

CT_EX 宏集手册

CTEX.ORG

2018/05/02 v2.4.14*

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	25
第 2 节 简明教程	3	8.1 字号与间距	25
2.1 CT _E X 宏集的组成	3	8.2 中文数字转换	26
2.2 CT _E X 宏集的安装和更新	3	8.3 杂项	26
2.3 使用 CT _E X 文档类	4	第 9 节 LuaL ^A T _E X 下的中文支持方式	26
2.4 使用 <code>ctex</code> 宏包	4	9.1 LuaL ^A T _E X 下替代字体的设置	27
第 3 节 宏包选项与 <code>\ctexset</code> 命令	5	第 10 节 CT _E X 宏集的配置文	28
第 4 节 编译方式、编码与中文字库	5	10.1 修改宏包默认选项	28
4.1 编译方式	5	10.2 宏包载入后的配置	29
4.2 中文编码	6	10.3 配置标题中文翻译	29
4.3 中文字库	6	10.4 自定义字体集	29
第 5 节 排版格式设定	8	第 11 节 对旧版本的兼容性	29
5.1 文档默认字号	8	11.1 CT _E X 0.8a 及以前的版本	29
5.2 章节标题风格	9	11.2 CT _E X 0.9–CT _E X 1.0d	30
5.3 排版方案选项	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	31
第 6 节 文档汉化	11	11.4 CT _E X 2.2 之前的版本	32
6.1 日期汉化	11	11.5 CT _E X 2.4.1 和 2.4.2	32
6.2 文档标题汉化	11	第 12 节 宏集依赖情况与手工安装方法	32
6.3 页面格式设置与汉化	13	第 13 节 开发人员	33
第 7 节 章节标题格式设置	14	第 14 节 代码实现	35
		版本历史	139
		代码索引	141

*`ctex-kit` rev. 096f319.

第 1 节 介绍

历史

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

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

2009 年 5 月，我们在 Google Code 建立了 `ctex-kit` 项目¹，对 `ctex` 宏包及相关脚本进行了整合，并加入了对 X_YTeX 引擎的支持。在开发新版本时，考虑到合作开发和调试的方便，我们放弃了 `doc` 和 `DocStrip`，采取了直接编写宏包代码的方式。

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

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 GitHub²。

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

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

CT_EX 宏集的许多实现细节离不开热心朋友们在 bbs.ctex.org 论坛上的讨论，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“CT_EX”。

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

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

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 C_TE_X 宏集组成

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

表 1 C_TE_X 宏集的组成

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

2.2 C_TE_X 宏集的安装和更新

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

T_EX Live 的宏包管理器是 `tlmgr`。用户可以在系统命令行中⁴执行

```
tlmgr gui
```

启动管理器的图形界面 (Windows 用户也可以通过开始菜单的 TeX Live 2015 → TeX Live Manager 打开)。连接上远程仓库之后,搜索 `ctex` 安装即可。`tlmgr` 的图形界面使用 Perl 编写,容易造成系统假死。遇到这种问题的用户,也可以直接在系统命令行执行

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

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

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

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

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

⁴Windows 系统的命令行是 CMD 命令提示符,你可以使用 Win + R 组合键打开“运行”对话框,然后输入 `cmd` 确认打开命令提示符窗口。

⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

2.3 使用 C_TE_X 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 C_TE_X 宏集提供的四个中文文档类。

C_TE_X 宏集提供了四个中文文档类: **ctexart**、**ctexrep**、**ctexbook** 和 **ctexbeamer**, 分别对应 L^AT_EX 的标准文档类 **article**、**report**、**book** 和 **beamer**。使用它们的时候, 需要将涉及到的所有源文件使用 UTF-8 编码保存⁶。

