790     { \exp_not:c { CTEX@#1@beforeskip } }
2791     { \exp_not:c { CTEX@#1@afterskip } }
2792     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2793   }
2794 }

```

14.8.2.4 附录标题

```

appendix/name 2795 \keys_define:nn { ctex }
appendix/number 2796 {
appendix/numbering 2797   appendix .meta:nn = { ctex / appendix } {#1} ,
2798   appendix / name .code:n =
2799   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2800   appendix / number .tl_set:N = \CTEX@appendix@number ,
2801   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2802   appendix / numbering .initial:n = true
2803 }
2804 \tl_new:N \CTEX@preappendix
2805 \tl_new:N \CTEX@postappendix
(End definition for appendix/name, appendix/number, and appendix/numbering.)

```

```

\appendix 2806 \cs_new_eq:NN \CTEX@save@appendix \appendix
2807 \cs_gset_protected_nopar:Npn \appendix
2808 {
2809   \CTEX@save@appendix
2810 <*article>
2811   \gdef \CTEX@presection { \CTEX@preappendix }

```



```

2812 \gdef \CTEX@thesection { \CTEX@appendix@number }
2813 \gdef \CTEX@postsection { \CTEX@postappendix }
2814 \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2815 </article>
2816 <*book|report>
2817 \gdef \CTEX@prechapter { \CTEX@preappendix }
2818 \gdef \CTEX@thechapter { \CTEX@appendix@number }
2819 \gdef \CTEX@postchapter { \CTEX@postappendix }
2820 \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2821 </book|report>
2822 }
(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 (`<before skip>`) 和第五个 (`<after skip>`) 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 `\ttlhf@<section>` 的宏中备用, 它的内容是

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

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

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

其中 `<afterindent>` 为 1 或 0, 分别对应是否保留段首缩进。我们需要根据 `CTEX` 的 `runin` 和 `afterindent` 选项调整 `\ttlh@<shape>` 和 `<afterindent>`。注意, 由 `\ttl@extract` 得的 `<before>` 和 `<after>` 的值总是非负的, 而 `CTEX` 的 `before skip` 和 `after skip` 是可以取负值的, 但我们不打算调整它们了。如果使用了 `titlesec` 的 `indentafter` 等选项, 也不需要调整 `\ttls@<section>`。

```

2823 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
2824 {
2825   \ifpackagewith { titlesec } { explicit }
2826   {
2827     \cs_set_eq:NN \__ctex_titlesec_format:Nn
2828     \__ctex_titlesec_format_explicit:Nn
2829   }
2830   { }
2831   \clist_map_inline:nn
2832   { indentafter , noindentafter , indentfirst , nonindentfirst }
2833   {
2834     \ifpackagewith { titlesec } { ##1 }
2835     {
2836       \clist_map_break:n
2837       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
2838     }
2839     { }
2840   }
2841   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
2842 }
2843 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
2844 {
2845   \__ctex_titlesec_format:n {#1}
2846   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
2847 }
2848 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1

```

```

2849 {
2850   \cs_if_free:cF { ttlf@#1 }
2851   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
2852 }
2853 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
2854 {
2855   \tl_set:Nx #1
2856   {
2857     \bool_if:cTF { CTEX@#2@runin }
2858     { \exp_not:N \ttlh@runin }
2859     { \exp_not:N \ttlh@hang }
2860     \tl_tail:N #1
2861   }
2862 }
2863 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
2864 {
2865   \cs_set_nopar:Npx #1 ##1
2866   {
2867     \bool_if:cTF { CTEX@#2@runin }
2868     { \exp_not:N \ttlh@runin }
2869     { \exp_not:N \ttlh@hang }
2870     \exp_args:No \tl_tail:n { #1 { } }
2871   }
2872 }
2873 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
2874 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
2875 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
2876 {
2877   \exp_not:n { {#1} {#2} {#3} {#4} }
2878   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
2879 }

(End definition for \ctex_titlesec_hook:.)

2880 \@ifpackageloaded { titlesec }
2881 { }
2882 {
2883   \ctex_at_end_package:nn { titlesec }
2884   {
2885     \@ifpackagewith { titlesec } { loadonly }
2886     { }
2887     { \ctex_titlesec_hook: }
2888   }
2889 }

```

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。

```

2890 \group_begin:
2891 \char_set_catcode_other:N \#
2892 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
2893 {
2894   \ctex_patch_cmd:Nnn \ttl@settopmark
2895   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
2896   {
2897     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
2898     \CTEX@titlepslabel@set {#1}
2899   }
2900   \ctex_patch_cmd:Nnn \ttl@setsubmark

```

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

```

2901     { \protect \@namedef { the#1 } { } }
2902     {
2903         \protect \@namedef { the#1 } { }
2904         \CTEX@titlepslabel@clear {#1}
2905     }
2906     \ctex_patch_cmd:Nnn \ttl@setsubmark
2907     { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
2908     {
2909         \protect \@namedef { the#2 } { \@nameuse { the#2 } }
2910         \CTEX@titlepslabel@set {#2}
2911     }
2912 }
2913 \group_end:

```

(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。

```

2914 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
2915 {
2916     \cs_if_free:cF { CTEXthe#1 }
2917     { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
2918 }
2919 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
2920 {
2921     \cs_if_free:cF { CTEXthe#1 }
2922     { \protect \@namedef { CTEXthe#1 } { } }
2923 }

```

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

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

```

2924 \ctex_at_end_package:nn { titlesec }
2925 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
2926 \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 2927 \cs_new_protected:Npn \CTEX@toc@width@n #1
2928 {
2929     \hbox_set:Nn \l__ctex_tmp_box {#1}
2930     \dim_set:Nn \@tempdima
2931     {
2932         \dim_max:nn { \@tempdima }
2933         { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2934     }
2935 }
2936 \group_begin:
2937 \char_set_catcode_other:N \#
2938 \use:n
2939 {
2940     \group_end:
2941     \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
2942     { \CTEX@toc@width@n {#1} }
2943     { }
2944     { \ctex_patch_failure:N \numberline }
2945     \ifpackageloaded { tocloft }
2946     { }

```

```

2947     {
2948         \ctex_at_end_package:nn { tocloft }
2949         {
2950             \ctex_preto_cmd:NnnTF \numberline
2951             { \char_set_catcode_letter:n { 64 } }
2952             { \CTEX@toc@width@n {#1} }
2953             { }
2954             { \ctex_patch_failure:N \numberline }
2955         }
2956     }
2957 }
(End definition for \numberline.)

```

14.8.4 页眉信息的修改

```

\ps@headings 2958 <*article>
2959 \ctex_patch_cmd:Nnn \ps@headings
2960 { \thesection \quad }
2961 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2962 \if@twoside
2963 \ctex_patch_cmd:Nnn \ps@headings
2964 { \thesubsection \quad }
2965 { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2966 \fi:
2967 </article>
2968 <*book|report>
2969 \ctex_patch_cmd:Nnn \ps@headings
2970 { \@chapapp\ \thechapter.\~\ }
2971 { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2972 \if@twoside
2973 \ctex_patch_cmd:Nnn \ps@headings
2974 { \thesection.\~\ }
2975 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2976 \fi:
2977 </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。

```

2978 \ctex_at_end_package:nn { fancyhdr }
2979 {
2980 <*article>
2981 \ctex_patch_cmd:Nnn \ps@fancy
2982 { \thesection \hspace{1em} \relax }
2983 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2984 \ctex_patch_cmd:Nnn \ps@fancy
2985 { \thesubsection \hspace{1em} \relax }
2986 { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2987 </article>
2988 <*book|report>
2989 \ctex_patch_cmd:Nnn \ps@fancy
2990 { \@chapapp\ \thechapter.\~\ }
2991 <book> { \if@mainmatter \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi \fi }
2992 <report> { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2993 \ctex_patch_cmd:Nnn \ps@fancy
2994 { \thesection.\~\ }
2995 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2996 </book|report>
2997 }
(End definition for \ps@fancy.)
2998 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

2999 <*beamer>

3000 \ExplSyntaxOff

对应 \partpage。

```

3001 \defbeamer*template*{part page}{CTEX}[1][]{
3002   \begingroup
3003   % \centering
3004   % {\usebeamerfont{part name}\usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3005   % \vskip1em\par
3006   \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3007   \parindent \dimexpr \CTEX@part@indent \relax
3008   \CTEX@part@format
3009   \ifodd \CTEX@part@numbering
3010     \CTEX@partname \CTEX@part@aftername
3011   \fi
3012   \begin{beamer*colorbox}[sep=16pt,center,#1]{part title}
3013   % \usebeamerfont{part title}\insertpart\par
3014   \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3015   \end{beamer*colorbox}
3016   \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3017 \endgroup
3018 }
```

对应 \sectionpage。

```

3019 \defbeamer*template*{section page}{CTEX}[1][]{
3020   \begingroup
3021   % \centering
3022   % {\usebeamerfont{section name}\usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}
3023   % \vskip1em\par
3024   \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3025   \parindent \dimexpr \CTEX@section@indent \relax
3026   \CTEX@section@format
3027   \ifodd \CTEX@section@numbering
3028     \CTEX@sectionname \CTEX@section@aftername
3029   \fi
3030   \begin{beamer*colorbox}[sep=12pt,center,#1]{part title}
3031   % \usebeamerfont{section title}\insertsection\par
3032   \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3033   \end{beamer*colorbox}
3034   \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3035 \endgroup
3036 }
```

对应 \subsectionpage。

```

3037 \defbeamer*template*{subsection page}{CTEX}[1][]{
3038   \begingroup
3039   % \centering
3040   % {\usebeamerfont{subsection name}\usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
3041   % \vskip1em\par
3042   \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3043   \parindent \dimexpr \CTEX@subsection@indent \relax
3044   \CTEX@subsection@format
3045   \ifodd \CTEX@subsection@numbering
3046     \CTEX@subsectionname \CTEX@subsection@aftername
3047   \fi
3048   \begin{beamer*colorbox}[sep=8pt,center,#1]{part title}
3049   % \usebeamerfont{subsection title}\insertsubsection\par
3050   \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3051   \end{beamer*colorbox}
3052   \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3053 \endgroup
3054 }
```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3055 \defbeamertemplatealias{part page}{default}{CTEX}
3056 \defbeamertemplatealias{section page}{default}{CTEX}
3057 \defbeamertemplatealias{subsection page}{default}{CTEX}
3058 \ExplSyntaxOn
3059 </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。

```

3060 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
3061 {
3062   \protected@edef \@currentlabel
3063   {
3064     \cs_if_exist:cTF { CTEX@the#1 }
3065     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3066     { \exp_not:o { \@currentlabel } }
3067   }
3068 }

```

(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。

```

3069 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3070 {
3071   \seq_map_inline:Nn \c__ctex_headings_seq
3072   { \ctex_fix_varioref_label:n { ##1 } }
3073 }

```

(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。

```

3074 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3075 {
3076   \tl_if_empty:cT { p@#1 }
3077   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3078 }

```

(End definition for `__ctex_fix_varioref_label:n`.)

如果 `varioref` 已经被载入, 则使用它来设置。

```

3079 \@ifpackageloaded { varioref }
3080 { \ctex_varioref_hook: }
3081 {
3082   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3083   \RenewDocumentCommand \refstepcounter { m }
3084   {
3085     \CTEX@save@refstepcounter {#1}
3086     \CTEX@setcurrentlabel@n {#1}
3087   }
3088   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3089 }

```

14.8.7 载入 $\langle scheme \rangle$ 文件

```
3090 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3091 </class|heading>
```

14.8.8 标题格式的 scheme 定义

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

```
3092 <*scheme&(article|book|report|beamer)>

3093 \keys_set:nn { ctex / part }
3094 {
3095     aftertitle = \par ,
3096 <*plain>
3097     name       = \partname \space ,
3098 <*article|book|report>
3099     number     = \thepart ,
3100 </article|book|report>
3101 <*beamer>
3102     number     = \insertromanpartnumber ,
3103 </beamer>
3104 </plain>
3105 <*chinese>
3106     number     = \chinese { part } ,
3107 </chinese>
3108 <*article>
3109     beforeskip = 4ex ,
3110     afterskip  = 3ex ,
3111 <*plain>
3112     format     = \raggedright ,
3113     nameformat = \Large \bfseries ,
3114     aftername  = \par \nobreak ,
3115     titleformat = \huge \bfseries ,
3116     afterindent = false
3117 </plain>
3118 <*chinese>
3119     format     = \Large \bfseries \centering ,
3120     aftername  = \quad ,
3121     afterindent = true
3122 </chinese>
3123 </article>
3124 <*book|report>
3125     aftername  = \par \vskip 20 \p@ ,
3126     beforeskip = 0pt \@plus 1fil ,
3127     afterskip  = 0pt \@plus 1fil ,
3128     pagestyle  = plain ,
3129 <*plain>
3130     format     = \centering ,
3131     nameformat = \huge \bfseries ,
3132     titleformat = \Huge \bfseries
3133 </plain>
3134 <*chinese>
3135     format     = \huge \bfseries \centering
3136 </chinese>
3137 </book|report>
3138 <*beamer>
3139     format     = \centering ,
3140     nameformat = \usebeamerfont { part ~ name }
3141                \usebeamercolor [fg] { part ~ name } ,
3142     aftername  = \vskip 1em \par ,
3143     titleformat = \usebeamerfont { part ~ title }
3144 </beamer>
3145 }

3146 <*book|report>
```

```

3147 \keys_set:nn { ctex / chapter }
3148 {
3149     pagestyle     = plain ,
3150     aftertitle    = \par ,
3151     beforeskip    = 50 \p@ ,
3152     afterskip     = 40 \p@ ,
3153     <*plain>
3154     name          = \chaptername \space ,
3155     number        = \thechapter ,
3156     format        = \raggedright ,
3157     nameformat    = \huge \bfseries ,
3158     aftername     = \par \nobreak \vskip 20 \p@ ,
3159     titleformat   = \Huge \bfseries ,
3160     afterindent   = false
3161     </plain>
3162     <*chinese>
3163     number        = \chinese { chapter } ,
3164     format        = \huge \bfseries \centering ,
3165     aftername     = \quad ,
3166     afterindent   = true
3167     </chinese>
3168 }
3169 </book|report>

3170 <@@=>

3171 \keys_set:nn { ctex / section }
3172 {
3173     <*article|book|report>
3174     number        = \thesection ,
3175     aftername     = \quad ,
3176     aftertitle    = \@@par ,
3177     beforeskip    = 3.5ex \@plus 1ex \@minus .2ex ,
3178     afterskip     = 2.3ex \@plus .2ex ,
3179     runin         = false ,
3180     <*plain>
3181     format        = \Large \bfseries ,
3182     afterindent   = false
3183     </plain>
3184     <*chinese>
3185     format        = \Large \bfseries \centering ,
3186     afterindent   = true
3187     </chinese>
3188     </article|book|report>
3189     <*beamer>
3190     <*plain>
3191     name          = \sectionname \space ,
3192     </plain>
3193     format        = \centering ,
3194     number        = \insertsectionnumber ,
3195     nameformat    = \usebeamerfont { section ~ name }
3196                   \usebeamercolor [fg] { section ~ name } ,
3197     aftername     = \vskip 1em \par ,
3198     titleformat   = \usebeamerfont { section ~ title } ,
3199     aftertitle    = \par
3200     </beamer>
3201 }

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

```



```

3212 <*plain>
3213     afterindent = false
3214 </plain>
3215 <*chinese>
3216     afterindent = true
3217 </chinese>
3218 </article|book|report>
3219 <*beamer>
3220 <*plain>
3221     name          = \subsectionname \space ,
3222     number        = \insertsubsectionnumber ,
3223 </plain>
3224 <*chinese>
3225     number        = \arabic { section } . \arabic { subsection } ,
3226 </chinese>
3227     format        = \centering ,
3228     nameformat    = \usebeamerfont { subsection ~ name }
3229                  \usebeamercolor [fg] { subsection ~ name } ,
3230     aftername     = \vskip 1em \par ,
3231     titleformat   = \usebeamerfont { subsection ~ title } ,
3232     aftertitle    = \par
3233 </beamer>
3234 }

3235 <*article|book|report>

3236 \keys_set:nn { ctex / subsubsection }
3237 {
3238     number        = \thesubsubsection ,
3239     format        = \normalsize \bfseries ,
3240     aftername     = \quad ,
3241     aftertitle    = \@@par ,
3242     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3243     afterskip     = 1.5ex \@plus .2ex ,
3244     runin         = false ,
3245 <*plain>
3246     afterindent   = false
3247 </plain>
3248 <*chinese>
3249     afterindent   = true
3250 </chinese>
3251 }

3252 \keys_set:nn { ctex / paragraph }
3253 {
3254     number        = \theparagraph ,
3255     format        = \normalsize \bfseries ,
3256     aftername     = \quad ,
3257     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3258 <*plain>
3259     afterindent   = false
3260 </plain>
3261 <*chinese>
3262     afterindent   = true
3263 </chinese>
3264 }

3265 \keys_set:nn { ctex / subparagraph }
3266 {
3267     number        = \thesubparagraph ,
3268     format        = \normalsize \bfseries ,
3269     aftername     = \quad ,
3270     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3271 <*plain>
3272     afterindent   = false
3273 </plain>
3274 <*chinese>
3275     afterindent   = true

```

```
3276 </chinese>
3277 }
```

处理 sub3section 与 sub4section 的格式。

```
3278 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3279 {
3280   \keys_set:nn { ctex / paragraph }
3281   {
3282     aftertitle = \@@par ,
3283     afterskip  = 1ex \@plus .2ex ,
3284     runin      = false
3285   }
3286 }
3287 {
3288   \keys_set:nn { ctex / paragraph }
3289   {
3290     afterskip  = 1em ,
3291     runin      = true
3292   }
3293 }
3294 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3295 {
3296   \keys_set:nn { ctex / subparagraph }
3297   {
3298     aftertitle = \@@par ,
3299     afterskip  = 1ex \@plus .2ex ,
3300     runin      = false
3301   }
3302 }
3303 {
3304   \keys_set:nn { ctex / subparagraph }
3305   {
3306     afterskip  = 1em ,
3307     runin      = true
3308   }
3309 }
3310 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3311 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3312 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3313 <@@=ctex>
```

处理附录的格式。

```
3314 \keys_set:nn { ctex / appendix }
3315 <*article>
3316 { number      = \@Alph \c@section }
3317 </article>
3318 <*book|report>
3319 {
3320   name         = \appendixname \space ,
3321   number       = \@Alph \c@chapter
3322 }
3323 </book|report>
3324 </article|book|report>
3325 </scheme&(article|book|report|beamer)>
```

14.8.9 ctex.sty 的 heading 选项

```
3326 <*ctex|ctexheading>
```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```
3327 \clist_map_inline:nn { article , book , report , beamer }
3328 {
```

```

3329 \ifclassloaded {#1}
3330 { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3331 { }
3332 }

```

(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。

```

3333 \msg_new:nnn { ctex } { not-standard-class }
3334 {
3335   None~of~the~standard~document~classes~was~loaded.\\
3336   Heading~`#1'~is~selected.\\
3337   ctex~may~not~work~as~expected.
3338 }
3339 <ctex>\bool_if:NTF \l__ctex_heading_bool
3340 <ctexheading>\use:n
3341 {
3342   \tl_if_exist:NTF \c__ctex_std_class_tl
3343   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3344   {
3345     \cs_if_exist:NTF \chapter
3346     {
3347       \cs_if_exist:NF \if@mainmatter
3348       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3349       \tl_const:Nn \c__ctex_class_tl { book }
3350     }
3351     { \tl_const:Nn \c__ctex_class_tl { article } }
3352     \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3353   }
3354   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3355 }
3356 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3357 </ctex|ctexheading>

```

14.8.10 标题配置文件

```

3358 <*name>
3359 \keys_set_known:nn { ctex }
3360 {
3361   contentsname = 目录 ,
3362   listfigurename = 插图 ,
3363   listtablename = 表格 ,
3364   figurename = 图 ,
3365   tablename = 表 ,
3366   abstractname = 摘要 ,
3367   indexname = 索引 ,
3368   bibname = 参考文献 ,
3369   appendixname = 附录 ,
3370   proofname = 证明 ,
3371   algorithmname = 算法 ,
3372   refname = 参考文献 ,
3373   continuation = (续) ,
3374   part / name = { 第 , 部分 } ,
3375   chapter / name = { 第 , 章 }
3376 }
3377 </name>

```

14.9 chinese 方案的其他设置

```

3378 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```
3379 <article|book|report>\pagestyle { headings }
```

日期格式。

```
3380 \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。

```
3381 \ctex_if_autoindent_touched:F
```

```
3382 { \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。

```
3383 <!*generic>
```

```
3384 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
```

```
3385 <*beamer>
```

```
3386 {
```

```
3387   \uselanguage { ChineseGBK }
```

```
3388   \languagealias { chinese } { ChineseGBK }
```

```
3389   \ctex_file_input:n { ctex-name-gbk.cfg }
```

```
3390 }
```

```
3391 {
```

```
3392   \uselanguage { ChineseUTF8 }
```

```
3393   \languagealias { chinese } { ChineseUTF8 }
```

```
3394   \ctex_file_input:n { ctex-name-utf8.cfg }
```

```
3395 }
```

让 `translator` 包优先查找中文翻译。

```
3396 \clist_put_left:Nn \trans@languagepath { chinese }
```

```
3397 </beamer>
```

```
3398 <!*beamer>
```

```
3399 { \ctex_file_input:n { ctex-name-gbk.cfg } }
```

```
3400 { \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。

```
3401 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
```

```
3402 { \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。

```
3403 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
```

```
3404 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
```

```
3405 </!beamer>
```

```
3406 </!generic>
```

不使用标题定义时的通用设置。

```
3407 <!*generic>
```

```
3408 \tl_set:Nn \l__ctex_tmp_tl { beamer }
```

```
3409 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
```

```
3410 {
```

```
3411   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
```

```
3412   {
```

```
3413     \uselanguage { ChineseGBK }
```

```
3414     \languagealias { chinese } { ChineseGBK }
```

```
3415     \ctex_file_input:n { ctex-name-gbk.cfg }
```

```
3416   }
```

```
3417   {
```

```
3418     \uselanguage { ChineseUTF8 }
```

```
3419     \languagealias { chinese } { ChineseUTF8 }
```

```
3420     \ctex_file_input:n { ctex-name-utf8.cfg }
```

```
3421   }
```

```
3422   \clist_put_left:Nn \trans@languagepath { chinese }
```

```
3423 }
```

```
3424 {
```

```
3425   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
```

```
3426   { \ctex_file_input:n { ctex-name-gbk.cfg } }
```

```

3427 { \ctex_file_input:n { ctex-name-utf8.cfg } }
3428 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3429 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3430 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3431 { \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。

```

3432 \tl_if_exist:NT \c__ctex_std_class_tl
3433 { \RequirePackage { indentfirst } }
3434 }
3435 </generic>
3436 </scheme&chinese>

```

14.10 中文字号

```

3437 <*class|ctex|ctexsize>

```

```

\zihao 3438 \NewDocumentCommand \zihao { m }
3439 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

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

```

\ctex_zihao:n 3440 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3441 {
3442   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3443   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3444   { \msg_error:nnn { ctex } { fontsize } {#1} }
3445 }
3446 \msg_new:nnnn { ctex } { fontsize }
3447 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3448 {
3449   The~old~font~size~is~used~if~you~continue.\\
3450   The~available~font~sizes~are~listed~as~follow.\\
3451   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3452 }

```

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

14.10.1 定义中文字号

```

\c__ctex_font_size_prop
\__ctex_save_font_size:nn

```

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。

```

3453 \prop_new:N \c__ctex_font_size_prop
3454 \seq_new:N \c__ctex_font_size_seq
3455 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3456 {
3457   \use:x
3458   {
3459     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3460     {
3461       { \dim_to_decimal:n {#2} }
3462       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3463     }
3464   }
3465   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3466 }
3467 \clist_map_inline:nn
3468 {
3469   { 8 } { 5 bp } ,
3470   { 7 } { 5.5 bp } ,
3471   { -6 } { 6.5 bp } ,
3472   { 6 } { 7.5 bp } ,

```

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

```

3473 { -5 } { 9 bp } ,
3474 { 5 } { 10.5 bp } ,
3475 { -4 } { 12 bp } ,
3476 { 4 } { 14 bp } ,
3477 { -3 } { 15 bp } ,
3478 { 3 } { 16 bp } ,
3479 { -2 } { 18 bp } ,
3480 { 2 } { 22 bp } ,
3481 { -1 } { 24 bp } ,
3482 { 1 } { 26 bp } ,
3483 { -0 } { 36 bp } ,
3484 { 0 } { 42 bp }
3485 }
3486 { \__ctex_save_font_size:nn #1 }

(End definition for \c__ctex_font_size_prop and \__ctex_save_font_size:nn.)

```

```

\ctex_declare_math_sizes:nmmn 3487 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nmmn #1#2#3#4
3488 {
3489   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3490   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3491 }

(End definition for \ctex_declare_math_sizes:nmmn.)

```

```

\__ctex_get_font_sizes:Nn 3492 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3493 {
3494   \tl_clear:N #1
3495   \tl_map_inline:nn {#2}
3496   {
3497     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3498     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3499     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3500   }
3501 }

(End definition for \__ctex_get_font_sizes:Nn.)

```

```

3502 \clist_map_inline:nn
3503 {
3504   { 8 } { 8 } { 5pt } { 5pt } ,
3505   { 7 } { 7 } { 5pt } { 5pt } ,
3506   { -6 } { -6 } { 5pt } { 5pt } ,
3507   { 6 } { 6 } { 5pt } { 5pt } ,
3508   { -5 } { -5 } { 6pt } { 5pt } ,
3509   { 5 } { 5 } { 7pt } { 5pt } ,
3510   { -4 } { -4 } { 8pt } { 6pt } ,
3511   { 4 } { 4 } { 5 } { 6 } ,
3512   { -3 } { -3 } { -4 } { -5 } ,
3513   { 3 } { 3 } { 4 } { 5 } ,
3514   { -2 } { -2 } { -3 } { -4 } ,
3515   { 2 } { 2 } { 3 } { 4 } ,
3516   { -1 } { -1 } { -2 } { -3 } ,
3517   { 1 } { 1 } { 2 } { 3 } ,
3518   { -0 } { -0 } { -1 } { -2 } ,
3519   { 0 } { 0 } { 1 } { 2 }
3520 }
3521 { \ctex_declare_math_sizes:nmmn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3522 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3523 {
3524   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3525   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3526   { \msg_error:nnn { ctex } { fontsize } {#2} }
3527 }
3528 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4

```

```

3529 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }
(End definition for \ctex_set_font_size:Nnn.)

3530 \if_case:w \g__ctex_font_size_flag
3531 \ctex_file_input:n { ctex-c5size.clo }
3532 \or:
3533 \ctex_file_input:n { ctex-cs4size.clo }
3534 \fi:

3535 </class|ctex|ctexsize>

3536 <*c5size>
3537 \ctex_set_font_size:Nnn \normalsize { 5 }
3538 {
3539 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3540 \abovedisplayshortskip \z@ \@plus3\p@
3541 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3542 \belowdisplayskip \abovedisplayskip
3543 \let\@listi\@listI
3544 }
3545 \ctex_set_font_size:Nnn \small { -5 }
3546 {
3547 \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3548 \abovedisplayshortskip \z@ \@plus2\p@
3549 \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3550 \def\@listi{\leftmargin\leftmarginI
3551 \topsep 4\p@ \@plus2\p@ \@minus2\p@
3552 \parsep 2\p@ \@plus\p@ \@minus\p@
3553 \itemsep \parsep}
3554 \belowdisplayskip \abovedisplayskip
3555 }
3556 \ctex_set_font_size:Nnn \footnotesize { 6 }
3557 {
3558 \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3559 \abovedisplayshortskip \z@ \@plus\p@
3560 \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3561 \def\@listi{\leftmargin\leftmarginI
3562 \topsep 3\p@ \@plus\p@ \@minus\p@
3563 \parsep 2\p@ \@plus\p@ \@minus\p@
3564 \itemsep \parsep}
3565 \belowdisplayskip \abovedisplayskip
3566 }
3567 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3568 \ctex_set_font_size:Nnn \tiny { 7 } { }
3569 \ctex_set_font_size:Nnn \large { -4 } { }
3570 \ctex_set_font_size:Nnn \Large { -3 } { }
3571 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3572 \ctex_set_font_size:Nnn \huge { 2 } { }
3573 \ctex_set_font_size:Nnn \Huge { 1 } { }
3574 </c5size>
3575 <*cs4size>
3576 \ctex_set_font_size:Nnn \normalsize { -4 }
3577 {
3578 \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3579 \abovedisplayshortskip \z@ \@plus3\p@
3580 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3581 \belowdisplayskip \abovedisplayskip
3582 \let\@listi\@listI
3583 }
3584 \ctex_set_font_size:Nnn \small { 5 }
3585 {
3586 \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3587 \abovedisplayshortskip \z@ \@plus3\p@
3588 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3589 \def\@listi{\leftmargin\leftmarginI
3590 \topsep 9\p@ \@plus3\p@ \@minus5\p@
3591 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3592 \itemsep \parsep}

```

```

3593 \belowdisplayskip \abovedisplayskip
3594 }
3595 \ctex_set_font_size:Nnn \footnotesize { -5 }
3596 {
3597 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3598 \abovedisplayshortskip \z@ \@plus3\p@
3599 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3600 \def\@listi{\leftmargin\leftmargin
3601 \topsep 6\p@ \@plus2\p@ \@minus2\p@
3602 \parsep 3\p@ \@plus2\p@ \@minus\p@
3603 \itemsep \parsep}
3604 \belowdisplayskip \abovedisplayskip
3605 }
3606 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3607 \ctex_set_font_size:Nnn \tiny { -6 } { }
3608 \ctex_set_font_size:Nnn \large { -3 } { }
3609 \ctex_set_font_size:Nnn \Large { -2 } { }
3610 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3611 \ctex_set_font_size:Nnn \huge { -1 } { }
3612 \ctex_set_font_size:Nnn \Huge { 1 } { }
3613 </cs4size>
3614 <ctexsize>\normalsize
3615 <*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`。

```

3616 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3617 {
3618 \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。

```

3619 \group_begin: \footnotesize \exp_args:NNNo \group_end:
3620 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3621 }

```

(End definition for `\footnotesep`.)

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

```

3622 \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。

```

3623 \NewDocumentCommand \CTeX { }
3624 { \ifincsname CTeX \else: C \TeX \fi: }
3625 \ctex_at_end_package:nn { hyperref }
3626 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

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

`captiondelimiter` 过时选项。

```

3627 \keys_define:nn { ctex }
3628 {
3629   captiondelimiter .code:n =
3630   {
3631     \msg_warning:nnn { ctex } { deprecated-option }
3632     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3633   }
3634 }

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

3635 </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`

```

3636 <*scheme&chinese&(article|book|report)>
3637 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3638 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3639 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3640 </scheme&chinese&(article|book|report)>

(End definition for \verse and \quotation.)

3641 <*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。

```

3642 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3643 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3644 \msg_new:nnn { ctex } { fontset-unavailable }
3645 { 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。

```

3646 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3647 {
3648   \tl_if_empty:NTF \g__ctex_fontset_tl
3649   {
3650     \ctex_detected_platform:
3651     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3652   }
3653   {
3654     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3655     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3656     {
3657       \use:x
3658       {
3659         \ctex_detected_platform:
3660         \msg_error:nxxx { ctex } { fontset-not-found }
3661         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3662       }
3663       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3664     }
3665   }
3666 }
3667 \@onlypreamble \ctex_load_fontset:
3668 \msg_new:nnnn { ctex } { fontset-not-found }
3669 {

```

```

3670 CTeX~fontset~`#1'~could~not~be~found.\\
3671 Fontset~`#2'~will~be~used~instead.
3672 }
3673 { 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。

```

3674 \keys_define:nn { ctex }
3675 {
3676   fontset .code:n =
3677   {
3678     \ctex_if_preamble:TF
3679     {
3680       \str_if_eq_x:nnTF {#1} { none }
3681       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3682       {
3683         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3684         {
3685           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3686           \ctex_load_fontset:
3687         }
3688         {
3689           \msg_error:nxxx { ctex } { fontset-loaded }
3690           { \g__ctex_fontset_tl } {#1}
3691         }
3692       }
3693     }
3694     { \msg_error:nn { ctex } { fontset-only-preamble } }
3695   }
3696 }
3697 \msg_new:nnnn { ctex } { fontset-loaded }
3698 {
3699   CTeX~fontset~`#1'~has~been~loaded.
3700   \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
3701 }
3702 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3703 \msg_new:nnn { ctex } { fontset-only-preamble }
3704 {
3705   The~`fontset'~option~can~be~used~only~in~preamble.
3706 }

```

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

载入中文字库。

```

3707 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3708 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

3709 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3710 </class|ctex>
3711 <*config>
3712 %%
3713 </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。

```

3714 <*ctexopts>

```

```

3715 %%
3716 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3717 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

3718 <*c19|c70>
3719 %%
3720 %% Chinese characters
3721 %%
3722 <c19>%% character set: GBK (extension of GB 2312)
3723 <c70>%% character set: Unicode
3724 %% font encoding: Unicode
3725 %%
3726 </c19|c70>

```

CJK 宏包使用的字体族。

```

3727 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3728 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3729 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3730 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3731 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3732 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

upTeX 使用的字体族。upL^AT_EX 在 NFSS 下使用字体编码 JY2 和 JT2 来分别表示横排与直排的日文。

```

3733 <*uptex|aptex>
3734 \DeclareKanjiFamily{JY2}{rm}{\hyphenchar\font\m@ne}
3735 \DeclareKanjiFamily{JY2}{sf}{\hyphenchar\font\m@ne}
3736 \DeclareKanjiFamily{JY2}{tt}{\hyphenchar\font\m@ne}
3737 \DeclareKanjiFamily{JT2}{rm}{\hyphenchar\font\m@ne}
3738 \DeclareKanjiFamily{JT2}{sf}{\hyphenchar\font\m@ne}
3739 \DeclareKanjiFamily{JT2}{tt}{\hyphenchar\font\m@ne}
3740 </uptex|aptex>
3741 <*rm>
3742 <*c19>
3743 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3744 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3745 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3746 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3747 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3748 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3749 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3750 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3751 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3752 </c19>
3753 <*c70>
3754 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3755 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3756 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3757 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3758 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3759 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3760 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3761 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3762 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3763 </c70>
3764 </rm>
3765 <*uptex|aptex>
3766 \DeclareFontShape{JY2}{rm}{m}{n}{<-> upzhserif-h}{\hyphenchar\font\m@ne}
3767 \DeclareFontShape{JY2}{rm}{m}{it}{<-> upzhserifit-h}{\hyphenchar\font\m@ne}
3768 \DeclareFontShape{JY2}{rm}{bx}{n}{<-> upzhserifb-h}{\hyphenchar\font\m@ne}

```

```

3769 \DeclareFontShape{JT2}{rm}{m}{n}{<-> upzhserif-v}{}
3770 \DeclareFontShape{JT2}{rm}{m}{it}{<-> upzhserifit-v}{}
3771 \DeclareFontShape{JT2}{rm}{bx}{n}{<-> upzhserifb-v}{}
3772 </uptex|aptex>

3773 <*sf>
3774 <*c19>
3775 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3776 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3777 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3778 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3779 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3780 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3781 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3782 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3783 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3784 </c19>
3785 <*c70>
3786 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
3787 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
3788 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
3789 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
3790 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3791 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3792 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
3793 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
3794 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
3795 </c70>
3796 </sf>
3797 <*uptex|aptex>
3798 \DeclareFontShape{JY2}{sf}{m}{n}{<-> upzhsans-h}{}
3799 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{}
3800 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{}
3801 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{}
3802 </uptex|aptex>

3803 <*tt>
3804 <*c19>
3805 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
3806 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
3807 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
3808 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
3809 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3810 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3811 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
3812 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
3813 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
3814 </c19>
3815 <*c70>
3816 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
3817 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
3818 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
3819 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
3820 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
3821 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
3822 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
3823 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
3824 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
3825 </c70>
3826 </tt>
3827 <*uptex|aptex>
3828 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{}
3829 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{}
3830 </uptex|aptex>

3831 <*fontset>

```

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

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。)

```
3832 <*windows>
3833 \file_if_exist:nTF { C:/bootfont.bin }
3834 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3835 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3836 </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。

```
3837 <*windowsnew>
3838 \tl_new:N \l__ctex_msyh_suffix_tl
3839 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
3840 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
3841 {
3842   \file_if_exist:nF { msyh.ttc }
3843   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
3844 }
3845 </windowsnew>
3846 <*windowsnew|windowsold>
3847 \sys_if_engine_pdfTeX:TF
3848 {
3849   \ctex_zhmap_case:nnn
3850   {
3851     \ctex_punct_set:n { windows }
3852     \setCJKmainfont
3853     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3854 <*windowsold>
3855     \setCJKsansfont { simhei.ttf }
3856     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3857 </windowsold>
3858 <*windowsnew>
3859     \setCJKsansfont [ BoldFont = msyhbdl__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
3860     \setCJKfamilyfont { zhyahei }
3861     [ BoldFont = msyhbdl__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
3862     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3863     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3864 </windowsnew>
3865     \setCJKmonofont { simfang.ttf }
3866     \setCJKfamilyfont { zhkai } { simkai.ttf }
3867     \setCJKfamilyfont { zhfs } { simfang.ttf }
3868     \setCJKfamilyfont { zhsong } { simsun.ttc }
3869     \setCJKfamilyfont { zhhei } { simhei.ttf }
3870     \setCJKfamilyfont { zhli } { simli.ttf }
3871     \setCJKfamilyfont { zhyou } { simyou.ttf }
3872     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3873     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3874     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3875     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3876   }
3877   {
3878     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3879     \ctex_punct_set:n { windows }
3880     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3881     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3882     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3883   }
3884   {
3885     \tl_set:Nn \CJKrmdefault { rm }
3886     \tl_set:Nn \CJKsfdefault { sf }
```

```

3887     \tl_set:Nn \CJKttdefault { tt }
3888   }
3889 }
3890 {
3891   \sys_if_engine_uptex:TF
3892   {
3893     \ctex_set_upfonts:nnnnn
3894     {simsum.ttc} {simhei.ttf} {simkai.ttf}
3895     <windowsnew> {msyh\l__ctex_msyh_suffix_tl} {msyhbd\l__ctex_msyh_suffix_tl}
3896     <windowsold> {simhei.ttf} {simhei.ttf}
3897     {simfang.ttf}
3898     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3899     \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
3900     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3901     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3902     <windowsnew> \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
3903     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
3904     \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
3905     \ctex_set_upfamily:nnn { zhyou } { upschg } {}
3906     \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
3907   }
3908   {
3909     <*windowsold>
3910     \setCJKmainfont
3911     [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3912     \setCJKsansfont { SimHei }
3913     \setCJKmonofont { FangSong_GB2312 }
3914     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3915     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3916     </windowsold>
3917     <*windowsnew>
3918     \setCJKmainfont
3919     [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3920     \setCJKsansfont
3921     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3922     \setCJKmonofont { FangSong }
3923     \setCJKfamilyfont { zhkai } { KaiTi }
3924     \setCJKfamilyfont { zhfs } { FangSong }
3925     </windowsnew>
3926     \setCJKfamilyfont { zhsong } { SimSun }
3927     \setCJKfamilyfont { zhhei } { SimHei }
3928     \setCJKfamilyfont { zhli } { LiSu }
3929     \setCJKfamilyfont { zhyou } { YouYuan }
3930     <*windowsnew>
3931     \setCJKfamilyfont { zhyahei }
3932     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3933     </windowsnew>
3934   }
3935 }
3936 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```

3937 <*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。

```

3938 \sys_if_engine_pdftex:TF
3939 {
3940   \sys_if_output_pdf:TF
3941   { \ctex_fontset_error:n { adobe } }
3942   {
3943     \ctex_zhmap_case:nnn
3944     {
3945       \setCJKmainfont
3946       [

```

```

3947         cmap = UniGB-UTF16-H ,
3948         BoldFont = AdobeHeitiStd-Regular.otf ,
3949         ItalicFont = AdobeKaitiStd-Regular.otf
3950     ] { AdobeSongStd-Light.otf }
3951     \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3952     \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3953     \setCJKfamilyfont { zhsong }
3954         [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3955     \setCJKfamilyfont { zhhei }
3956         [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3957     \setCJKfamilyfont { zhkai }
3958         [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3959     \setCJKfamilyfont { zhfs }
3960         [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3961     \ctex_punct_set:n { adobe }
3962     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3963     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3964     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3965     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3966     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3967 }
3968 {
3969     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3970     \ctex_punct_set:n { adobe }
3971     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3972     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3973     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3974 }
3975 { \ctex_fontset_error:n { adobe } }
3976 }
3977 }
3978 {
3979     \sys_if_engine_uptex:TF
3980     {
3981         \ctex_set_upfonts:nnnnnn
3982         {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
3983         {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
3984         {AdobeFangsongStd-Regular.otf}
3985         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3986         \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
3987         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3988         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3989     }
3990     {
3991         \setCJKmainfont
3992         [
3993             BoldFont = AdobeHeitiStd-Regular ,
3994             ItalicFont = AdobeKaitiStd-Regular
3995         ] { AdobeSongStd-Light }
3996         \setCJKsansfont { AdobeHeitiStd-Regular}
3997         \setCJKmonofont { AdobeFangsongStd-Regular}
3998         \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3999         \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4000         \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4001         \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4002     }
4003 }
4004 </adobe>
4005 <*fandol>

```

14.15.4 ctex-fontset-fandol.def

```

4006 \sys_if_engine_pdftex:TF
4007 {
4008     \sys_if_output_pdf:TF
4009     { \ctex_fontset_error:n { fandol } }
4010     {

```

```

4011 \ctex_zhmap_case:nnn
4012 {
4013   \setCJKmainfont
4014   [
4015     cmap = UniGB-UTF16-H ,
4016     BoldFont = FandolSong-Bold.otf ,
4017     ItalicFont = FandolKai-Regular.otf
4018   ] { FandolSong-Regular.otf }
4019   \setCJKsansfont
4020   [
4021     cmap = UniGB-UTF16-H ,
4022     BoldFont = FandolHei-Bold.otf
4023   ] { FandolHei-Regular.otf }
4024   \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4025   \setCJKfamilyfont { zhsong }
4026   [
4027     cmap = UniGB-UTF16-H ,
4028     BoldFont = FandolSong-Bold.otf
4029   ] { FandolSong-Regular.otf }
4030   \setCJKfamilyfont { zhhei }
4031   [
4032     cmap = UniGB-UTF16-H ,
4033     BoldFont = FandolHei-Bold.otf
4034   ] { FandolHei-Regular.otf }
4035   \setCJKfamilyfont { zhfs }
4036   [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4037   \setCJKfamilyfont { zhkai }
4038   [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4039   \ctex_punct_set:n { fandol }
4040   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4041   \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4042   \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4043   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4044   \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4045   \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4046 }
4047 {
4048   \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4049   \ctex_punct_set:n { fandol }
4050   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4051   \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4052   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4053 }
4054 { \ctex_fontset_error:n { fandol } }
4055 }
4056 {
4057   \sys_if_engine_uptex:TF
4058   {
4059     \ctex_set_upfonts:nnnnnn
4060     {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4061     {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4062     {FandolFang-Regular.otf}
4063     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4064     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4065     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4066     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4067   }
4068   {
4069     \setCJKmainfont
4070     [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
4071     { FandolSong-Regular }
4072     \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4073     \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4074     \setCJKfamilyfont { zhsong }
4075     [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4076     \setCJKfamilyfont { zhhei }

```



```

4078      [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4079      \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4080      \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4081    }
4082  }
4083 </fandol>
4084 <*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:

```

** 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

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

```

** 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。

```

4085 \sys_if_engine_pdftex:TF
4086 { \ctex_fontset_error:n { mac } }
4087 {
4088   \sys_if_engine_uptex:TF
4089   { \ctex_fontset_error:n { mac } }
4090   {
4091     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4092     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4093     \setCJKmonofont { STFangsong }
4094     \setCJKfamilyfont { zhsong } { STSong }
4095     \setCJKfamilyfont { zhhei } { STHeiti }
4096     \setCJKfamilyfont { zhfs } { STFangsong }
4097     \setCJKfamilyfont { zhkai } { STKaiti }
4098   }
4099 }

4100 </mac>
4101 <*founder>

```

14.15.6 ctex-fontset-founder.def

```

4102 \sys_if_engine_pdftex:TF
4103 {
4104   \ctex_zhmap_case:nnn
4105   {
4106     \setCJKmainfont
4107     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4108     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4109     \setCJKmonofont { FZFSK.TTF }
4110     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4111     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4112     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4113     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4114     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4115     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4116     \ctex_punct_set:n { founder }
4117     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4118     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4119     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4120     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4121     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4122     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4123     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4124   }
4125   {
4126     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4127     \ctex_punct_set:n { founder }
4128     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4129     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4130     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4131   }
4132   { \ctex_fontset_error:n { founder } }
4133 }
4134 {
4135   \sys_if_engine_uptex:TF
4136   {
4137     \ctex_set_upfonts:nnnnnn

```

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>

```

4138     { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4139     { FZXH1K.TTF } { FZHTK.TTF }
4140     { FZFSK.TTF }
4141     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4142     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4143     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4144     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4145     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4146     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4147     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4148     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4149   }
4150   {
4151     \setCJKmainfont
4152     [ 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），X_YTeX 找该字体时会出现明显的延迟，而用字体文件名 FZXH1K.TTF 又可能会出现大小写问题，遂采用汉字名称。由于 luaotfload 不支持汉字名称，故使用 Postscript 名字，速度不受影响。

```

4153     \setCJKsansfont [ BoldFont = FZHei-B01 ]
4154     { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4155     \setCJKmonofont { FZFangSong-Z02 }
4156     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4157     \setCJKfamilyfont { zhhei } { FZHei-B01 }
4158     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4159     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4160     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4161     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4162   }
4163 }
4164 </founder>
4165 <*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。

```

4166 \sys_if_engine_pdftex:TF
4167 {
4168   \ctex_zhmap_case:nnn
4169   {
4170     \setCJKmainfont
4171     [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4172     \setCJKsansfont { wqy-zenhei.ttc }
4173     \setCJKmonofont { uming.ttc }
4174     \setCJKfamilyfont { zhsong } { uming.ttc }
4175     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4176     \setCJKfamilyfont { zhkai } { ukai.ttc }
4177     \ctex_punct_set:n { ubuntu }
4178     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4179     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4180     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4181     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }

```

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

```

4182     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4183   }
4184   {
4185     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4186     \ctex_punct_set:n { ubuntu }
4187     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4188     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4189     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4190   }
4191   { \ctex_fontset_error:n { ubuntu } }
4192 }
4193 {
4194   \sys_if_engine_uptex:TF
4195   {
4196     \ctex_set_upfonts:nnnnnn
4197     { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4198     { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4199     { uming.ttc }
4200     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4201     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4202     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4203   }
4204   {
4205     \setCJKmainfont
4206     [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4207     \setCJKsansfont { WenQuanYi~Zen~Hei }
4208     \setCJKmonofont { AR~PL~UMing~CN }
4209     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4210     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4211     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4212   }
4213 }
4214 </ubuntu>

```

14.15.8 中文字体命令

```

4215 <*!windows>
4216 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4217 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
4218 <!ubuntu>\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4219 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4220 <*windowsnew|windowsold|founder>
4221 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4222 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4223 </windowsnew|windowsold|founder>
4224 <windowsnew>\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4225 </!windows>
4226 </fontset>
4227 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4228 \begingroup\catcode61\catcode48\catcode32=10\relax%
4229 \catcode 35=6 % #
4230 \catcode123=1 % {
4231 \catcode125=2 % }
4232 \toks0{\endlinechar=\the\endlinechar\relax}%
4233 \toks2{\endlinechar=13}%
4234 \def\x#1 #2 {%
4235   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4236   \toks2\expandafter{\the\toks2 \catcode#1=#2}%
4237 \x 13 5 % carriage return
4238 \x 32 10 % space

```

```

4239 \x 35 6 % #
4240 \x 40 12 % (
4241 \x 41 12 % )
4242 \x 45 12 % -
4243 \x 46 12 % .
4244 \x 47 12 % /
4245 \x 58 12 % :
4246 \x 60 12 % <
4247 \x 61 12 % =
4248 \x 64 11 % @
4249 \x 91 12 % [
4250 \x 93 12 % ]
4251 \x 123 1 % {
4252 \x 125 2 % }
4253 \edef\x#1{\endgroup%
4254 \edef\noexpand#1{%
4255 \the\toks0 %
4256 \let\noexpand\noexpand\noexpand#1%
4257 \noexpand\noexpand\noexpand\noexpand\undefined%
4258 \noexpand\noexpand\noexpand\noexpand\endinput}%
4259 \the\toks2}%
4260 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4261 \begingroup\expandafter\endgroup
4262 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4263 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4264 \expandafter\ifx\csname pdfoutput\endcsname\relax
4265 \iffalse\else\ifnum\pdfoutput < 1 \iffalse\else \iftrue\fi\fi
4266 \else ifctexpdf\fi
4267 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4268 \begingroup
4269 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4270 \long\def\x#1\ProvidesFile#2[#3]{%
4271 #1%
4272 \immediate\write-1{File: #2 #3}%
4273 \expandafter\xdef\csname ver@#2\endcsname{#3}}
4274 \expandafter\x%
4275 \fi
4276 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4277 <*windows>
4278 \ProvidesFile{zhwindowsfonts.tex}%
4279 [2016/08/27 v2.4.3 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4280
4281 \ifzhmappdf
4282 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4283 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4284 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4285 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4286 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4287 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4288 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4289
4290 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4291 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4292 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4293 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4294 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4295 \pdfmapline{=unili@Unicode@ <simli.ttf}
4296 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4297
4298 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}

```

```

4299 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4300 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4301 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4302 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4303 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4304
4305 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4306 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4307 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4308 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4309 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4310 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4311
4312 \else
4313 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4314 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4315 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4316 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4317 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4318 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4319 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4320
4321 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4322 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4323 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4324 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4325 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4326 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4327 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4328
4329 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4330 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4331 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4332 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4333 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4334 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4335
4336 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4337 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4338 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4339 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4340 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4341 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4342
4343 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4344 <*adobe>
4345 \ProvidesFile{zhadobefonts.tex}%
4346 [2016/08/27 v2.4.3 Adobe font map loader for DVIPDFMx (CTEX)]
4347
4348 \ifzhmappdf
4349 %% pdfTeX does not support OTF fonts
4350
4351 \else
4352 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4353 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4354 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4355 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4356 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4357
4358 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4359 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4360 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4361 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4362 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4363

```



```

4364 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4365 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4366 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4367 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4368
4369 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4370 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4371 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4372 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4373
4374 </adobe>

```

14.15.9.3 zhfandolfonts.tex

```

4375 <*fandol>
4376 \ProvidesFile{zhfandolfonts.tex}%
4377 [2016/08/27 v2.4.3 Fandol font map loader for DVIPDFMx (CTEX)]
4378
4379 \ifzhmappdf
4380 %% pdfTeX does not support OTF fonts
4381
4382 \else
4383 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4384 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4385 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4386 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4387 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4388
4389 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4390 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4391 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4392 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4393 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4394
4395 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4396 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4397 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4398 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4399
4400 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4401 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4402 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4403 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4404
4405 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4406 <*founder>
4407 \ProvidesFile{zhfounderfonts.tex}%
4408 [2016/08/27 v2.4.3 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4409
4410 \ifzhmappdf
4411 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4412 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4413 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4414 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4415 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4416 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4417 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4418
4419 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4420 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4421 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4422 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4423 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4424 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4425 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}

```

```

4426
4427 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4428 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4429 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4430 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4431 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4432 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4433
4434 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4435 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4436 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4437 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4438 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4439 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4440
4441 \else
4442 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4443 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4444 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4445 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4446 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4447 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4448 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4449
4450 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4451 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4452 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4453 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4454 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4455 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4456 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4457
4458 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4459 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4460 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4461 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4462 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4463 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4464
4465 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4466 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4467 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4468 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4469 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4470 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4471
4472 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4473 <*ubuntu>
4474 \ProvidesFile{zhubuntufonts.tex}%
4475 [2016/08/27 v2.4.3 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4476
4477 \ifzhmappdf
4478 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4479 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4480 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4481 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4482 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4483 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4484
4485 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4486 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4487 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4488 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4489 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4490 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}

```


14.15.10 制作 spa 文件

```

4532 <*spa>
4533 <*macro>
4534 \input expl3-generic %
4535 \ExplSyntaxOn
4536 \sys_if_engine_xetex:F
4537 {
4538   \msg_new:nn { ctex } { xetex }
4539   { XeTeX~is~required~to~compile~this~document! }
4540   \msg_fatal:nn { ctex } { xetex }
4541 }

```

“ ‘ “ [] { < 《 【
—… : ; ! ? %) } } »] ’ ” |

```

4542 \seq_new:N \c__ctex_punct_seq
4543 \seq_gset_from_clist:Nn \c__ctex_punct_seq

```

```

4544 {
4545   "2018", "201C", "300C", "300E", "3014", "FF08", "FF3B", "FF5B",
4546   "3008", "300A", "3016", "3010",
4547   "2014", "2026", "3001", "3002", "FF0C", "FF0E", "FF1A", "FF1B",
4548   "FF01", "FF1F", "FF05", "3015", "FF09", "FF3D", "FF5D", "3009",
4549   "300B", "3017", "3011", "2019", "201D", "300D", "300F"
4550 }

```

`\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。

```

4551 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4552 {
4553   \iow_open:Nn \g__ctex_spa_iow {#1}
4554   \clist_map_inline:nn {#2}
4555   { \__ctex_write_family:nn ##1 }
4556   \iow_close:N \g__ctex_spa_iow
4557 }
4558 \iow_new:N \g__ctex_spa_iow
4559 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn
(End definition for \ctex_make_spa:nn.)

4560 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4561 {
4562   \group_begin:
4563   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4564   \l__ctex_punct_font
4565   \clist_clear:N \l__ctex_punct_bounds_clist
4566   \seq_map_inline:Nn \c__ctex_punct_seq
4567   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4568   \iow_now:Nx \g__ctex_spa_iow
4569   {
4570     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 `CJKpunct` 来说是必要的。

```

4571     { \l__ctex_punct_bounds_clist , , , }
4572   }
4573   \group_end:
4574 }
4575 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4576 {
4577   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4578   {
4579     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4580     \__ctex_calc_bounds:nn { \c_three } {#1}
4581   }
4582 }
4583 \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。

```

4584 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4585 {
4586   \fp_eval:n
4587   {
4588     round
4589     (
4590       \dim_to_decimal_in_unit:nn
4591       { 100 \xetex_glyphbounds:D #1 ~ #2 }
4592       { 1 em }
4593     )
4594   }
4595 }
4596 \ExplSyntaxOff
4597 </macro>

```

下面是 $\text{CT}_{\text{E}}\text{X}$ 定义的一些字体。

```

4598 <*make>

```

```

4599 \input ctexspamacro %
4600
4601 \MAKESPA {ctexpunct.spa}
4602 {
4603   {adobezhhsong}      {AdobeSongStd-Light} ,
4604   {adobezhhei}        {AdobeHeitiStd-Regular} ,
4605   {adobezhkai}        {AdobeKaitiStd-Regular} ,
4606   {adobezhfs}         {AdobeFangsongStd-Regular} ,
4607   {fandolzhhsong}     {FandolSong} ,
4608   {fandolzhsongb}     {FandolSong-Bold} ,
4609   {fandolzhhei}       {FandolHei} ,
4610   {fandolzhheib}      {FandolHei-Bold} ,
4611   {fandolzhkai}       {FandolKai} ,
4612   {fandolzhfs}        {FandolFang} ,
4613   {founderzhhsong}    {FZShuSong-Z01} ,
4614   {founderzhsongb}    {FZXiaoBiaoSong-B05} ,
4615   {founderzhhei}      {FZHei-B01} ,
4616   {founderzhheil}     {FZXiHeiI-Z08} ,
4617   {founderzhkai}      {FZKai-Z03} ,
4618   {founderzhfs}       {FZFangSong-Z02} ,
4619   {founderzhli}       {FZLiShu-S01} ,
4620   {founderzhyou}      {FZXiYuan-M01} ,
4621   {founderzhyoub}     {FZZhunYuan-M02} ,
4622   {ubuntuzhsong}      {AR PL UMinG CN} ,
4623   {ubuntuzhhei}       {WenQuanYi Zen Hei} ,
4624   {ubuntuzhkai}       {AR PL UKai CN} ,
4625   {windowszhhsong}    {SimSun} ,
4626   {windowszhhei}      {SimHei} ,
4627   {windowszhkai}      {KaiTi} ,
4628   {windowszhfs}       {FangSong} ,
4629   {windowszhli}       {LiSu} ,
4630   {windowszhyou}      {YouYuan} ,
4631   {windowszhyahui}    {Microsoft YaHei} ,
4632   {windowszhyahuib}   {Microsoft YaHei Bold}
4633 }
4634
4635 \primitive\end
4636 </make>
4637 </spa>

```

14.16 translator 宏包的中文字典

```
4638 <*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。

```

4639 <*theorem>
4640 \providetranslation{Comments}{评论}
4641 \providetranslation{comments}{评论}
4642 \providetranslation{Comment}{评论}
4643 \providetranslation{comment}{评论}
4644 \providetranslation{Corollaries}{推论}
4645 \providetranslation{corollaries}{推论}
4646 \providetranslation{Corollary}{推论}
4647 \providetranslation{corollary}{推论}
4648 \providetranslation{Definitions}{定义}
4649 \providetranslation{definitions}{定义}
4650 \providetranslation{Definition}{定义}
4651 \providetranslation{definition}{定义}
4652 \providetranslation{Examples}{例}
4653 \providetranslation{examples}{例}
4654 \providetranslation{Example}{例}
4655 \providetranslation{example}{例}
4656 \providetranslation{Exercises}{练习}
4657 \providetranslation{exercises}{练习}

```

```

4658 \providetranslation{Exercise}{练习}
4659 \providetranslation{exercise}{练习}
4660 \providetranslation{Facts}{事实}
4661 \providetranslation{facts}{事实}
4662 \providetranslation{Fact}{事实}
4663 \providetranslation{fact}{事实}
4664 \providetranslation{Key Lemmas}{关键引理}
4665 \providetranslation{key lemmas}{关键引理}
4666 \providetranslation{Key Lemma}{关键引理}
4667 \providetranslation{key lemma}{关键引理}
4668 \providetranslation{Key Observations}{关键观察}
4669 \providetranslation{key observations}{关键观察}
4670 \providetranslation{Key Observation}{关键观察}
4671 \providetranslation{key observation}{关键观察}
4672 \providetranslation{Lemmas}{引理}
4673 \providetranslation{lemmas}{引理}
4674 \providetranslation{Lemma}{引理}
4675 \providetranslation{lemma}{引理}
4676 \providetranslation{Main Theorems}{主要定理}
4677 \providetranslation{main theorems}{主要定理}
4678 \providetranslation{Main Theorem}{主要定理}
4679 \providetranslation{main theorem}{主要定理}
4680 \providetranslation{Observations}{观察}
4681 \providetranslation{observations}{观察}
4682 \providetranslation{Observation}{观察}
4683 \providetranslation{observation}{观察}
4684 \providetranslation{Problems}{问题}
4685 \providetranslation{problems}{问题}
4686 \providetranslation{Problem}{问题}
4687 \providetranslation{problem}{问题}
4688 \providetranslation{Proofs}{证明}
4689 \providetranslation{proofs}{证明}
4690 \providetranslation{Proof}{证明}
4691 \providetranslation{proof}{证明}
4692 \providetranslation{Proof Sketch}{证明提要}
4693 \providetranslation{Proof sketch}{证明提要}
4694 \providetranslation{proof sketch}{证明提要}
4695 \providetranslation{Proof Sketches}{证明提要}
4696 \providetranslation{Proof sketches}{证明提要}
4697 \providetranslation{proof sketches}{证明提要}
4698 \providetranslation{Sketch of Proof}{证明提要}
4699 \providetranslation{Sketch of Proofs}{证明提要}
4700 \providetranslation{Sketch of proof}{证明提要}
4701 \providetranslation{Sketch of proofs}{证明提要}
4702 \providetranslation{sketch of proof}{证明提要}
4703 \providetranslation{sketch of proofs}{证明提要}
4704 \providetranslation{Propositions}{命题}
4705 \providetranslation{propositions}{命题}
4706 \providetranslation{Proposition}{命题}
4707 \providetranslation{proposition}{命题}
4708 \providetranslation{Remarks}{注}
4709 \providetranslation{remarks}{注}
4710 \providetranslation{Remark}{注}
4711 \providetranslation{remark}{注}
4712 \providetranslation{Solutions}{解}
4713 \providetranslation{solutions}{解}
4714 \providetranslation{Solution}{解}
4715 \providetranslation{solution}{解}
4716 \providetranslation{Theorems}{定理}
4717 \providetranslation{theorems}{定理}
4718 \providetranslation{Theorem}{定理}
4719 \providetranslation{theorem}{定理}
4720 </theorem>
4721 </dict>

```

14.17 ctexcap 宏包

4722 <ctexcap>

ctexcap 是过时宏包。

```
4723 \clist_new:N \l__ctex_ctexcap_options_clist
4724 \clist_set:Nx \l__ctex_ctexcap_options_clist
4725 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
4726 \msg_new:nnn { ctexcap } { deprecated }
4727 {
4728   Package~`ctexcap'~is~deprecated.\\
4729   Please~use~package~`ctex'~with~option~`#1'~instead: \\
4730   \iow_indent:n { \token_to_str:N \usepackage [ #1 ] \{ ctex \} } \\
4731 }
4732 \msg_warning:nnx { ctexcap } { deprecated }
4733 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }
```

ctexcap 是默认打开 heading 选项的 ctex。

```
4734 \PassOptionsToPackage { heading = true } { ctexcap }
4735 \RequirePackageWithOptions { ctex }
4736 </ctexcap>
```

14.18 ctexhook 宏包

4737 <ctexhook>

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```
\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4738 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4739 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl { #1 } }
4740 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4741 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl { #1 } }
4742 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4743 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4744 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4745 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4746 \cs_set_nopar:Npx \document
4747 {
4748   \CTEX@document@left@hook
4749   \exp_not:o { \document }
4750   \CTEX@document@right@hook
4751 }
4752 \tl_new:N \g__ctex_end_preamble_hook_tl
4753 \tl_new:N \g__ctex_after_end_preamble_hook_tl
```

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

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

```
4754 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4755 {
4756   \@ifpackageloaded { #1 }
4757   { #2 }
4758   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } { #2 } }
4759 }
```

(End definition for \ctex_at_end_package:nn.)

\ctex_gadd_hook:Nn 给钩子附加内容。

```
\ctex_gadd_hook:cn
4760 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4761 {
4762   \tl_if_exist:NF #1 { \tl_new:N #1 }
4763   \tl_gput_right:Nn #1 { #2 }
4764 }
4765 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }
```

(End definition for \ctex_gadd_hook:Nn.)

\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
4766 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
4767 {
4768   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
4769   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
4770 }
4771 \cs_generate_variant:Nn \ctex_package_end_hook:n { o }
(End definition for \ctex_package_end_hook:n.)
```

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

```
4772 \tl_put_left:Nn \@popfilename
4773 {
4774   \cs_if_eq:NNT \@currentx \pkgextension
4775   { \ctex_package_end_hook:o { \@currname } }
4776 }
4777 </ctexhook>
```

14.19 ctexpatch 宏包

4778 <*ctexpatch>

```
4779 \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。

```
4780 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
4781 {
4782   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4783   { once } {#2} { \use_i:nn } { \use_ii:nn }
4784 }
(End definition for \ctex_patch_cmd_once:NnnnTF.)
```

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

```
4785 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
4786 {
4787   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4788   { all } {#2} { \use_i:nn } { \use_ii:nn }
4789 }
(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。

```
4790 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
4791 {
4792   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4793   { once }
4794   {
4795     \ExplSyntaxOff
4796     \char_set_catcode_letter:n { 64 }
4797   }
4798   { }
4799   { \ctex_patch_failure:N #1 }
4800 }
4801 \cs_new_protected:Npn \ctex_patch_failure:N #1
4802 { \msg_warning:nx { ctex } { patch-failure } { \token_to_str:N #1 } }
4803 \msg_new:nnn { ctex } { patch-failure }
4804 { 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。

```
4805 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
4806 {
4807   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
4808   { left } {#2} { \use_i:nn } { \use_ii:nn }
4809 }
(End definition for \ctex_preto_cmd:NnnTF.)
```

`\ctex_appto_cmd:NnnTF` 在宏的原本定义后面追加钩子。

```
4810 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
4811 {
4812   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
4813   { right } {#2} { \use_i:nn } { \use_ii:nn }
4814 }
(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。

```
4815 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
4816 {
4817   \tl_set:Nn \__ctex_patch_true:w {#5}
4818   \tl_set:Nn \__ctex_patch_false:w {#6}
4819   \group_begin:
4820     \char_set_catcode_other:n { 35 }
4821     \ctex_parse_name:NN #1 #2 {#3} {#4}
4822   }
4823   \tl_new:N \__ctex_patch_true:w
4824   \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。

```
4825 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
4826 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
4827 \group_begin:
4828 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
4829 {
4830   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
4831   {
4832     \bool_if:nTF
4833     {
4834       \cs_if_exist_p:c { ##3 ~ } ||
4835       \cs_if_exist_p:c { #1##3 }
4836     }
4837     {
4838       \group_begin:

```

```

4839         \use:x
4840         {
4841             \group_end:
4842             \__ctex_parse_name:nNNNnN
4843             { \token_get_replacement_spec:N ##2 }
4844             \exp_not:N ##2
4845             \exp_not:c { ##3 ~ }
4846             \exp_not:c { #1##3 }
4847             } {##3} ##1
4848         }
4849         { ##1##2 }
4850     }
4851     \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
4852     {
4853         \exp_args:Nc ##6
4854         {
4855             \str_case:nnTF {##1}
4856             {
4857                 { \protect ##3 } { }
4858                 { \x@protect ##2 \protect ##3 } { }
4859             }
4860             {
4861                 \str_if_eq_x:nnTF
4862                 { \exp_not:n { #1@protected@ ##3 #1##3 } }
4863                 {
4864                     \exp_last_unbraced:Nf \__ctex_parse_name:w
4865                     \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
4866                 }
4867                 { #1##5 ~ } { ##5 ~ }
4868             }
4869             {
4870                 \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
4871                 {
4872                     { #1@protected@ ##2 ##4 } { }
4873                     { #1@ ##4 } { }
4874                 }
4875                 { #1##5 } {##5}
4876             }
4877         }
4878     }
4879     \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
4880 }
4881 \use:x
4882 {
4883     \__ctex_tmp:w
4884     { \c_backslash_str }
4885     { \c_left_brace_str }
4886     { \tl_to_str:n { testopt } }
4887 }
4888 \group_end:
4889 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

(End definition for \ctex_parse_name:NN.)

```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

```

分别保存宏的 \meaning 中的前缀、参数文本和替换文本。

```

4890 \str_new:N \l__ctex_prefix_str
4891 \str_new:N \l__ctex_parameter_str
4892 \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
\__ctex_get_macro_meaning:w

```

解构待补丁宏的 \meaning。若命令不是宏，则走向 false 分支。

```

4893 \group_begin:
4894 \cs_set_protected:Npn \__ctex_tmp:w #1
4895 {
4896     \prg_new_protected_conditional:Npnn

```



```

4897 \ctex_get_macro_meaning:N ##1 { TF }
4898 {
4899   \exp_after:wN \__ctex_get_macro_meaning:w
4900   \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
4901 }
4902 \cs_new_protected:Npn \__ctex_get_macro_meaning:w
4903   ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
4904 {
4905   \tl_if_empty:nTF { ##4 }
4906     { \prg_return_false: }
4907     {
4908       \str_set:Nn \l__ctex_prefix_str { ##1 }
4909       \str_set:Nn \l__ctex_parameter_str { ##2 }
4910       \str_set:Nn \l__ctex_replacement_str { ##3 }
4911       \prg_return_true:
4912     }
4913   }
4914 }
4915 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
4916 \group_end:

(End definition for \ctex_get_macro_meaning:NTF and \__ctex_get_macro_meaning:w)

```

`\ctex_if_rescanable:NnTF` 检查宏是否可以重建。

```

4917 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
4918 {
4919   \ctex_get_macro_meaning:NTF #1
4920   {
4921     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
4922     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
4923   }
4924   {#4}
4925 }
4926 \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。

```

4927 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
4928 {
4929   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
4930   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
4931   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
4932   \use:x
4933   {
4934     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
4935     \exp_not:o { \l__ctex_parameter_tl }
4936     { \exp_not:o { \l__ctex_replacement_tl } }
4937   }
4938 }
4939 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
4940 {
4941   \str_if_empty:NTF #2
4942     { \tl_clear:N #1 }
4943     { \tl_set_rescan:Nno #1 {#3} {#2} }
4944 }
4945 \tl_new:N \l__ctex_prefix_tl
4946 \tl_new:N \l__ctex_parameter_tl
4947 \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。

```

4948 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
4949 {

```

```

4950 \group_end:
4951 \ctex_if_rescanable:NnTF #1 {#3}
4952 {
4953   \use:x
4954   {
4955     \__ctex_patch_replace:nnnTF {#2}
4956     { \tl_to_str:n {#4} }
4957     { \tl_to_str:n {#5} }
4958   }
4959   {
4960     \__ctex_patch_rebuild:Nn #1 {#3}
4961     \__ctex_patch_true:w
4962   }
4963   { \__ctex_patch_false:w }
4964 }
4965 { \__ctex_patch_false:w }
4966 }

```

(End definition for __ctex_patch_cmd:Nnnnw.)

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

```

4967 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
4968 {
4969   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
4970   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
4971 }

```

(End definition for __ctex_patch_replace:nnnTF.)

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

4972 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
4973 {
4974   \group_end:
4975   \ctex_get_macro_meaning:NnTF #1
4976   {
4977     \str_if_empty:NnTF \l__ctex_parameter_str
4978     { \__ctex_hookto_cmd_parameterless:Nnnnw }
4979     { \__ctex_hookto_cmd_parameter:Nnnnw }
4980     #1 {#2} {#3} {#4}
4981   }
4982   { \__ctex_patch_false:w }
4983 }

```

(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。

```

4984 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
4985 {
4986   \str_if_empty:NnTF \l__ctex_prefix_str
4987   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
4988   \tex_edef:D #1
4989   {
4990     \use:c { __ctex_ #2 _hook_aux:nn }
4991     { \exp_not:o {#1} }
4992     { \exp_not:n {#4} }
4993   }
4994   \__ctex_patch_true:w
4995 }
4996 \cs_generate_variant:Nn \tl_rescan:nn { no }
4997 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
4998 \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。

```

4999 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4

```

```
5000 {
5001   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5002   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5003   {
5004     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5005     \__ctex_patch_rebuild:Nn #1 {#3}
5006     \__ctex_patch_true:w
5007   }
5008   { \__ctex_patch_false:w }
5009 }

(End definition for \__ctex_hookto_cmd_parameter:Nnnnw.)

5010 </ctexpatch>
```

版本历史

v2.0	(2014/03/06 – 2015/05/06)	将章节标题设置功能提取到可以独立使用的宏包 <code>ctexheading</code> 中。 1
General: <code>c5size</code> , <code>cs4size</code> 是过时选项。 37		新的标题格式选项 <code>aftertitle</code> 。 84
<code>captiondelimiter</code> 是过时选项。 108		修复 <code>ctexbook</code> 和 <code>ctexrep</code> 类的中文 <code>part/number</code> 选项初值为空的错误。 99
<code>fancyhdr</code> 成为过时选项, 原选项功能总是打开。 40		<code>\ctex_if_macosx:TF</code> : 改用 <code>/Library/Fonts/Songti.ttc</code> 为特征文件。 71
<code>fnctef</code> 成为过时选项, 原选项功能总是打开。 40		<code>hyperref</code> : 补充定义 <code>\hypersetup</code> 。 40
<code>hyperref</code> 成为过时选项, 原选项功能总是打开。 40		v2.2 (2015/06/21 – 2015/06/30)
<code>indent</code> , <code>noindent</code> 是过时选项。 37		General: <code>before skip</code> 和 <code>after skip</code> 选项的符号不再有特殊意义。 84
<code>nofonts</code> , <code>adobe fonts</code> , <code>win fonts</code> 是过时选项。 38		<code>before skip</code> , <code>after skip</code> 和 <code>indent</code> 选项支持表达式。 86
<code>nopunct</code> 是过时选项。 39		不再依赖 <code>etoolbox</code> 宏包。 1
<code>nospace</code> 是过时选项。 39		非 <code>ctexart</code> 类的 <code>part/before skip</code> 和 <code>part/after skip</code> 选项有意义。 86
<code>nozhmap</code> 是过时选项。 38		给 <code>enumitem</code> 宏包注册 <code>\chinese</code> 和 <code>\zhnum</code> 。 81
<code>punct</code> 选项可以设置标点格式。 39		将文档开头和宏包末尾钩子提取到 <code>ctexhook</code> 宏包中。 129
<code>ctex</code> 宏包新增 <code>heading</code> 选项。 39		将中文版式下的 <code>part</code> 和 <code>chapter</code> 标题的 <code>nameformat</code> 和 <code>titleformat</code> 选项的初值合并到 <code>format</code> 中。 99
<code>\CTEXindent</code> , <code>\CTEXnoindent</code> 是过时命令。 81		删去 <code>etoolbox</code> 与 <code>breqn</code> 的兼容补丁。 109
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。 14, 42		新的标题格式选项 <code>afterindent</code> 。 84
<code>\CTEXunderdot</code> , <code>\CTEXunderline</code> , <code>\CTEXunderdblline</code> , <code>\CTEXunderwave</code> , <code>\CTEXsout</code> , <code>\CTEXxout</code> 是过时命令; <code>CTEXfilltwosides</code> 是过时环境。 73		新的标题格式选项 <code>numbering</code> 。 84
标题设置新增 <code>pagestyle</code> 选项。 85		新的标题格式选项 <code>runin</code> 。 84
调整 <code>\footnotesep</code> 的大小, 以适合行距的变化。 108		新增子宏包 <code>ctexpatch</code> 实现给宏打补丁功能。 130
兼容 <code>extsizes</code> 宏包、 <code>beamer</code> 、 <code>memoir</code> 等提供的更多字号选项。 41		v2.3 (2015/09/17 – 2016/01/05)
将标题汉化功能加入 <code>ctex.sty</code> 。 82		General: <code>.value_required:</code> 和 <code>.value_forbidden:</code> 已过时。 37
将中文字号功能提取到可以独立使用的 <code>ctexsize</code> 。 105		代码实现避免使用 <code>\lowercase</code> 技巧 (Joseph Wright)。 35
解决 <code>etoolbox</code> 与 <code>breqn</code> 关于 <code>\end</code> 的冲突。 109		更新 <code>LuaTeX-j</code> 支持 (20150922.0)。 48
默认关闭 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 的彩色设置。 73		更新 <code>unicode-math</code> 宏包补丁。 49
删除 <code>c19gsn.fd</code> 和 <code>c19gkai.fd</code> 。 1		兼容 <code>titleps</code> 宏包。 94
通过 <code>LuaTeX-j</code> 宏包支持 <code>Lua\LaTeX</code> 。 48		修复 <code>nameformat</code> 作用域问题。 83
新增 <code>autoindent</code> 选项。 37		与 <code>L\LaTeXX3</code> (2015/12/20) 同步。 33
新增 <code>fontset</code> 选项。 38		v2.4 (2015/02/19 – 2016/04/25)
新增 <code>linespread</code> 选项。 37		General: 初步支持 <code>up\LaTeX</code> 。 68
新增 <code>linestretch</code> 选项。 79		加强 <code>beamer</code> 宏包支持。 1
新增 <code>scheme</code> 选项, 并将 <code>cap</code> 和 <code>nocap</code> 列为过时选项。 40		提供 <code>translator</code> 宏包的中文定理名称翻译。 127
新增 <code>zhmCJK</code> 支持选项。 38		提供 <code>up\LaTeX</code> 的 NFSS 字体定义。 111
新增 <code>zihao</code> 选项。 37		正确更新 <code>CJK</code> 包的 <code>\CJKfamilydefault</code> 。 70
新增统一设置接口 <code>\ctexset</code> 。 42		正确设置 <code>up\TeX</code> 下字体命令。 68
应用 <code>L\LaTeXX3</code> 重新整理代码。 1		<code>\ctex_parse_name:NN</code> : 修复宏名解析错误。 131
中文字号不再采用近似值。 105		v2.4.1 (2016/04/26 – 2016/05/14)
自动检测操作系统, 载入对应的字体配置。 71		General: <code>beamer</code> 不调整默认行距。 104
<code>\CJK@surr</code> : 解决与 <code>\nouppercase</code> 的冲突。 45		<code>beamer</code> 不调整默认字体大小。 104
v2.0.1 (2015/05/15)		使用 <code>bootfont.bin</code> 判断 Windows XP 以避免权限问题。 113
General: 修复 10pt、11pt 等选项无效的问题。 41		随字体更新 <code>up\TeX</code> 的 <code>\xkanjiskip</code> 。 76
v2.0.2 (2015/05/16)		新的标题格式选项 <code>part/fixbefore skip</code> 和 <code>chapter/fixbefore skip</code> 。 85
General: 修复加载 <code>ctex</code> 宏包后章节标题后第一段无段首缩进的问题。 105		正确更新 <code>up\LaTeX</code> 的 <code>\CJKfamilydefault</code> 。 70
v2.1 (2015/05/18 – 2015/06/19)		<code>\ccwd</code> : 正确设置 <code>up\TeX</code> 下的 <code>\ccwd</code> 。 74
General: <code>format+</code> , <code>nameformat+</code> 等带加号的选项, 加号与前面的文字之间可以有可选的空格。 84		<code>\chinese</code> : 支持 <code>\pagenumbering</code> 。 81
<code>nameformat</code> 可以接受章节名字为参数。 83		<code>zihao</code> : 不允许无参 <code>zihao</code> 选项。 37
不依赖 <code>ifpdf</code> 宏包。 35		
不再设置 <code>hyperref</code> 宏包的 <code>colorlinks</code> 选项。 72		
给 <code>pdf\LaTeX</code> 下的非 UTF8 编码 <code>CJK</code> 字体族加上 <code>CMap</code> 。 42		

v2.4.2	(2016/05/15)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	49
<code>\em</code> : 兼容 <code>upL^AT_EX 2016/05/07u00</code> 的定义。	68
v2.4.3	(2016/06/03 – 2016/08/26)
General: 更新 <code>unicode-math</code> 补丁。	49

简化 <code>fontspec</code> 补丁。	49
确保 <code>\proofname</code> 非空。	82
删除选项 <code>part/fixbeforeskip</code> 和 <code>chapter/fixbeforeskip</code> 。	85
新的标题格式选项 <code>fixskip</code> 。	84

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	
<code>_</code>	2970, 2974, 2990, 2994
<code>\#</code>	491, 2891, 2937
<code>.../afterindent</code>	20
<code>.../aftername</code>	18
<code>.../aftername+</code>	18
<code>.../afterskip</code>	21
<code>.../aftertitle</code>	19
<code>.../aftertitle+</code>	19
<code>.../beforeskip</code>	20
<code>.../fixskip</code>	21
<code>.../format</code>	16
<code>.../format+</code>	16
<code>.../indent</code>	21
<code>.../name</code>	15
<code>.../nameformat</code>	16
<code>.../nameformat+</code>	16
<code>.../number</code>	15
<code>.../numberformat</code>	18
<code>.../numberformat+</code>	18
<code>.../titleformat</code>	18
<code>.../titleformat+</code>	18
<code>\</code>	1612
<code>\</code>	15, 33,
	34, 102, 145, 147, 149, 205, 215, 677, 1103, 1185, 2245,
	3335, 3336, 3449, 3450, 3670, 3700, 4728, 4729, 4730, 4804
<code>\{</code>	1114, 1115, 4730
<code>\}</code>	1114, 1115, 4730
10pt	8, 366
11pt	8, 366
12pt	8, 366
A	
<code>\abovedisplayshortskip</code>	3540, 3548, 3559, 3579, 3587, 3598
<code>\abovedisplayskip</code>	3539, 3542,
	3547, 3554, 3558, 3565, 3578, 3581, 3586, 3593, 3597, 3604
<code>\abstractname</code>	2229
<code>abstractname</code>	12
<code>\addCJKfontfeature</code>	1185, 1208
<code>\addCJKfontfeatures</code>	1188
<code>\addcontentsline</code>	2457, 2458, 2461, 2464,
	2489, 2490, 2493, 2496, 2574, 2579, 2582, 2585, 2694, 2710
<code>\AddEnumerateCounter</code>	2184, 2186, 2187, 2188
<code>\addpenalty</code>	2658, 2659
<code>\addtocontents</code>	2588, 2589
<code>\addvspace</code>	2420, 2423,
	2588, 2589, 2658, 2662, 3006, 3016, 3024, 3034, 3042, 3052
<code>adobefonts</code>	28
<code>\algorithmname</code>	2236, 2252
<code>algorithmname</code>	12
<code>AlternateFont</code>	25, 1308
<code>\appendix</code>	2806
<code>appendix/name</code>	22, 2795
<code>appendix/number</code>	22, 2795
<code>appendix/numbering</code>	22, 2795
<code>\appendixname</code>	2231, 3320
<code>appendixname</code>	12
<code>\arabic</code>	3225
<code>\AtBeginDocument</code>	579, 1601
<code>\AtBeginDvi</code>	449
<code>\AtBeginShipoutFirst</code>	451, 1812, 1828
<code>\AtBeginUTFCommand</code>	721
<code>\AtEndOfClass</code>	135
<code>\AtEndOfPackage</code>	136
<code>\AtEndUTFCommand</code>	727
<code>autoindent</code>	10, 184, 2039
B	
<code>\baselinestretch</code>	2094
<code>\begin</code>	3012, 3030, 3048
<code>\begingroup</code>	2685, 2724, 2756, 3002, 3020, 3038, 4228, 4261, 4268
<code>\belowdisplayshortskip</code>	3541, 3549, 3560, 3580, 3588, 3599
<code>\belowdisplayskip</code>	3542, 3554, 3565, 3581, 3593, 3604
<code>\bfdefault</code>	966, 972, 1298, 1305
<code>\bfseries</code>	2470, 2473, 2503, 2506, 2517,
	2532, 2607, 2613, 2631, 3113, 3115, 3119, 3131, 3132,
	3135, 3157, 3159, 3164, 3181, 3185, 3206, 3239, 3255, 3268
<code>\bibname</code>	2234, 2237, 2245, 2253, 2259, 2260
<code>bibname</code>	12
bool commands:	
<code>\bool_gset_false:N</code>	265, 270
<code>\bool_gset_true:N</code>	260
<code>\bool_if:NTF</code>	70, 1143, 2857, 2867, 2878, 3339
<code>\bool_if:nTF</code>	4832

- \bool_new:N 131, 1180
- \bool_set_false:N 1155
- \bool_set_true:N 1173
- \c_false_bool 2779, 2780
- \c_true_bool 2778
- box commands:
 - \box_ht:N 3620
 - \box_new:N 61
 - \box_wd:N 1900, 2933
- C**
- c5size 28
- cap 28
- captiondelimiter 29, 3627
- \catcode 4228, 4229, 4230, 4231, 4235, 4236
- CCTF 28
- CCTfont 28
- \ccwd 24, 115, 188, 1896, 1946, 1947, 1948, 2045, 2083, 2102, 2103, 2110, 2137, 2138, 2139, 2142, 2144, 2145, 2156, 2167, 2168, 3637, 3638, 3639
- \centering 2500, 2530, 3003, 3021, 3039, 3119, 3130, 3135, 3139, 3164, 3185, 3193, 3227
- \chapter 2556, 3345
- chapter/beforeskip 30
- chapter/fixbeforeskip 30
- chapter/numbering 15
- chapter/pagestyle 21, 2372
- \chaptermark 2587
- \chaptername 3154
- char commands:
 - \char_set_catcode:nn 73, 552, 553
 - \char_set_catcode_active:N 1333
 - \char_set_catcode_letter:n 67, 1604, 2951, 4796
 - \char_set_catcode_math_superscript:n 547
 - \char_set_catcode_other:N 491, 899, 1316, 2891, 2937
 - \char_set_catcode_other:n 546, 4820
 - \char_value_catcode:n 73, 552, 553
- CharRange 25, 1308
- \Chinese 29, 2180
- \chinese 24, 2178, 2188, 3106, 3163
- \CJK 1865
- \CJKbold 3750, 3751, 3761, 3762, 3776, 3777, 3779, 3780, 3782, 3783, 3787, 3788, 3790, 3791, 3793, 3794, 3806, 3807, 3809, 3810, 3812, 3813, 3817, 3818, 3820, 3821, 3823, 3824
- \CJKfamily 526, 530, 534, 538, 1040, 1062, 1242, 1246, 1250, 1254, 1402, 1623, 4216, 4217, 4218, 4219, 4221, 4222, 4224
- \CJKfamilydefault 520, 521, 538, 594, 1236, 1237, 1254, 1262, 1264, 1268, 1275, 1288, 1619, 1628, 1702, 1706, 1714, 1727, 1729
- \CJKfilltwosides 1873
- \CJKfontspec 1188
- \CJKglue 1899, 1913, 1930, 1943
- \CJKhook 609
- \CJKnormal 3743, 3744, 3745, 3746, 3747, 3748, 3749, 3754, 3755, 3756, 3757, 3758, 3759, 3760, 3775, 3778, 3781, 3786, 3789, 3792, 3805, 3808, 3811, 3816, 3819, 3822
- \CJKpunctmapfamily 642, 643, 644, 645, 646, 647, 651, 652, 653, 654, 655, 656
- \CJKrmdefault 514, 521, 522, 526, 1110, 1121, 1211, 1237, 1238, 1242, 1264, 1267, 1268, 1616, 1619, 1620, 1710, 3872, 3874, 3875, 3880, 3881, 3882, 3885, 3962, 3965, 3966, 3971, 3972, 3973, 4040, 4043, 4044, 4050, 4051, 4052, 4117, 4120, 4121, 4128, 4129, 4130, 4178, 4181, 4182, 4187, 4188, 4189
- \CJKsfdefault 515, 523, 530, 1111, 1122, 1217, 1239, 1246, 1617, 1621, 1711, 3856, 3862, 3863, 3886, 3963, 4041, 4045, 4118, 4122, 4179
- \CJKtilde 595
- \CJKttdefault 516, 524, 534, 1112, 1123, 1222, 1240, 1250, 1618, 1622, 1712, 3873, 3887, 3964, 4042, 4119, 4180
- clearalternatefont 26, 1494
- \cleardoublepage 2434, 2556
- \clearpage 2436, 2556
- clist commands:
 - \clist_clear:N 987, 4565
 - \clist_concat:NNN 1009
 - \clist_const:Nn 622
 - \clist_gput_right:Nn 382, 394, 404, 406
 - \clist_gset:Nn 1228
 - \clist_if_empty:NTF 1321, 1340, 1580
 - \clist_map_break:n 2836, 3330
 - \clist_map_function:nN 1497, 1499, 1531, 1545
 - \clist_map_inline:Nn 612, 1433
 - \clist_map_inline:nn 367, 634, 1538, 1857, 1883, 2831, 3327, 3467, 3502, 4554
 - \clist_new:N 138, 1006, 1229, 1375, 4583, 4723
 - \clist_put_left:Nn 1010, 3396, 3422
 - \clist_put_right:Nn 1172, 1421, 4577
 - \clist_set:Nn 991, 1358, 1359, 4724
 - \clist_use:Nn 4733
- \clubpenalty 2755, 2762
- \contentsname 2224
- contentsname 11
- continuation 12
- cs commands:
 - \cs:w 3077
 - \cs_end: 3077
 - \cs_generate_variant:Nn 93, 467, 479, 951, 1074, 1088, 1161, 1181, 1182, 1354, 1391, 1493, 1531, 1539, 4765, 4771, 4889, 4996
 - \cs_gset_eq:NN 161, 162, 163, 261, 266, 271, 309, 311, 377, 381, 499, 794, 798, 1148, 1151, 3402, 3429
 - \cs_gset_nopar:Npx 795
 - \cs_gset_protected_nopar:Npn 2396, 2807
 - \cs_gset_protected_nopar:Npx 833, 1036, 1395, 1489, 2785
 - \cs_if_eq:NNTF 935, 4774, 4922, 5002
 - \cs_if_exist:NTF 44, 358, 476, 744, 789, 801, 831, 869, 1394, 1486, 1964, 3064, 3345, 3347, 4779

- \cs_if_exist_p:N 4834, 4835
- \cs_if_exist_use:N 1058, 1059
- \cs_if_exist_use:NTF 4, 772, 1080, 1508, 4768
- \cs_if_free:NTF 461, 471, 614,
629, 1145, 1297, 2058, 2072, 2184, 2850, 2916, 2921, 2925
- \cs_new:Npn
110, 117, 124, 890, 896, 900, 983, 1061, 2875, 4879, 4997
- \cs_new_eq:NN 6, 7, 8, 9, 20, 23,
95, 106, 107, 108, 133, 134, 152, 359, 468, 599, 660, 810,
835, 911, 942, 1208, 1214, 1723, 1878, 1879, 1880, 1892,
1943, 1954, 1955, 2179, 2180, 2181, 2414, 2776, 2777,
2778, 2779, 2780, 2806, 3082, 3343, 3348, 4559, 4926, 4998
- \cs_new_nopar:Npn
850, 1053, 1106, 1117, 2178, 2914, 2919, 4584
- \cs_new_protected:Npn
62, 734, 984, 1003, 1319, 1410, 1424, 1575, 1949,
2365, 2927, 3522, 3528, 4560, 4738, 4740, 4754, 4760,
4780, 4785, 4790, 4801, 4805, 4810, 4815, 4825, 4830,
4851, 4902, 4917, 4927, 4939, 4948, 4967, 4972, 4984, 4999
- \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, 770, 779, 792, 797, 799, 811, 825, 852, 867, 880,
904, 913, 930, 944, 961, 1007, 1015, 1034, 1054, 1064,
1089, 1127, 1139, 1141, 1163, 1165, 1258, 1283, 1292,
1338, 1349, 1376, 1385, 1392, 1398, 1418, 1431, 1450,
1464, 1471, 1483, 1503, 1519, 1537, 1540, 1542, 1555,
1567, 1609, 1641, 1654, 1668, 1700, 1732, 1750, 1765,
1781, 1792, 1796, 1835, 1842, 1846, 1869, 1875, 1896,
1910, 1945, 1979, 2057, 2059, 2084, 2119, 2129, 2134,
2288, 2318, 2394, 2398, 2405, 2412, 2768, 2823, 2843,
2848, 2853, 2863, 2873, 2892, 3060, 3069, 3074, 3440,
3455, 3487, 3492, 3642, 3646, 4551, 4575, 4742, 4744, 4766
- \cs_new_protected_nopar:Npx 484, 1194, 1862
- \cs_set:Npn 1317
- \cs_set_eq:NN 109,
189, 194, 199, 539, 557, 577, 604, 608, 616, 630, 751,
804, 805, 912, 943, 981, 1040, 1255, 1394, 1402, 1412,
1413, 1414, 1486, 1487, 1488, 1612, 1705, 1960, 2195,
2200, 2203, 2208, 2770, 2771, 2772, 2773, 2774, 2827, 2837
- \cs_set_nopar:Npn 1640, 1676
- \cs_set_nopar:Npx 1893, 2865, 4746
- \cs_set_protected:Npn 728, 737, 4828, 4894
- \cs_set_protected_nopar:Npn 603, 752, 1913, 3529
- \cs_set_protected_nopar:Npx 1951
- \cs_to_str:N 54, 1193, 4826
- \cs_undefine:N 1019, 1020, 1024, 1025, 1415, 4769
- cs4size 28
- \csname 2668, 2669,
2674, 2690, 2691, 2693, 2696, 2697, 2698, 2707, 2708,
2709, 2712, 2713, 2714, 4260, 4262, 4263, 4264, 4269, 4273
- \CTeX 24, 3623
- ctex commands:
 - \ctex_add_cmap:n 464, 469
 - \ctex_add_cmap:Nn 472, 474, 479
 - \ctex_add_to_selectfont:n 1949, 1970, 1998, 2071
 - \ctex_after_end_preamble:n 35, 109, 4738
 - \ctex_appto_cmd:NnnTF 495, 1603, 4810
 - \ctex_assign_heading_name:nn 2324, 2362, 2799
 - \ctex_at_end:n 135, 293, 356, 1943, 3709
 - \ctex_at_end_package:nn
130, 450, 498, 742, 759, 1806, 1808, 1826,
2182, 2395, 2883, 2924, 2926, 2948, 2978, 3088, 3625, 4754
 - \ctex_at_end_preamble:n 361, 584, 591, 1257, 1629, 4738
 - \ctex_auto_ignorespaces: 598, 608, 2008
 - \ctex_CJK_input:n 542
 - \ctex_declare_math_sizes:nnnn 3487, 3521
 - \ctex_default_pt:n 113, 117
 - \ctex_detected_platform: 1731, 3650, 3659
 - \ctex_family_cmap:nn 459, 468
 - \ctex_file_input:n 77, 397,
661, 2038, 3354, 3389, 3394, 3399, 3400, 3415, 3420,
3426, 3427, 3531, 3533, 3651, 3655, 3663, 3709, 3834, 3835
 - \ctex_file_wrapper:nnn 62, 78, 81, 544
 - \ctex_fix_varioref_label:n 3072, 3074
 - \ctex_fontset_error:n
3642, 3941, 3975, 4009, 4054, 4086, 4089, 4132, 4191
 - \ctex_gadd_hook:Nn 609, 4758, 4760
 - \ctex_get_macro_meaning:N 4897
 - \ctex_get_macro_meaning:NTF 4893, 4919, 4975
 - \ctex_hypersetup:n
72, 359, 1789, 1801, 1804, 1819, 1822, 1825
 - \ctex_if_autoindent_touched:TF 133, 189, 194, 199, 3381
 - \ctex_if_ccglue_touched:TF 1927, 2090
 - \ctex_if_ccglue_touched_p: 1927
 - \ctex_if_macosx:TF 1772, 1773, 1775, 1781
 - \ctex_if_preamble:TF 108, 3678
 - \ctex_if_rescanable:NnTF 4917, 4951
 - \ctex_ignorespaces_case:N 601, 2005, 2008, 2011
 - \ctex_load_fontset: 3646, 3686, 3708
 - \ctex_load_zhmap:nnnn 512, 3878, 3969, 4048, 4126, 4185
 - \ctex_ljt_add_font_features:n 1163, 1205
 - \ctex_ljt_add_font_features:nn 1163, 1164
 - \ctex_ljt_char_range_key:nn 1572, 1575
 - \ctex_ljt_clear_alternate_font:n 1497, 1503
 - \ctex_ljt_declare_alternate_shape:nnnnnn 1404, 1424
 - \ctex_ljt_declare_char_range:n 1534, 1537, 1539
 - \ctex_ljt_declare_char_range:nn 1541, 1542
 - \ctex_ljt_def_char_range_key:n 1547, 1567
 - \ctex_ljt_ensure_default_family: 1258, 1719
 - \ctex_ljt_extract_font: 804, 811
 - \ctex_ljt_family_if_exist:nN 1075
 - \ctex_ljt_family_if_exist:nNTF
1066, 1262, 1267, 1285, 1288, 1506
 - \ctex_ljt_family_if_exist:nTF 1075
 - \l_ctex_ljt_family_tl
1068, 1073, 1130, 1131, 1146, 1150, 1153, 1164, 1500, 1501
 - \ctex_ljt_fontspec:nn 1127, 1139, 1174, 1200
 - \ctex_ljt_get_and_define_fonts:nN 904
 - \ctex_ljt_get_and_define_fonts_al:nN 908, 911
 - \ctex_ljt_get_and_define_fonts_ja:nN 907, 913

- `\ctex_ltj_if_alternate_shape_exist:nTF` 814, 827, 845, 854
`\ctex_ltj_if_jfont:nTF` 55, 890, 901, 906
`\ctex_ltj_if_jfont_math:NTF` 896, 937
`\ctex_ltj_math_group_hook:` 54, 752, 936, 942
`\ctex_ltj_pickup_font:` ... 785, 799, 874, 916, 918, 920
`\ctex_ltj_reset_alternate_font:n` 1499, 1519
`\ctex_ltj_save_alternate_family:Nnnn` 1369, 1389, 1392
`\ctex_ltj_save_alternate_seq:Nn` 1344, 1349
`\ctex_ltj_save_alternate_seq:Nnnwn` 1349
`\ctex_ltj_save_alternate_seq:Nnnwnw` 1352, 1355
`\ctex_ltj_save_char_range:n` 1545, 1552
`\ctex_ltj_save_char_range:nn` 1554, 1555
`\ctex_ltj_select_alternate_font:` 852, 1973
`\ctex_ltj_select_font:` 770, 1972
`\ctex_ltj_set_alternate_family:nn` 1347, 1376
`\ctex_ltj_set_alternate_family:nnnn` . 1388, 1396, 1398
`\ctex_ltj_set_alternate_family:Nnnnn` 1380, 1385, 1391
`\ctex_ltj_set_alternate_family:nnTF` 992, 1338
`\ctex_ltj_set_alternate_seq:n` 1310, 1315
`\ctex_ltj_set_alternate_shape:n` ... 1452, 1464, 1475
`\ctex_ltj_set_alternate_shape:nnN` 1437, 1471
`\ctex_ltj_set_alternate_shape:nnn` 1440, 1447
`\ctex_ltj_set_alternate_shape:nnnn` 1449, 1450
`\ctex_ltj_set_alternate_shape:Nnnnnnn` .. 1427, 1431
`\ctex_ltj_set_family:nnnn`
... 984, 1003, 1158, 1189, 1196, 1211, 1217, 1222, 1226
`\ctex_ltj_set_math_letter:NN` 734, 746, 754
`\ctex_ltj_subst_font:` 805, 825
`\ctex_ltj_swap_cs:NN` 1403, 1410
`\ctex_ltj_switch_family:n`
... 1063, 1064, 1074, 1131, 1159, 1195
`\ctex_ltj_update_mathfont:` 1280, 1283
`\ctex_ltj_update_mathfont:n` 1286, 1289, 1292
`\ctex_ltj_use_math_group:Nn` 751, 930
`\ctex_make_spa:nn` 4551
`\ctex_mono_jfm:n` 944, 952, 2028
`\ctex_package_end_hook:n` 4766, 4775
`\ctex_parse_name:NN` 4821, 4825
`\ctex_parse_name:NNn` 4826, 4830, 4889
`\ctex_patch_boot:NNnnTF`
... 4782, 4787, 4792, 4807, 4812, 4815
`\ctex_patch_cmd:Nnn`
... 1636, 1678, 2894, 2900, 2906, 2959, 2963,
2969, 2973, 2981, 2984, 2989, 2993, 3637, 3638, 3639, 4790
`\ctex_patch_cmd_all:NnnnTF` 4785
`\ctex_patch_cmd_once:NnnnTF` 1630, 4780
`\ctex_patch_failure:N` 501, 528, 532, 536, 540,
1244, 1248, 1252, 1256, 1607, 1962, 2944, 2954, 4799, 4801
`\ctex_plane_to_utfxvibe:Nn` 558
`\ctex_preto_cmd:NnnTF` 525, 529,
533, 537, 1241, 1245, 1249, 1253, 1956, 2941, 2950, 4805
`\ctex_punct_map_bfseries:nn`
... 632, 3863, 3875, 3881, 3966,
3972, 4044, 4045, 4051, 4121, 4122, 4123, 4129, 4182, 4188
`\ctex_punct_map_family:nn` 627, 3856, 3862,
3872, 3873, 3880, 3962, 3963, 3964, 3971, 4040, 4041,
4042, 4050, 4117, 4118, 4119, 4128, 4178, 4179, 4180, 4187
`\ctex_punct_map_itshape:nn` 649,
3874, 3882, 3965, 3973, 4043, 4052, 4120, 4130, 4181, 4189
`\ctex_punct_map_series:nnn` 636, 637, 640
`\ctex_punct_set:n` 610,
3851, 3879, 3961, 3970, 4039, 4049, 4116, 4127, 4177, 4186
`\ctex_punct_space:nn` 658
`\ctex_scheme_input:n` 79, 3090, 3356
`\ctex_select_size:` .. 2046, 2053, 2056, 2057, 2078, 2132
`\ctex_set_default_ccwd:Nn` 110, 198, 2052, 2077
`\ctex_set_font_size:Nnn` 3522, 3537,
3545, 3556, 3567, 3568, 3569, 3570, 3571, 3572, 3573,
3576, 3584, 3595, 3606, 3607, 3608, 3609, 3610, 3611, 3612
`\ctex_set_ignorespaces:` 601, 609
`\ctex_set_upfamily:nnn`
... 1641, 3898, 3899, 3900, 3901, 3902,
3903, 3905, 3985, 3986, 3987, 3988, 4064, 4065, 4066,
4067, 4141, 4142, 4143, 4144, 4145, 4147, 4200, 4201, 4202
`\ctex_set_upfonts:nnnnnn`
... 1668, 3893, 3981, 4060, 4137, 4196
`\ctex_set_upmap:nnn`
... 1654, 1670, 1671, 1672, 1673, 3904, 3906, 4146, 4148
`\ctex_set_zhmap:n` 446, 517, 1656
`\ctex_titleps_hook:` 2890, 2925, 2926
`\ctex_titlesec_hook:` 2823, 2887
`\ctex_update_ccglue:` 1910, 2097, 2116, 2160
`\ctex_update_ccwd:` 1896, 2091, 2168
`\ctex_update_default_family:` 46, 584, 1257, 1629, 1699
`\ctex_update_em_unit:` 1945, 2086, 2136
`\ctex_update_parindent:` 2064, 2119, 2161
`\ctex_update_size:` 2058, 2059
`\ctex_update_stretch:` 2063, 2084
`\ctex_update_xkanjiskip:` 1978, 1998
`\ctex_update_ziju:` 2066, 2134
`\ctex_varioref_hook:` 3069, 3080, 3088
`\ctex_zhmap_case:nnn`
... 134, 261, 266, 271, 3849, 3943, 4011, 4104, 4168
`\ctex_zihao:n` 3439, 3440
`\ctex_ziju:n` 2128, 2129
ctex internal commands:
`\g__ctex_after_end_preamble_hook_tl` 4741, 4745, 4753
`__ctex_assign_heading_name:nnn` 2362
`\l__ctex_autoindent_tl`
... 132, 188, 193, 198, 2045, 2049, 2052, 2121, 2124
`__ctex_calc_bounds:nn` 4579, 4580, 4584
`__ctex_ccglue:` 1930, 1943
`\l__ctex_ccglue_skip` 1914, 1918,
1921, 1924, 1926, 1934, 1938, 2095, 2106, 2115, 2151, 2159
`\c__ctex_class_tl`
.. 84, 86, 409, 414, 419, 424, 3343, 3349, 3351, 3352, 3354
`__ctex_clear_fntef_color:n` ... 1835, 1842, 1846, 1861
`\c__ctex_cmap_encoding_seq` 456, 463
`\l__ctex_ctexcap_options_clist` 4723, 4724, 4733
`__ctex_def_heading_keys:n` 2318, 2386

`_ctex_default_pt:w` 119, 124
`\l_ctex_encoding_tl` 128, 221,
 222, 505, 508, 586, 594, 1802, 2034, 2176, 3384, 3411, 3425
`\g_ctex_end_preamble_hook_tl` 4739, 4743, 4752
`\c_ctex_engine_file_str` 43, 2038
`\c_ctex_engine_str` 43
`\l_ctex_family_default_init_tl` 1702, 1722
`_ctex_family_default_wrap:n` 1705, 1723, 1726
`_ctex_fix_varioref_label:n` 3074
`\g_ctex_font_size_flag` 151,
 161, 162, 163, 377, 381, 403, 3401, 3402, 3428, 3429, 3530
`\c_ctex_font_size_prop` 3442, 3453, 3497, 3524
`\c_ctex_font_size_seq` 3451, 3454, 3465
`\l_ctex_font_size_tl` 3442, 3443, 3489, 3490, 3524, 3525
`\g_ctex_fontset_tl` 225, 1734, 1752, 1777, 1784, 1785,
 3648, 3651, 3654, 3655, 3661, 3663, 3683, 3685, 3690, 3707
`_ctex_get_font_sizes:Nn` 3489, 3492
`_ctex_get_macro_meaning:w` 4893
`\l_ctex_heading_bool` 304, 3339
`\l_ctex_heading_skip` ... 2402, 2409, 2411, 2413, 2414
`\c_ctex_headings_seq` 2278, 2383, 3071
`_ctex_hookto_cmd:Nnnnw` 4807, 4812, 4972
`_ctex_hookto_cmd_parameter:Nnnnw` 4979, 4999
`_ctex_hookto_cmd_parameterless:Nnnnw` . 4978, 4984
`_ctex_initial_heading:n` 2288, 2385
`_ctex_left_hook_aux:nn` 4997
`\l_ctex_line_spread_fp`
 108, 181, 3403, 3404, 3430, 3431, 3616, 3618
`\l_ctex_line_stretch_tl` 2077, 2082, 2083, 2087
`\l_ctex_ltj_add_alternate_bool` 1143, 1155, 1173, 1180
`_ctex_ltj_alternate_cs:n`
 1020, 1023, 1024, 1025, 1058, 1059, 1061, 1146,
 1149, 1150, 1152, 1153, 1344, 1381, 1443, 1468, 1508, 1525
`\l_ctex_ltj_alternate_family_tl` 1406, 1409
`\l_ctex_ltj_alternate_options_clist`
 1359, 1360, 1371, 1375
`\l_ctex_ltj_alternate_seq`
 988, 1328, 1329, 1337, 1343, 1351, 1599
`\l_ctex_ltj_base_CJKfamily_tl`
 989, 1005, 1056, 1443, 1468
`\l_ctex_ltj_base_family_tl` 65,
 1021, 1051, 1057, 1387, 1428, 1487, 1506, 1511, 1512, 1524
`_ctex_ltj_change_encoding:` 953, 1039, 1401
`\l_ctex_ltj_char_range_clist` 987,
 1312, 1321, 1340, 1358, 1370, 1382, 1405, 1427, 1578, 1580
`_ctex_ltj_char_range_parse_feature:w` . 1583, 1587
`\g_ctex_ltj_char_range_prop` 1435, 1542
`\l_ctex_ltj_char_range_tl`
 1435, 1438, 1544, 1546, 1548, 1550, 1557
`_ctex_ltj_check_family:n` 1015, 1342
`\l_ctex_ltj_current_font_tl`
 50, 772, 776, 777, 789, 831, 836
`\l_ctex_ltj_current_shape_tl` 854, 859, 864, 865
`_ctex_ltj_declare_char_range:nn` 1538, 1540
`\g_ctex_ltj_default_features_clist` 1009, 1228, 1229
`_ctex_ltj_family_csname:n` ... 1019, 1036, 1053, 1080
`\g_ctex_ltj_family_font_name_prop`
 994, 1012, 1017, 1091, 1167, 1260, 1272
`\g_ctex_ltj_family_font_options_prop` 995, 1012, 1170
`\g_ctex_ltj_family_int` . 1133, 1135, 1420, 1421, 1423
`\g_ctex_ltj_family_name_prop`
 1012, 1021, 1043, 1077, 1387
`_ctex_ltj_family_unknown_warning:n` 1071, 1089, 1516
`\l_ctex_ltj_font_options_clist` 991, 996,
 997, 998, 1006, 1042, 1171, 1172, 1175, 1378, 1379, 1383
`_ctex_ltj_fontspec:nnn` 1134, 1141, 1161
`\g_ctex_ltj_fontspec_family_tl`
 1041, 1044, 1045, 1046, 1052, 1057, 1082
`\g_ctex_ltj_fontspec_prop` 1129, 1157, 1162
`_ctex_ltj_gset_family_cs:nn` 999, 1034
`_ctex_ltj_if_jfont_math:w` 897, 900
`\l_ctex_ltj_jfm_tl` 944, 965, 967, 1010
`\c_ctex_ltj_math_fam_int` 973, 974
`\c_ctex_ltj_math_family_tl`
 1294, 1296, 1298, 1301, 1305
`\c_ctex_ltj_math_tl`
 968, 969, 971, 973, 1226, 1285, 1295, 1300, 1304
`_ctex_ltj_msg_def_family_map:n` 1104, 1106
`_ctex_ltj_msg_family_map:n` 1033, 1103, 1117
`_ctex_ltj_noembed_wrap:n` 981, 983
`_ctex_ltj_patch_external_font:w` 817, 850
`_ctex_ltj_pop_fontname:` 788, 797, 876
`_ctex_ltj_push_fontname:n` 784, 792, 873
`\g_ctex_ltj_reset_alternate_prop`
 1026, 1510, 1522, 1530
`_ctex_ltj_save_alternate_shape:Nnn` 1442, 1467, 1483
`_ctex_ltj_save_fontname:w` 794, 798
`_ctex_ltj_select_font_aux:` 773, 779
`_ctex_ltj_set_alternate_family:n` 1047, 1054
`_ctex_ltj_tmp:w` 1317, 1334, 1412, 1414, 1415
`\l_ctex_ltj_tmp_seq` 1325, 1326, 1329, 1336
`\l_ctex_ltj_tmp_tl`
 990, 991, 1017, 1028, 1031, 1168, 1176,
 1192, 1195, 1196, 1262, 1267, 1285, 1286, 1288, 1289,
 1323, 1324, 1325, 1362, 1364, 1365, 1372, 1582, 1590, 1599
`\g_ctex_ltj_unknown_family_seq` ... 1093, 1095, 1100
`_ctex_ltj_update_family_uid:N` 997, 1378, 1417
`_ctex_ltj_use_global_options:N` 998, 1007, 1360, 1379
`\c_ctex_macosx_file_str` 1783, 1787
`\l_ctex_msyh_suffix_tl`
 3838, 3839, 3843, 3859, 3861, 3895
`\l_ctex_parameter_str` 4890, 4909, 4930, 4977
`\l_ctex_parameter_tl` 4930, 4935, 4946
`_ctex_parse_name:nNNnnN` 4842, 4851
`_ctex_parse_name:w` 4864, 4870, 4879
`_ctex_patch_cmd:Nnnnw` 4782, 4787, 4792, 4948
`_ctex_patch_false:w` 4818, 4824, 4963, 4965, 4982, 5008
`_ctex_patch_rebuild:Nn` 4921, 4927, 4960, 5001, 5005
`_ctex_patch_replace:nnnTF` 4955, 4967
`_ctex_patch_rescan:NNn` 4929, 4930, 4931, 4939
`_ctex_patch_true:w` 4817, 4823, 4961, 4994, 5006
`\l_ctex_prefix_str` 4890, 4908, 4929, 4986, 4987

\l__ctex_prefix_tl	4929, 4934, 4945	\CTEXoptions	29, 432
\l__ctex_punct_bounds_clist	4565, 4571, 4577, 4583	\ctexset	5, 429, 436, 442
\c__ctex_punct_family_clist	612, 622	\CTEXsetfont	29, 2056
\l__ctex_punct_font	4563, 4564	\CTEXsetup	29, 432
\c__ctex_punct_seq	4542, 4543, 4566	\CTEXsout	29
\l__ctex_punct_tl		\ctexspadef	658, 4570
.....	280, 581, 582, 669, 952, 2025, 2026, 2027, 2028	\CTEXthechapter	16, 2573, 2576, 2971, 2991, 2992
__ctex_rebuild_cmd:w	4921, 4922, 4926, 5001, 5002	\CTEXtheparagraph	16
\l__ctex_replacement_str		\CTEXthepart	16, 2458, 2490
.....	4890, 4910, 4931, 4969, 4970, 5004	\CTEXthesection	16, 2961, 2975, 2983, 2995
\l__ctex_replacement_tl	4931, 4936, 4947	\CTEXthesubparagraph	16
__ctex_right_hook_aux:nn	4998	\CTEXthesubsection	16, 2965, 2986
__ctex_save_bounds:n	4567, 4575	\CTEXthesubsubsection	16
__ctex_save_cmap:Nn	476, 480	\CTEXunderdblline	29
__ctex_save_font_size:nn	3453	\CTEXunderdot	29
__ctex_save_selectfont:	1955, 1960	\CTEXunderline	29
\l__ctex_scheme_tl	315, 3090, 3356	\CTEXunderwave	29
\g__ctex_section_depth_flag		\CTEXxout	29
.....	94, 309, 311, 3278, 3294, 3310	\CurrentOption	394
\c__ctex_section_headings_seq	2273, 2280, 2782, 2841		
__ctex_set_font_size:nnNn	3525, 3528	D	
\g__ctex_spa_iow	4553, 4556, 4558, 4568	declarecharrange	25, 1532
\c__ctex_std_class_tl	3327, 3342, 3343, 3409, 3432	\DeclareFontEncoding	954
\g__ctex_std_options_clist		\DeclareFontFamily	
.....	137, 382, 394, 404, 406, 410, 415, 420, 425	490, 963, 3727, 3728, 3729, 3730, 3731, 3732
__ctex_titlesec_format:n	2837, 2845, 2848	\DeclareFontShape	
__ctex_titlesec_format:Nn	2827, 2851, 2853	964, 966, 1645, 1646, 1649, 1650, 3743, 3744,
__ctex_titlesec_format_explicit:Nn	2828, 2863		3745, 3746, 3747, 3748, 3749, 3750, 3751, 3754, 3755,
__ctex_titlesec_hook:n	2837, 2841, 2843		3756, 3757, 3758, 3759, 3760, 3761, 3762, 3766, 3767,
__ctex_titlesec_spacing:Nn	2846, 2873		3768, 3769, 3770, 3771, 3775, 3776, 3777, 3778, 3779,
__ctex_titlesec_spacing:nnnnnn	2874, 2875		3780, 3781, 3782, 3783, 3786, 3787, 3788, 3789, 3790,
__ctex_tmp:w	4828, 4883, 4894, 4915		3791, 3792, 3793, 3794, 3798, 3799, 3800, 3801, 3805,
\l__ctex_tmp_box	58, 1899, 1900, 2929, 2933		3806, 3807, 3808, 3809, 3810, 3811, 3812, 3813, 3816,
\l__ctex_tmp_dim	58,		3817, 3818, 3819, 3820, 3821, 3822, 3823, 3824, 3828, 3829
.....	2087, 2088, 2093, 2096, 2102, 2141, 2143, 2145, 2154, 2156	\DeclareFontSubstitution	957
\l__ctex_tmp_int	58, 562,	\DeclareKanjiFamily	
.....	564, 565, 567, 571, 573, 2101, 2103, 2104, 2110, 2111,	1643, 1644, 3734, 3735, 3736, 3737, 3738, 3739
.....	2143, 2144, 2145, 2148, 2149, 2154, 2156, 2781, 2784, 2788	\DeclareMathSizes	3490
\l__ctex_tmp_tl	58, 366, 373, 388, 389, 482, 483,	\DeclareSymbolFont	969, 1295
.....	561, 563, 2320, 2374, 2381, 2391, 3408, 3409, 3497, 3498	\def	2453, 2485, 2512,
\c__ctex_upper_case_file_str	1771, 1779		2528, 2538, 2567, 2596, 2622, 2641, 2667, 2670, 2703,
\c__ctex_version_tl	3, 6, 7, 8, 9, 20, 23		2720, 2733, 2734, 2738, 3550, 3561, 3589, 3600, 4234, 4270
__ctex_write_family:nn	4555, 4560	\defaultCJKfontfeatures	1209
\l__ctex_xkanjiskip_skip	1978	\defbeamertemplate	3001, 3019, 3037
\l__ctex_xkanjiskip_tl	1985, 1990, 1991	\defbeamertemplatealias	3055, 3056, 3057
\c__ctex_zero_tl	2061, 2068, 2070	dim commands:	
\g__ctex_zhmCJK_bool	131, 260, 265, 270, 504, 585	\dim_abs:n	2155
\l__ctex_ziju_dim	2137, 2138, 2142, 2153, 2155, 2159, 2163	\dim_add:Nn	2138
\l__ctex_ziju_tl	2061, 2069, 2070, 2131, 2137	\dim_compare:nNnTF	112, 2088, 2123, 2139, 2146, 2401
\ctexbibname	2246, 2266	\dim_eval:n	2108, 2154
\CTEXcounter	2181	\dim_max:nn	2932
\CTEXdigits	24, 2191	\dim_min:nn	2155
\CTEXfilltwosides	29, 1869, 1879	\dim_new:N	60, 1909, 2163
\CTEXindent	29, 2164	\dim_set:Nn	1900, 1946, 1947,
\CTEXnoindent	29, 2164		1948, 2087, 2093, 2124, 2137, 2141, 2168, 2408, 2930, 3620
\CTEXnumber	24, 2191	\dim_sub:Nn	2145

\dim_to_decimal:n	35, 3461, 3462, 3499	1370, 1371, 1372, 1396, 1490, 1705, 1727, 1952, 2391,
\dim_to_decimal_in_unit:nn	4590	2877, 3066, 4725, 4749, 4862, 4934, 4935, 4936, 4991, 4992
\dim_use:N	120, 3620	\exp_stop_f: 731
\dim_zero:N	2174	\expandafter
\c_max_dim	2088, 2401	4235, 4236, 4260, 4261, 4262, 4263, 4264, 4269, 4273, 4274
\c_zero_dim	2096, 2108, 2123, 2139, 2146, 2356, 3311	\ExplSyntaxOff
\dimexpr	2468, 2515, 2602, 2628, 3007, 3025, 3043	66, 72, 495, 525, 529, 533, 537, 689,
\directlua	707, 708	1241, 1245, 1249, 1253, 1631, 1956, 2941, 3000, 4596, 4795
\document	4746, 4749	\ExplSyntaxOn
		71, 718, 3058, 4535
		F
E		\familydefault 1708
\edef	4253, 4254	fancyhdr 28, 341
\else	1632, 1633, 1637, 1638, 2426, 2435,	\fangsong 4218
	2443, 2459, 2463, 2491, 2495, 2556, 2563, 2577, 2581,	\fi
	2584, 2592, 2652, 2657, 2673, 2677, 2695, 2702, 2711,	1692, 2418, 2422, 2428, 2437, 2445, 2448, 2462, 2465,
	2732, 2748, 2761, 4265, 4266, 4312, 4351, 4382, 4441, 4504	2472, 2480, 2494, 2497, 2505, 2523, 2541, 2549, 2550,
else commands:		2553, 2556, 2565, 2580, 2583, 2586, 2595, 2599, 2609,
\else:	507, 848, 894, 1931, 3624	2610, 2611, 2619, 2625, 2637, 2642, 2648, 2654, 2661,
\em	1630	2663, 2680, 2681, 2699, 2700, 2715, 2716, 2718, 2736,
\eminnershape	1632, 1633, 1638, 1640	2745, 2764, 2765, 2961, 2965, 2971, 2975, 2983, 2986,
\end	3015, 3033, 3051, 4635	2991, 2992, 2995, 3011, 3029, 3047, 4265, 4266, 4275, 4528
\endCJKfilltwosides	1875	fi commands:
\endcsname	2668, 2669, 2674,	\fi: 407, 541, 578, 597, 848, 894, 1931, 2966, 2976, 3534, 3624
	2690, 2691, 2693, 2696, 2697, 2698, 2707, 2708, 2709,	\figurename 2227
	2712, 2713, 2714, 4260, 4262, 4263, 4264, 4267, 4269, 4273	figurename 11
\endCTEXfilltwosides	1875, 1880	file commands:
\endgroup	2692, 2731, 2756, 3017, 3035, 3053, 4253, 4261, 4276	\g_file_current_name_tl 681, 1780
\endinput	4258	\file_if_exist:nTF
\endlinechar	4232, 4233	48, 1767, 1769, 1771, 1783, 3654, 3833, 3840, 3842
etex commands:		\file_if_exist_input:nTF 86
\etex_dimexpr:D	120, 2102	\file_input:n 78, 87, 89, 517, 550
\etex_glueexpr:D	2094	fntef 29, 341
\everypar	2656, 2751, 2763	\font
\EverySelectfont	1688, 1964, 1967	730, 3727, 3728, 3729, 3730, 3731, 3732
exp commands:		font commands:
\exp_after:wN		\font_glyph_if_exist:NnTF 730
	119, 801, 817, 819, 894, 897, 2874, 3443, 3490, 3525, 4899	fontset 7, 225, 3674
\exp_args:cc	3065	\fontsize 3443
\exp_args:Nc	2846, 2851, 4853	fontspec commands:
\exp_args:Nnc	98	\g_fontspec_encoding_tl 962
\exp_args:NNf	1590	\l_fontspec_family_tl 1429
\exp_args:NNNo	3619	\fontspec_set_family:Nnn 1041, 1406
\exp_args:Nno	3077	\fontspec_visible_space: 728
\exp_args:No	563, 1780, 2870, 4567, 4915	\fontspec_visible_space_fallback: 732
\exp_args:Nx	293, 591, 665, 2128, 3439, 3618	fontspec internal commands:
\exp_last_unbraced:Nn	4864	_fontspec_namewrap:n 55, 981
\exp_not:N	124, 125, 486, 593, 594, 595, 763,	\footnotesep 3619
	764, 766, 836, 872, 957, 1043, 1044, 1045, 1046, 1369,	\footnotesize 3556, 3595, 3619
	1490, 1710, 1711, 1712, 1726, 1864, 1865, 1866, 1893,	fp commands:
	2294, 2295, 2296, 2301, 2303, 2304, 2305, 2306, 2309,	\fp_compare:nNnTF 3403, 3430, 3616
	2310, 2313, 2325, 2326, 2327, 2328, 2329, 2330, 2331,	\fp_eval:n 2131, 4586
	2332, 2333, 2334, 2335, 2337, 2338, 2339, 2342, 2344,	\fp_set:Nn 3404, 3431
	2346, 2348, 2350, 2352, 2787, 2789, 2790, 2791, 2792,	\fp_use:N 2068, 3618
	2858, 2859, 2868, 2869, 3459, 3661, 4844, 4845, 4846, 4934	\c_nan_fp 182
\exp_not:n		\c_zero_fp 2068, 3403, 3430, 3616
	69, 388, 765, 767, 1040, 1041, 1042, 1175, 1176,	G
		GBK 6, 221
		\gdef 1683, 2811, 2812, 2813, 2814, 2817, 2818, 2819, 2820

`\global` 2559, 2750, 2753
`\globaljfont` 819
`\glueexpr` 2687, 2704,
 2726, 2733, 2734, 2760, 3006, 3016, 3024, 3034, 3042, 3052
group commands:
`\group_begin:` 490, 723, 736, 781, 803, 829, 871, 898, 986,
 1038, 1315, 1400, 1485, 1505, 1521, 1704, 2300, 2308,
 2372, 2890, 2936, 3619, 4562, 4743, 4819, 4827, 4838, 4893
`\group_end:` 494,
 503, 727, 740, 786, 807, 842, 875, 902, 1001, 1048,
 1335, 1407, 1491, 1517, 1528, 1716, 2311, 2315, 2390,
 2913, 2940, 3619, 4573, 4743, 4841, 4888, 4916, 4950, 4974
`\group_insert_after:N` 839, 840
`\gtfamily` 1632, 1637

H

hbox commands:
`\hbox_set:Nn` 1899, 2929
heading 9, 304
`\heiti` 4217
`\hskip` ... 2687, 2704, 2726, 2733, 2734, 2759, 2760, 2982, 2985
`\hspace` 2457, 2458, 2489, 2490, 2576
`\Huge` 2506, 2532, 2613, 2631, 3132, 3159, 3573, 3612
`\huge` 2473,
 2503, 2517, 2607, 3115, 3131, 3135, 3157, 3164, 3572, 3611
hyperref 29, 341
`\hypersetup` 358, 359, 1793
`\hyphenchar` 3727, 3728, 3729, 3730, 3731, 3732

I

if commands:
`\if_bool:N` 504, 585
`\if_case:w` 403, 3530
`\if_cs_exist:N` 558
`\if_false:` 107
`\if_meaning:w` 1930
`\if_true:` 106
`\ifctexpdf` 105
`\ifdim` 2646, 2683, 2722, 2740
`\ifincsname` 3624
`\IfNoValueTF` ... 437, 443, 1193, 1454, 1559, 1592, 1594, 2368
`\ifnum` 1690,
 2454, 2469, 2486, 2502, 2568, 2604, 2671, 2695, 2711, 4265
`\ifodd` 2422,
 2424, 2448, 2455, 2469, 2480, 2487, 2502, 2523, 2541,
 2561, 2570, 2599, 2606, 2619, 2625, 2637, 2650, 2661,
 2674, 2684, 2696, 2712, 2723, 2741, 2745, 2961, 2965,
 2971, 2975, 2983, 2986, 2991, 2992, 2995, 3009, 3027, 3045
`\ifx` 4263, 4264, 4269
`\ifzhmappdf` 4281, 4348, 4379, 4410, 4477
`\ignorespaces` 2766
`\immediate` 4272
indent 28, 201
`\indexname` 2230
indexname 12
`\input` 4534, 4599
`\insertcontinuationtext` 2239, 2248, 2255

`\insertpart` 3013, 3014
`\insertromanpartnumber` 3004, 3102
`\insertsection` 3031, 3032
`\insertsectionnumber` 3022, 3194
`\insertsubsection` 3049, 3050
`\insertsubsectionnumber` 3040, 3222

int commands:

`\c_five` 3462
`\c_four` 311, 571, 573
`\int_compare:nNnTF`
 564, 2104, 2148, 3278, 3294, 3310, 3401, 3428
`\int_const:Nn` 973
`\int_div_truncate:nn` 79, 571
`\int_eval:n` 1455, 1457, 1458, 1560, 1562, 1563
`\int_from_hex:n` 563
`\int_gincr:N` 1133, 1420
`\int_incr:N` 2784
`\int_mod:nn` 573
`\int_new:N` 59, 1423
`\int_set:Nn` 554, 562, 2101, 2143
`\int_set_eq:NN` 548
`\int_sub:Nn` 567, 2149
`\int_to_Hex:n` 565, 570, 572
`\int_use:N` 487, 554, 886, 1135, 1421, 2788, 4567
`\int_zero:N` 2781
`\c_one` 162, 1448, 1553, 2156, 2363, 4579
`\c_six` 3462
`\c_three` 309, 2148, 3294, 4580
`\c_two` 95, 163, 377, 381, 2149, 2933, 3278, 3310
`\c_zero` 161, 2104, 3402, 3429
`\interlinepenalty`
 2466, 2499, 2513, 2529, 2612, 2630, 2688, 2689, 2727, 2728
iow commands:
`\iow_close:N` 4556
`\iow_indent:n` 4730
`\iow_new:N` 4558
`\iow_now:Nn` 4568
`\iow_open:Nn` 4553
`\itemsep` 3553, 3564, 3592, 3603
`\itshape` 1632, 1633, 1637, 1638

J

`\jfam` 938, 974

K

`\kaishu` 4219
`\kanjifamily` 1620, 1621, 1622, 1625
`\kanjifamilydefault` 1628

kernel internal commands:

`\l_kernel_expl_bool` 70
keys commands:
`\l_keys_choice_tl` 294
`\keys_define:nn` 155, 388, 391, 978,
 1308, 1417, 1494, 1532, 1571, 2000, 2021, 2039, 2073,
 2196, 2222, 2250, 2260, 2263, 2266, 2391, 2795, 3627, 3674
`\keys_if_exist:nnTF` 1569
`\l_keys_key_tl` 141, 143, 145

\keys_set:nn 27,
171, 177, 233, 242, 251, 277, 287, 294, 301, 331, 337,
430, 437, 443, 3093, 3147, 3171, 3202, 3236, 3252, 3265,
3280, 3288, 3296, 3304, 3311, 3312, 3314, 3380, 3382, 3716
\keys_set_known:nn 3359
\keys_set_known:nnN 990

L

\labelformat 3077
\languagealias 3388, 3393, 3414, 3419
\LARGE 3571, 3610
\Large 2470, 3113, 3119, 3181, 3185, 3570, 3609
\large 3206, 3569, 3608
\lastbox 2754
\leavevmode 2418, 2642
left commands:
 \c_left_brace_str 4885
\leftmargin 3550, 3561, 3589, 3600
\leftmargini 3550, 3561, 3589, 3600
\let 1696, 2672, 2679, 3543, 3582, 4256, 4262
\linespread 3618
linespread 10, 181
linestretch 10, 2073
\linewidth 2102, 2110, 2142
\lishu 4221
\listfigurename 2225
listfigurename 11
\listtablename 2226
listtablename 11
\LoadClass 411, 416, 421, 426
\long 4270
\lstlistingname 764, 765
\lstlistlistingname 766, 767
\ltjalchar 731
\ltjdefcharrange 690, 691, 692, 694, 698, 699, 702, 705
LTJFONTUID 1417
\ltjgetparameter 1904, 1934, 1982, 1996
\ltjsetkanjiskip 1921
\ltjsetmathletter 738
\ltjsetparameter 706, 712, 1610
\ltjsetxkanjiskip 1986

lua commands:

 \lua_escape_x:n
 .. 847, 859, 886, 893, 1445, 1460, 1461, 1478, 1479, 1488
 \lua_now_x:n 724,
 821, 847, 851, 856, 861, 883, 892, 960, 1466, 1490, 1736

M

\MAKESPA 4559, 4601
\markboth 2475, 2498
\mathgroup 938
\mcfamily 1637
\mddefault 958, 964, 970, 1296, 1301
mode commands:
 \mode_if_math:TF 932
msg commands:
 \msg_critical:nnn 53, 681, 3643

\msg_error:nn 3694
\msg_error:nnn 19, 22, 38, 41, 672, 685, 688, 2212, 3444, 3526
\msg_error:nnnn 3660, 3689
\msg_fatal:nn 4540
\msg_new:nn 4538
\msg_new:nnn 140, 142, 144, 146, 148, 675,
1032, 1101, 1183, 1851, 2243, 3333, 3644, 3703, 4726, 4803
\msg_new:nnnn 12, 30, 45, 99, 2214, 3446, 3668, 3697
\msg_warning:nn 1178, 1855, 2018, 2029, 2265
\msg_warning:nnn
 169, 175, 203, 213, 228, 237, 246, 275, 285, 299,
329, 335, 343, 348, 354, 1096, 3352, 3631, 3681, 4732, 4802
\msg_warning:nnnn 435, 441, 1028, 1864, 1871, 2166, 2172

N

\newCJKfontfamily 1188
\NewDocumentCommand ... 430, 433, 439, 1062, 1188, 1190,
1198, 1203, 1209, 1215, 1220, 1225, 1227, 1355, 1447,
1552, 1587, 1623, 2056, 2127, 2164, 2170, 2191, 2193,
2362, 3438, 3623, 4216, 4217, 4218, 4219, 4221, 4222, 4224
\newfontfeature 975, 976, 977
\newpage 2543, 2548
\ngostype 44, 1754
\nobreak 2470,
2477, 2520, 2607, 2613, 2616, 2631, 2634, 2742, 3114, 3158
\nobreakspace 2470, 2503
nocap 28
\noexpand 4254, 4256, 4257, 4258
nofonts 28
noindent 28
nopunct 28
\normalem 1834
\normalfont 537, 539, 540, 1212,
1218, 1223, 1253, 1255, 1256, 1279, 2467, 2468, 2500,
2501, 2514, 2515, 2530, 2531, 2601, 2602, 2627, 2628, 2792
\normalsize 3239, 3255, 3268, 3537, 3576, 3614, 3622
nospace 28
nozhmap 28
\null 2446, 2546
\numberline 2575, 2576, 2697, 2698, 2713, 2714, 2927

O

one commands:

 \c_minus_one 152, 548, 3401, 3428
 \onecolumn 2441
or commands:
 \or: 405, 3532

P

\pagestyle 3379
\par 2407, 2419, 2470, 2503, 2506, 2517,
2532, 2607, 2613, 2631, 2643, 2742, 3005, 3006, 3013,
3016, 3023, 3024, 3031, 3034, 3041, 3042, 3049, 3052,
3095, 3114, 3125, 3142, 3150, 3158, 3197, 3199, 3230, 3232
paragraph/afterskip 30
paragraph/beforeskip 30
paragraph/numbering 15

- paragraph/runin 20
 \parindent
 2123, 2124, 2146, 2167, 2168, 2173, 2174, 2467, 2468,
 2514, 2515, 2601, 2602, 2627, 2628, 3007, 3025, 3043, 3312
 \parsep 3552, 3553, 3563, 3564, 3591, 3592, 3602, 3603
 \part 2416
 part/before skip 30
 part/fixbefore skip 30
 part/numbering 15
 part/pagestyle 21, 2372
 \partname 2470, 2503, 3004, 3097
 \PassOptionsToClass 410, 415, 420, 425
 \PassOptionsToPackage 505, 1797, 2176, 4734
 \pdfmapline 4282,
 4283, 4284, 4285, 4286, 4287, 4288, 4290, 4291, 4292,
 4293, 4294, 4295, 4296, 4298, 4299, 4300, 4301, 4302,
 4303, 4305, 4306, 4307, 4308, 4309, 4310, 4411, 4412,
 4413, 4414, 4415, 4416, 4417, 4419, 4420, 4421, 4422,
 4423, 4424, 4425, 4427, 4428, 4429, 4430, 4431, 4432,
 4434, 4435, 4436, 4437, 4438, 4439, 4478, 4479, 4480,
 4481, 4482, 4483, 4485, 4486, 4487, 4488, 4489, 4490,
 4492, 4493, 4494, 4495, 4496, 4498, 4499, 4500, 4501, 4502
 \pdfoutput 4265
 \pdfstringdefDisableCommands 3626
 pdftex commands:
 \pdftex_pdffontattr:D 486
 \pdftex_pdflastobj:D 487
 \pdftex_pdfobj:D 483
 \phantomsection 2396
 prg commands:
 \prg_do_nothing:
 942, 1394, 1486, 1880, 1954, 2005, 2777, 4926
 \prg_new_conditional:Npnn 845, 1927
 \prg_new_protected_conditional:Npnn 1075, 4896
 \prg_return_false: ... 848, 1085, 1931, 1935, 1939, 4906
 \prg_return_true: 848, 1078, 1083, 1931, 1935, 1939, 4911
 \primitive 4635
 \ProcessKeysOptions 400
 \proofname 2217, 2232
 proofname 12
 prop commands:
 \prop_get:NnN 1170
 \prop_get:NnNTF
 ... 1077, 1088, 1129, 1167, 1387, 1435, 3442, 3497, 3524
 \prop_gpop:NnNTF 1017, 1021, 1522
 \prop_gput:Nnn ... 994, 995, 1043, 1157, 1510, 1546, 3459
 \prop_gremove:Nn 1026
 \prop_if_empty:NTF 1091, 1260
 \prop_map_break:n 1274
 \prop_map_inline:Nn 1272
 \prop_new:N 1012, 1013, 1014, 1162, 1530, 1551, 3453
 \protect .. 2575, 2576, 2588, 2589, 2697, 2698, 2713, 2714,
 2895, 2897, 2901, 2903, 2907, 2909, 2917, 2922, 4857, 4858
 \ProvidesFile 4270, 4278, 4345, 4376, 4407, 4474
 \providetranslation
 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, 4715, 4716, 4717, 4718, 4719
 ptex commands:
 \ptex_kanjiskip:D 1907, 1924, 1938
 \ptex_xkanjiskip:D 1983, 1987, 1997
 punct 9, 28, 280, 2021
 \punctstyle 582, 2026

 Q
 \quad . 2668, 2960, 2961, 2964, 2965, 2971, 2975, 2983, 2986,
 2991, 2992, 2995, 3120, 3165, 3175, 3207, 3240, 3256, 3269
 quark commands:
 \q_mark 4900, 4903
 \q_stop 120, 125, 897,
 900, 1352, 1356, 1583, 1588, 4865, 4870, 4879, 4900, 4903
 \quotation 3636

 R
 \raggedright 2467, 2514, 2601, 2627, 3112, 3156
 \refname 2233, 2238, 2245, 2254, 2262, 2263
 refname 12
 \refstepcounter ... 2456, 2488, 2571, 2675, 3060, 3082, 3083
 \relax 2468, 2486, 2502, 2515, 2602, 2628, 2644,
 2676, 2682, 2687, 2704, 2721, 2726, 2733, 2734, 2739,
 2758, 2760, 2982, 2985, 3006, 3007, 3016, 3024, 3025,
 3034, 3042, 3043, 3052, 4228, 4232, 4235, 4263, 4264, 4269
 \renewcommand 2417, 2432, 2556
 \RenewDocumentCommand 3083
 \RequirePackage 27, 56, 57, 350, 361, 506, 509, 510, 511, 664,
 683, 686, 720, 763, 1806, 1815, 1833, 1839, 2177, 3433, 4779
 \RequirePackageWithOptions 4735
 resetalternatefont 26, 1494
 reverse commands:
 \reverse_if:N 585
 \rmdefault 1710
 \rmfamily 525, 528, 1241, 1244, 1620

 S
 scan commands:
 \scan_stop:
 ... 65, 120, 819, 938, 1487, 1488, 2094, 2102, 4563, 4745
 scheme 9, 315
 \scriptsize 3567, 3606
 \secdef 2429, 2450, 2566
 section/afterskip 30
 section/before skip 30
 section/numbering 15
 section/runin 20
 \sectionname 3022, 3191
 \selectfont
 1069, 1513, 1526, 1626, 1678, 1956, 1962, 1975, 2058, 3443

seq commands:

<code>\seq_clear:N</code>	988
<code>\seq_concat:NNN</code>	1328
<code>\seq_gput_left:Nn</code>	2281, 2282
<code>\seq_gput_right:Nn</code>	1095, 3465
<code>\seq_gset_eq:NN</code>	2280
<code>\seq_gset_from_clist:Nn</code>	2275, 2285, 4543
<code>\seq_if_empty:NTF</code>	1343
<code>\seq_if_in:NnTF</code>	463, 1093
<code>\seq_map_function:NN</code>	2841
<code>\seq_map_inline:Nn</code>	1351, 2383, 2782, 3071, 4566
<code>\seq_new:N</code> ..	456, 1100, 1336, 1337, 2274, 2278, 3454, 4542
<code>\seq_put_right:Nn</code>	1599
<code>\seq_set_filter:NNn</code>	1326
<code>\seq_set_from_clist:Nn</code>	457
<code>\seq_set_split:Nnn</code>	1325
<code>\seq_use:Nnnn</code>	3451
<code>\setbox</code>	2754
<code>\setCJKfamilyfont</code>	1114, 1188, 3860, 3866, 3867, 3868, 3869, 3870, 3871, 3914, 3915, 3923, 3924, 3926, 3927, 3928, 3929, 3931, 3953, 3955, 3957, 3959, 3998, 3999, 4000, 4001, 4025, 4030, 4035, 4037, 4075, 4077, 4079, 4080, 4094, 4095, 4096, 4097, 4110, 4111, 4112, 4113, 4114, 4115, 4156, 4157, 4158, 4159, 4160, 4161, 4174, 4175, 4176, 4209, 4210, 4211
<code>\setCJKmainfont</code>	1110, 1209, 3852, 3910, 3918, 3945, 3991, 4013, 4070, 4091, 4106, 4151, 4170, 4205
<code>\setCJKmathfont</code>	1209
<code>\setCJKmonofont</code>	1112, 1209, 3865, 3913, 3922, 3952, 3997, 4024, 4074, 4093, 4109, 4155, 4173, 4208
<code>\setCJKromanfont</code>	1214, 1234
<code>\setCJKsansfont</code>	1111, 1209, 3855, 3859, 3912, 3920, 3951, 3996, 4019, 4073, 4092, 4108, 4153, 4172, 4207
<code>\SetSymbolFont</code>	971, 1300, 1304
<code>\sfdefault</code>	1711
<code>\sffamily</code>	529, 532, 1245, 1248, 1621

skip commands:

<code>\skip_horizontal:N</code>	1914, 1918
<code>\skip_if_eq:nnTF</code>	1934, 1938, 1981
<code>\skip_new:N</code>	1926, 1994, 2411
<code>\skip_set:Nn</code>
...	1904, 1907, 1985, 1995, 2095, 2106, 2151, 2159, 2413
<code>\skip_set_eq:NN</code>	1924, 1987
<code>\skip_sub:Nn</code>	2402, 2409
<code>\skip_zero:N</code>	2115
<code>\c_zero_skip</code>	2354, 2355
<code>\small</code>	3545, 3584
<code>\songti</code>	4216
<code>\space</code>	2572, 2607, 3097, 3154, 3191, 3221, 3320
<code>space</code>	10, 28, 290, 2000
<code>\special</code>	1658, 1659, 1662, 1663, 1813, 1828, 4313, 4314, 4315, 4316, 4317, 4318, 4319, 4321, 4322, 4323, 4324, 4325, 4326, 4327, 4329, 4330, 4331, 4332, 4333, 4334, 4336, 4337, 4338, 4339, 4340, 4341, 4352, 4353, 4354, 4355, 4356, 4358, 4359, 4360, 4361, 4362, 4364, 4365, 4366, 4367, 4369, 4370,

4371, 4372, 4383, 4384, 4385, 4386, 4387, 4389, 4390, 4391, 4392, 4393, 4395, 4396, 4397, 4398, 4400, 4401, 4402, 4403, 4442, 4443, 4444, 4445, 4446, 4447, 4448, 4450, 4451, 4452, 4453, 4454, 4455, 4456, 4458, 4459, 4460, 4461, 4462, 4463, 4465, 4466, 4467, 4468, 4469, 4470, 4505, 4506, 4507, 4508, 4509, 4511, 4512, 4513, 4514, 4515, 4517, 4518, 4519, 4520, 4522, 4523, 4524, 4525	
<code>\SplitArgument</code>	1448, 1553, 2363

str commands:

<code>\c_backslash_str</code>	4884
<code>\c_colon_str</code>	924, 925, 926
<code>\str_case:nnTF</code>	1708, 1754, 4855, 4870
<code>\str_case_x:nnTF</code>	1108, 1119
<code>\str_const:Nn</code>	43, 50, 1779, 1787
<code>\str_if_empty:NTF</code>	4941, 4977, 4986
<code>\str_if_eq:nnTF</code>	508, 586, 946, 1802, 1810, 3384, 3411, 3425, 3683, 3700, 3707
<code>\str_if_eq_x:nnTF</code>	581, 1264, 3680, 4861
<code>\str_lower_case:n</code>	482
<code>\str_new:N</code>	4779, 4890, 4891, 4892
<code>\str_set:Nn</code>	4908, 4909, 4910
<code>\str_upper_case:n</code>	1780
<code>\strutbox</code>	3620
<code>sub3section</code>	9, 307
<code>sub4section</code>	9, 307
<code>subparagraph/afterskip</code>	30
<code>subparagraph/beforeskip</code>	30
<code>subparagraph/numbering</code>	15
<code>subparagraph/runin</code>	20
<code>subsection/afterskip</code>	30
<code>subsection/beforeskip</code>	30
<code>subsection/numbering</code>	15
<code>subsection/runin</code>	20
<code>\subsectionname</code>	3040, 3221
<code>subsubsection/afterskip</code>	30
<code>subsubsection/beforeskip</code>	30
<code>subsubsection/numbering</code>	15
<code>subsubsection/runin</code>	20

sys commands:

<code>\c_sys_engine_str</code>	44
<code>\sys_if_engine_pdftex:TF</code>
...	130, 3847, 3938, 4006, 4085, 4102, 4166
<code>\sys_if_engine_uptex:TF</code>
...	3891, 3979, 4058, 4088, 4135, 4194
<code>\sys_if_engine_xetex:TF</code>	4154, 4536
<code>\sys_if_output_pdf:TF</code>	105, 492, 1805, 3940, 4008

T

<code>\tablename</code>	2228
<code>tablename</code>	12
<code>\TeX</code>	3624

T_EX and L^AT_EX 2_ε commands:

<code>\@italiccorr</code>	1612
<code>\@par</code>	2688, 2727, 3176, 3208, 3241, 3282, 3298
<code>\@afterheading</code>	2482, 2525, 2594, 2747
<code>\@afterindentfalse</code>	2427, 2560, 2564, 2647, 2653

\@afterindenttrue	2425, 2562, 2645, 2651	\@ssect	2665, 2720
\@Alph	3316, 3321	\@startsection	92, 93, 99, 2641, 2787
\@chapapp	2572, 2607, 2970, 2990	\@svsec	2672, 2676, 2679, 2687, 2705, 2706
\@chapter	2566, 2567	\@svsechd	2703, 2733, 2734, 2756
\@chinese	2179, 2188	\@tempdima	2930, 2932
\@clubpenalty	2762	\@tempskipa	2644, 2646, 2647, 2658, 2682, 2683, 2721, 2722, 2739, 2740, 2743, 2758, 2759
\@currentlabel	3062, 3066	\@tempswafalse	2444
\@currentx	4, 4725, 4774	\@tempswatrue	2442
\@currname	4, 4725, 4775	\@topnewpage	2591
\@dblarg	2666	\@topnum	2559
\@defaultunits	35	\@xssect	2719, 2737, 2738
\@empty	98, 2672, 2679	\@zhdig	2187
\@endpart	2509, 2535, 2537	\@zhnum	2179, 2186
\@EverySelectfont@AtNextHook	1682, 1683	\abstractname	12
\@EverySelectfont@EveryHook	1681	\addCJKfontfeature	65
\@EverySelectfont@Init	1676, 1959	\AfterEndPreamble	129
\@font@info	1691	\algorithmname	12
\@hangfrom	2687, 2726	\appendixname	12, 22, 22, 22
\@ifclassloaded	3329	\AtBeginDocument	35, 45
\@ifpackagelater	37, 40, 671, 684, 687, 1840	\AtBeginDvi	42
\@ifpackageloaded	18, 21, 680, 1686, 1790, 2880, 2945, 3079, 4756	\AtBeginShipoutFirst	42
\@ifpackagewith	2825, 2834, 2885	\AtEndOfClass	36
\@ifstar	2664	\AtEndOfPackage	36
\@listI	3543, 3582	\AtEndOfPackageFile*	129
\@listi	3543, 3550, 3561, 3582, 3589, 3600	\AtEndPreamble	129
\@M	2466, 2499, 2513, 2529, 2612, 2630, 2688, 2689, 2727, 2728, 2755	\baselineskip	10, 11, 76, 79, 79
\@makechapterhead	2591, 2593, 2596	\bfseries	47
\@makeschapterhead	2622	\bibname	12
\@minus	3177, 3209, 3242, 3257, 3270, 3539, 3541, 3547, 3549, 3551, 3552, 3558, 3560, 3562, 3563, 3578, 3580, 3586, 3588, 3590, 3591, 3597, 3599, 3601, 3602	\c@chapter	3321
\@namedef	2895, 2897, 2901, 2903, 2907, 2909, 2917, 2922	\c@secnumdepth	2454, 2469, 2486, 2502, 2568, 2604, 2671, 2695, 2711
\@nameuse	2895, 2897, 2907, 2909, 2917	\c@section	3316
\@ne	2878	\caption	29
\@nil	872	\catcode	34, 34, 44, 44, 120, 130
\@nobreakfalse	2749	\ccwd	10, 10, 11, 24, 24, 29, 29, 35, 74, 75, 76, 78, 79, 79, 80
\@noskipsecfalse	2753	\chapter	9, 15, 21, 22, 22, 85, 103
\@noskipsectrue	2750	\chaptermark	13
\@onlypreamble	35, 453, 519, 1230, 1231, 1232, 1233, 1234, 1235, 1653, 1667, 1675, 3667	\chaptername	15
\@part	2429, 2450, 2452	\char	49
\@pkgextension	6, 7, 8, 9, 20, 23, 682, 4774	\CheckCommand	76
\@plus	3126, 3127, 3177, 3178, 3209, 3210, 3242, 3243, 3257, 3270, 3283, 3299, 3539, 3540, 3541, 3547, 3548, 3549, 3551, 3552, 3558, 3559, 3560, 3562, 3563, 3578, 3579, 3580, 3586, 3587, 3588, 3590, 3591, 3597, 3598, 3599, 3601, 3602	\Chinese	29, 29
\@popfilename	130, 4772	\chinese	24, 24, 29, 29, 81
\@schapter	2566	\CJK@@ignorespaces	46, 599
\@secCNTformat	2667, 2676	\CJK@encoding	55, 57, 64, 778, 782, 866, 953, 1295, 1298, 1300, 1304, 1445, 1460, 1461, 1478, 1479
\@secpENALTY	2658, 2659	\CJK@envStart	45, 45, 45, 46, 593
\@sect	2666, 2670	\CJK@family	50, 770, 783, 830, 866, 1066, 1512, 1522, 1524, 1975, 1976
\@setfontsize	3529	\CJK@ignorespaces	46, 46, 604, 608
\@spart	2429, 2450, 2511	\CJK@input	45, 542
		\CJK@loadBinding	44, 45, 589
		\CJK@loadEncoding	44
		\CJK@makeActive	45, 588
		\CJK@plane	45, 76, 471, 472, 482, 1964
		\CJK@surr	45, 45, 558

<code>\CJK@upperReset</code>	46	<code>\CTEX@prechapter</code>	2817
<code>\CJKfamily</code>	56	<code>\CTEX@presection</code>	2811
<code>\CJKfamilydefault</code>	45, 46, 60, 60, 68, 70, 70, 70, 70, 70	<code>\CTEX@runin</code>	2684, 2723, 2741, 2774, 2780
<code>\CJKglue</code>	79, 79, 108	<code>\CTEX@save@appendix</code>	2806, 2809
<code>\CJKhook</code>	46, 46	<code>\CTEX@save@refstepcounter</code>	3082, 3085
<code>\CJKrmdefault</code>	44	<code>\CTEX@section@aftername</code>	3028
<code>\CJKsymbol</code>	76	<code>\CTEX@section@afterskip</code>	3034
<code>\CJKunderdotbasesep</code>	29	<code>\CTEX@section@aftertitle</code>	3032
<code>\CJKunderline</code>	29	<code>\CTEX@section@beforeskip</code>	3024
<code>\clearpage</code>	46	<code>\CTEX@section@format</code>	3026
<code>\contentsname</code>	11, 11	<code>\CTEX@section@indent</code>	3025
<code>\CS</code>	51, 51	<code>\CTEX@section@numbering</code>	2814, 2961, 2975, 2983, 2995, 3027
<code>\csname</code>	108	<code>\CTEX@section@titleformat</code>	3032
<code>\CTEX@afterindent</code>	2650, 2772, 2778	<code>\CTEX@sectionname</code>	3028
<code>\CTEX@aftertitle</code>	2730, 2735, 2771, 2777	<code>\CTEX@selectfont@hook</code>	1949, 1964, 1967
<code>\CTEX@appendix@number</code>	2800, 2812, 2818	<code>\CTEX@setcurrentlabel@n</code>	3060, 3086
<code>\CTEX@appendix@numbering</code>	2801, 2814, 2820	<code>\CTEX@setheadingskip</code>	2412, 2421, 2447, 2479, 2522, 2540, 2598, 2618, 2624, 2636, 2660, 2744
<code>\CTEX@chapter@afterindent</code>	2561	<code>\CTEX@subsection@aftername</code>	3046
<code>\CTEX@chapter@aftername</code>	2608	<code>\CTEX@subsection@afterskip</code>	3052
<code>\CTEX@chapter@afterskip</code>	2618, 2636	<code>\CTEX@subsection@aftertitle</code>	3050
<code>\CTEX@chapter@aftertitle</code>	2615, 2633	<code>\CTEX@subsection@beforeskip</code>	3042
<code>\CTEX@chapter@beforeskip</code>	2598, 2624	<code>\CTEX@subsection@format</code>	3044
<code>\CTEX@chapter@fixskip</code>	2599, 2619, 2625, 2637	<code>\CTEX@subsection@indent</code>	3043
<code>\CTEX@chapter@format</code>	2603, 2629	<code>\CTEX@subsection@numbering</code>	2965, 2986, 3045
<code>\CTEX@chapter@indent</code>	2602, 2628	<code>\CTEX@subsection@titleformat</code>	3050
<code>\CTEX@chapter@numbering</code>	2570, 2606, 2820, 2971, 2991, 2992	<code>\CTEX@subsectionname</code>	3046
<code>\CTEX@chapter@pagestyle</code>	2377, 2558	<code>\CTEX@thechapter</code>	2818
<code>\CTEX@chapter@titleformat</code>	2614, 2632	<code>\CTEX@thesection</code>	2812
<code>\CTEX@chaptername</code>	2608	<code>\CTEX@titleformat@n</code>	92, 2729, 2735, 2770, 2776
<code>\CTEX@document@left@hook</code>	4742, 4748	<code>\CTEX@titleslabel@clear</code>	2904, 2914
<code>\CTEX@document@right@hook</code>	4744, 4750	<code>\CTEX@titleslabel@set</code>	2898, 2910, 2914
<code>\CTEX@Family@CMap</code>	468, 496, 499	<code>\CTEX@toc@width@n</code>	2927, 2942, 2952
<code>\CTEX@fixheadingskip</code>	2400, 2405, 2422, 2480, 2523, 2541, 2619, 2637, 2661, 2745	<code>\CTEX@todayold</code>	2195, 2200
<code>\CTEX@fixskip</code>	2661, 2745, 2773, 2779	<code>\CTEX@update@sectionformat@n</code>	2649, 2768
<code>\CTEX@fixtopskip</code>	2397, 2448, 2599, 2625	<code>\CTEX@verbatim@font@hook</code>	1605, 1609
<code>\CTEX@headingskip</code>	2414, 2423, 2449, 2481, 2524, 2542, 2600, 2620, 2626, 2638, 2662, 2746	<code>\ctex@zhmap@endinput</code>	4530
<code>\CTEX@makeheadinganchor</code>	2394, 2460, 2492, 2578, 2678	<code>\CTEX@digits</code>	24, 24, 24, 24, 24
<code>\CTEX@part@afterindent</code>	2424	<code>\CTEX@number</code>	24, 24, 24, 24, 24
<code>\CTEX@part@aftername</code>	2471, 2504, 3010	<code>\CTEX@options</code>	29, 29
<code>\CTEX@part@afterskip</code>	2479, 2522, 2540, 3016	<code>\ctexset</code>	5, 5, 5, 5, 5, 5, 7, 11, 14, 14, 25, 26, 29, 29, 110
<code>\CTEX@part@aftertitle</code>	2476, 2508, 2519, 2534, 3014	<code>\CTEX@setup</code>	29
<code>\CTEX@part@beforeskip</code>	2421, 2447, 3006	<code>\CTEX@thechapter</code>	12, 13, 13, 16
<code>\CTEX@part@fixskip</code>	2422, 2448, 2480, 2523, 2541	<code>\CTEX@thesection</code>	12, 13, 16, 96
<code>\CTEX@part@format</code>	2468, 2501, 2516, 2531, 3008	<code>\CTEXunderdotbasesep</code>	29
<code>\CTEX@part@indent</code>	2468, 2515, 3007	<code>\CTEXunderline</code>	29
<code>\CTEX@part@numbering</code>	2455, 2469, 2487, 2502, 3009	<code>\curr@fontshape</code>	784, 814, 823, 827, 873
<code>\CTEX@part@pagestyle</code>	2376, 2439	<code>\DeclareAlternateKanjiFont</code>	64
<code>\CTEX@part@titleformat</code>	2474, 2507, 2518, 2533, 3014	<code>\DeclareFontEncoding</code>	54
<code>\CTEX@partname</code>	2471, 2504, 3010	<code>\DeclareFontFamily</code>	43, 51
<code>\CTEX@postappendix</code>	2805, 2813, 2819	<code>\DeclareFontShape</code>	51
<code>\CTEX@postchapter</code>	2819	<code>\DeclareFontShape@</code>	64, 1404
<code>\CTEX@postsection</code>	2813	<code>\DeclareRobustCommand</code>	131
<code>\CTEX@preappendix</code>	2804, 2811, 2817	<code>\define@newfont</code>	51, 806
		<code>\do@subst@correction</code>	51, 52, 805

\em	68, 68, 68	\ltj@set@stackfont	924, 925, 926
\enc@update	1679, 1684	\ltj@allalchar	725
\escapechar	54	\ltj@curjfont	51, 76
\EverySelectfont	76	\ltj@pickup@altfont@auxy	867
\external@font	816, 817, 819	\ltj@pickup@altfont@copy	880
\extract@font	51, 804	\ltj@setpar@global	923
\f@baselineskip	2094	\ltj@tempcntc	53, 886
\f@encoding	782, 822, 935	\ltjalchar	49
\f@family	50, 783, 822, 830	\m@ne	
\f@series	778, 866		2454, 2469, 2568, 2604, 3727, 3728, 3729, 3730, 3731, 3732
\f@shape	778, 866	\markboth	13
\f@size	778, 784, 861, 869, 872, 873, 1900, 1946, 2933	\markright	13
\fangsong	8, 8	\math@bgroup	934
\figurename	11	\math@egroup	939
\font	51, 51	\math@fonts	921
\font@name	50, 51, 787, 794, 795, 798, 801, 819, 820, 837, 915, 916, 917, 918, 919, 926, 1691	\mathbb	108
\fontcharwd	75, 76	\maxdimen	11, 79
\footnotesep	108, 108, 108	\meaning	131, 132, 132, 133
\footnotesize	108	\newcommand	131
\get@external@font	813	\newrobustcmd	131
\getanddefine@fonts	904	\newtitlemark	95, 95
\globaljfont	51	\nfss@catcodes	55
\hbox	86	\ngostype	71
\heiti	8	\normalsize	8, 8, 104
\hrule	86	\nouppercase	45
\Hy@driver	1810	\p@	1900, 1946, 2503, 2588, 2589, 2597, 2607, 2617, 2623, 2635, 2933, 3125, 3151, 3152, 3158, 3539, 3540, 3541, 3547, 3548, 3549, 3551, 3552, 3558, 3559, 3560, 3562, 3563, 3578, 3579, 3580, 3586, 3587, 3588, 3590, 3591, 3597, 3598, 3599, 3601, 3602
\Hy@unicodetrue	72	\pagenumbering	24
\HyPsd@ConvertToUnicode	72	\pagestyle	9
\HyPsd@LoadUnicode	73	\paragraph	9, 9, 9, 9, 9, 20, 34, 34
\HyPsd@pdfencoding	73, 73	\parindent	10, 29, 78, 80, 108
\HyPsd@ToBigChars	73	\parskip	86, 86, 86
\if@mainmatter	2569, 2605, 2991, 3347, 3348	\part	14, 20, 21, 85
\if@nobreak	2655	\partname	15
\if@noskipsec	2418, 2642, 2752	\partpage	14, 97
\if@openright	2433, 2545, 2556	\PassOptionsToClass	41
\if@tempswa	2551	\PassOptionsToPackage	72
\if@twocolumn	2440, 2590	\pdfmapline	114, 117
\if@twoside	2544, 2962, 2972	\pdfstringdef	73
\iffalse	53	\pickup@font	50
\ifin@	53, 894	\pickup@jfont	810
\iftrue	53	\ProcessKeysOptions	27
\indexname	12	\proofname	12, 82
\insertcontinuationtext	12	\protected	95
\itdefault	52	\protected@edef	2192, 2194, 2676, 3062
\itshape	47	\ProvidesFile	121
\kaishu	8	\ps@fancy	2978
\kanjiskip	76	\ps@headings	2958
\labelformat	98, 98	\refname	12, 12
\leftmark	13	\refstepcounter	98
\linewidth	79, 79	\relax	45, 79
\lishu	8, 8	\reset@font	539, 1255
\listfigurename	11	\rightmark	13
\listtablename	11		
\ltj@does@alt@set	52		
\ltj@getjfontnumber	53, 882		

<code>\rmfamily</code>	48	<code>\zhdigits</code>	24
<code>\roman</code>	24	<code>\zhnum</code>	81
<code>\scantokens</code>	131	<code>\zhnumber</code>	24, 24, 24, 24
<code>\scriptfont@name</code>	918, 925	<code>\zihao</code>	23, 23
<code>\section</code>	9, 9, 14, 15, 20, 20, 21, 21, 22, 22, 30, 83	<code>\ziju</code>	24
<code>\section*</code>	15	<code>\zw</code>	76
<code>\sectionmark</code>	13, 13	tex commands:	
<code>\sectionname</code>	15	<code>\tex_def:D</code>	4934
<code>\sectionpage</code>	14, 97	<code>\tex_divide:D</code>	79, 2103, 2144
<code>\selectfont</code>	53, 69, 76, 76, 76, 76, 79	<code>\tex_edef:D</code>	4988
<code>\setCJKfamilyfont</code>	25, 25, 25	<code>\tex_endlinechar:D</code>	548, 554
<code>\setCJKmainfont</code>	25	<code>\tex_font:D</code>	486, 4563
<code>\sf@size</code>	917	<code>\tex_iftrue:D</code>	3348
<code>\sffamily</code>	48	<code>\tex_ignorespaces:D</code>	1063, 1201, 1206, 2011, 2128, 3439, 4745
<code>\size@update</code>	79, 79, 2058, 2072	<code>\tex_immediate:D</code>	483
<code>\sldefault</code>	52	<code>\tex_pagegoal:D</code>	2401
<code>\songti</code>	7	<code>\tex_parskip:D</code>	2409
<code>\special</code>	7, 72	<code>\tex_prevdepth:D</code>	2408
<code>\split@name</code>	872	<code>\tex_topskip:D</code>	2402
<code>\SplitArgument</code>	85	<code>\the</code>	4232, 4235, 4236, 4255, 4259
<code>\ssf@size</code>	919	<code>\thechapter</code>	2572, 2575, 2607, 2970, 2990, 3155
<code>\strutbox</code>	108	<code>\theparagraph</code>	3254
<code>\subparagraph</code>	9, 9, 9, 9, 20, 34	<code>\thepart</code>	2457, 2470, 2489, 2503, 3099
<code>\subsectionname</code>	15	<code>\thesection</code>	2960, 2974, 2982, 2994, 3174
<code>\subsectionpage</code>	14, 97	<code>\thesubparagraph</code>	3267
<code>\subst@correction</code>	833, 840	<code>\thesubsection</code>	2964, 2985, 3205
<code>\tablename</code>	12	<code>\thesubsubsection</code>	3238
<code>\textfont@name</code>	916, 924	<code>\thispagestyle</code>	2438, 2439, 2547, 2557, 2558
<code>\tf@size</code>	915	<code>\tiny</code>	3568, 3607
<code>\thechapter</code>	13, 16, 22	tl commands:	
<code>\thesection</code>	13, 15, 16, 22, 96	<code>\c_space_tl</code>	487
<code>\titleformat</code>	93	<code>\tl_clear:N</code>	193, 366, 389, 1544, 1548, 1836, 1847, 1976, 2049, 2369, 2381, 3494, 4942
<code>\titlespacing</code>	93	<code>\tl_const:Nn</code>	3, 409, 414, 419, 424, 521, 659, 682, 953, 968, 1237, 1238, 1239, 1240, 1294, 2068, 2292, 2298, 3330, 3349, 3351
<code>\today</code>	11, 11	<code>\tl_gput_right:Nn</code>	464, 4739, 4741, 4763
<code>\trans@languagepath</code>	3396, 3422	<code>\tl_gset:Nn</code>	565, 568, 915, 917, 919, 1268, 1706, 1734, 1752, 1777, 1784, 1785, 3685
<code>\ttfamily</code>	48	<code>\tl_gset_eq:NN</code>	1045, 1729
<code>\ttl@extract</code>	93	<code>\tl_gset_rescan:Nnn</code>	1275
<code>\ttl@setsubmark</code>	94, 2900, 2906	<code>\tl_head:N</code>	3498
<code>\ttl@settopmark</code>	94, 2894, 2925	<code>\tl_if_blank:nTF</code>	1361, 1457, 1458, 1562, 1563, 1577, 1595
<code>\ttlh@hang</code>	2859, 2869	<code>\tl_if_blank_p:n</code>	1327
<code>\ttlh@runin</code>	2858, 2868	<code>\tl_if_empty:N</code>	773, 2121, 2304, 3076, 3648
<code>\tw@</code>	1690	<code>\tl_if_empty:nTF</code>	1647, 1660, 4905
<code>\uppercase</code>	45, 45	<code>\tl_if_eq:NNTF</code>	1702, 2061, 3409
<code>\use@mathgroup</code>	751, 930	<code>\tl_if_exist:N</code>	84, 520, 522, 523, 524, 1236, 1238, 1239, 1240, 2217, 2248, 2259, 2262, 3342, 3432, 4762
<code>\usepackage</code>	28, 29	<code>\tl_if_in:NnTF</code>	4969
<code>\verbatim@font</code>	1603, 1607	<code>\tl_map_inline:Nn</code>	1473
<code>\wrong@fontshape</code>	50	<code>\tl_map_inline:nn</code>	3495
<code>\x@protect</code>	4858	<code>\tl_new:N</code>	58, 128, 132, 522, 523, 524, 775, 776, 864, 950, 1005, 1031, 1051, 1052, 1073, 1409,
<code>\xdef</code>	95		
<code>\XeTeXglyphbounds</code>	125		
<code>\xkanjiskip</code>	76, 76, 76, 76		
<code>\yahei</code>	8, 8		
<code>\youyuan</code>	8, 8		
<code>\z@</code>	712, 715, 2467, 2514, 2559, 2601, 2627, 2646, 2683, 2722, 2740, 2754, 2878, 3540, 3548, 3559, 3579, 3587, 3598		
<code>\zhdig</code>	81		

- 1550, 1722, 1990, 2069, 2082, 2219, 2290, 2291, 2804,
2805, 3838, 4752, 4753, 4762, 4823, 4824, 4945, 4946, 4947
- `\tl_put_left:Nn` 1620, 1621, 1622, 1959, 4772
- `\tl_put_right:Nn` 373, 921, 1557,
1590, 2320, 2342, 2344, 2346, 2348, 2350, 2352, 3498, 3499
- `\tl_replace_all:Nnn` 1324, 1365
- `\tl_rescan:nn` 133, 4987, 4996
- `\tl_set:Nn` 114, 115, 129, 188, 221, 222, 482, 514,
515, 516, 561, 764, 766, 777, 816, 865, 947, 948, 989,
1056, 1068, 1192, 1323, 1362, 1364, 1405, 1578, 1582,
1616, 1617, 1618, 1619, 1628, 1724, 1975, 1991, 2025,
2034, 2045, 2083, 2131, 2220, 2367, 2370, 2374, 2855,
2874, 3408, 3626, 3839, 3843, 3885, 3886, 3887, 4817, 4818
- `\tl_set_eq:NN`
782, 783, 830, 916, 918, 962, 1057, 1082, 1512, 1524, 2070
- `\tl_set_rescan:Nnn` 4943
- `\tl_tail:N` 2860
- `\tl_tail:n` 2870
- `\tl_to_str:n` 125, 126, 4886, 4915, 4956, 4957
- `\tl_trim_spaces:n` 1541
- `\today` 2195, 2200, 2203, 2208
- `today` 11, 2195
- token commands:
- `\token_get_replacement_spec:N` 4843, 4865
- `\token_to_meaning:N` 4900
- `\token_to_str:N`
.. 54, 897, 906, 915, 917, 919, 1110, 1111, 1112, 1114,
1121, 1122, 1123, 1185, 2245, 2246, 3447, 4570, 4730, 4802
- `\toks` 4232, 4233, 4235, 4236, 4255, 4259
- `\topsep` 3551, 3562, 3590, 3601
- `\tracingfonts` 1690
- `\TrimSpaces` 433
- `\ttdefault` 1712
- `\ttfamily` 533, 536, 1249, 1252, 1622
- two commands:
- `\c_two_hundred_fifty_six` 564, 567
- `\twocolumn` 2552
- `\typeout` 2572, 2573
- U**
- um commands:
- `\um_input_math_symbol_table:` 744, 748
- `\um_sym:nnn` 747
- um internal commands:
- `__um_input_math_symbol_table:` 756
- `__um_switchto_literal:` 753
- `__um_sym:nnn` 755
- `\undefined` 1696, 4257
- `\unless` 2684, 2723, 2741
- `\unskip` 2757
- `\updefault` ... 958, 964, 966, 970, 972, 1296, 1298, 1301, 1305
- `\upshape` 1637, 1640
- use commands:
- `\use:N` 784, 822,
823, 873, 915, 917, 919, 973, 1023, 1525, 4970, 4990, 5004
- `\use:n` 64, 122, 133, 388, 761, 872,
955, 1004, 1140, 1265, 1269, 1367, 1541, 1723, 1878,
2388, 2776, 2938, 3340, 3457, 3657, 4839, 4881, 4932, 4953
- `\use:nn` 4998
- `\use_i:nn` 108, 894, 4783, 4788, 4808, 4813
- `\use_i:nnn` 261
- `\use_ii:nn` 109, 894, 4783, 4788, 4808, 4813
- `\use_ii:nnn` 134, 266
- `\use_iii:nnn` 271
- `\use_none:n` .. 189, 194, 199, 1040, 1268, 1402, 1879, 2181
- `\use_none:nn` 499
- `\usebeamercolor` 3004, 3022, 3040, 3141, 3196, 3229
- `\usebeamerfont` 3004, 3013,
3022, 3031, 3040, 3049, 3140, 3143, 3195, 3198, 3228, 3231
- `\uselanguage` 3387, 3392, 3413, 3418
- `\usepackage` 4730
- `UTF8` 6, 221
- V**
- `\verse` 3636
- `\vfil` 2446, 2539
- `\vskip` 2478, 2481,
2503, 2521, 2524, 2542, 2607, 2617, 2620, 2635, 2638,
2743, 2746, 3005, 3023, 3041, 3125, 3142, 3158, 3197, 3230
- `\vspace` 2449, 2597, 2600, 2623, 2626
- W**
- `winfonts` 28
- `\write` 4272
- X**
- `\x` 4234, 4237,
4238, 4239, 4240, 4241, 4242, 4243, 4244, 4245, 4246,
4247, 4248, 4249, 4250, 4251, 4252, 4253, 4260, 4270, 4274
- `\xdef` 4273
- `\xeCJKsetup` 665, 1843, 1918, 2006, 2009, 2012, 2027
- xetex commands:
- `\xetex_charglyph:D` 4567
- `\xetex_glyphbounds:D` 4591
- Y**
- `\yahei` 4224
- `\youyuan` 4222
- Z**
- `\zhdig` 2187
- `\zhdigits` 24, 2194
- `zhmap` 7, 28, 256
- `\zhnum` 2186
- zhnum commands:
- `\zhnum_counter:n` 2178
- `\zhnumber` 24, 2192
- `\zhnumsetup` 2204, 2209
- `\zhtoday` 2203, 2208
- `\zihao` 23, 3438, 3447
- `zihao` 8, 158
- `\ziju` 24, 2127
- `\zw` 713, 1904, 1947, 1992