例 1

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

你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译, 也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。
\end{document}
```

以下是使用 **ctexbeamer** 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 \ctexset 命令

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

不过,CT_EX 宏集也提供了一系列选项。用户可以使用这些选项来控制 CT_EX 宏集的行为。具体来说,这些选项里,有的以传统的方式提供,也有的以 $\langle key \rangle = \langle value \rangle$ 的形式提供。对于以键值对形式提供的选项,在下面的说明中使用**粗体**来表示 CT_EX 的默认设置。

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

\ctexset

New: 2014-03-18

\ctexset { \langle 键值列表 \rangle }

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

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

例 4

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

\ctexset 采用 L^AT_EX3 风格的键值设置,支持不同类型的选项与层次化的选项设置,相关示例见 7 节。

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

表 2 CT_EX 宏集的中文支持方式

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

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

⁷L^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 = `<true|false|zhmCJK>`

Updated: 2014-03-08

指定字体映射机制。本选项只在使用 `pdfLATEX/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 = `<none|adobe|fandol|founder|mac|macnew|macold|ubuntu|windows|windowsnew|windowsold|...>`

New: 2014-03-08

指定 C_TE_X 宏集加载的字库。

如果没有指定 `fontset` 的值, C_TE_X 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_TE_X 预定义了以下六种中文字库。

- adobe 使用 Adobe 公司的四款中文字体, 不支持 `pdfLATEX`。
- fandol 使用 Fandol 中文字体, 不支持 `pdfLATEX`。
- founder 使用方正公司的中文字体。
- mac 使用 Mac OS X 系统下的字体, 不支持 `(pdf)LATEX`。
- macnew 使用 El Capitan 或之后的多字重宋体、苹方黑体。
- macold 使用 Yosemite 或之前的华文字库。
- 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_TE_X 预定义的中文字库, 可以设置 `fontset` 为下述值之一。

- none 不配置中文字体, 需要用户自己配置。
- `<name>` 这里 `<name>` 为自定义的名字。C_TE_X 宏集将载入名为 `ctex-fontset-<name>.def` 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录树下合适位置建立一个名为 `ctex-fontset-<name>.def` 的文件, 在这个文件里面自定义中文字体。然后通过使用 `fontset=<name>` 选项来调用它。字体配置文件的具体写法可以参考 C_TE_X 宏集 `fontset` 目录下的字体配置文件。

注意: 如果希望使用 `\ctexset` 在导言区指定字库, 则需要先在宏包/文档类选项中指定 `fontset = none`。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}
```

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 `CTeXFonts.lua`。

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

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

在 `windows` 和 `founder` 字库中,额外定义了 `\lishu` 和 `\youyuan`。

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

在 `windowsnew` 和 `macnew` 字库中,还有 `\yahei`。其中在 `macnew` 字库中,`\yahei` 实际调用苹方黑体。此举乃是为了更好的兼容性而设。

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

在 `macnew` 字库中,还定义了 `\pingfang`。

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

第 5 节 排版格式设定

5.1 文档默认字号

`zihao`

☆

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

New: 2015-05-06

将文章默认字号(`\normalsize`)设置为小四号字或五号字,具体情况见表 4。`false` 禁用本功能。本选项可以用于四个 CT_EX 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 ☆ CT_EX 文档类是在 L_AT_EX 标准文档类之上开发的。因此,除了可以使用 CT_EX 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,CT_EX 文档类的字号选项会被抑制。亦即,在 zihao 选项之后设置 10pt 选项,zihao 选项将不再起作用。

标准文档类的其他选项在 CT_EX 文档类中依旧有效。例如,设置纸张大小和方向的 a4paper 和 landscape,设置单双面的 oneside 和 twoside 等。CT_EX 会将这些选项传给标准文档类¹²。

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: 2015-04-15

选择文章的排版方案,预设有 chinese 和 plain 两种方案。

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

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

¹³使用 CT_EX 文档类,或者使用 ctex 宏包且开启该选项时。

`plain` 不调整默认字号和行距, 不会汉化文档中的标题名字, 也不会将章节标题风格修改为中文样式, 同时不会调整 `\pagestyle`, 并禁用 `autoindent` 选项。事实上, 此时的 $\text{CT}_{\text{E}}\text{X}$ 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 = \langle \text{quanjiao} | \text{banjiao} | \text{kaiming} | \text{CCT} | \text{plain} \rangle`

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;
`plain` 原样(不调整标点间距)。

`space` `space = \langle \text{true} | \text{false} | \text{auto} \rangle`

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 $(\text{pdf})\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译时); 等同于 `auto` 的效果(使用 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 及 $\text{upL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 = \langle \text{数值} \rangle`

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` $\text{CT}_{\text{E}}\text{X}$ 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 = \langle \text{true} | \text{false} | \text{数值} | \text{带单位的数值} \rangle`

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。

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。

¹⁵ $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 将单个换行视作一个空格。

(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。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CTeX 宏包对显示当前日期的 `\today` 命令进行了汉化, 使之以中文的方式显示今天的日期。如编译本文档的日期就是“2018 年 5 月 2 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“2018 年 5 月 2 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇一八年五月二日”。使用全汉字的日期格式。

`old` 效果为“May 2, 2018”。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}
```

CTeX 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。

<hr/> <code>contentsname</code> *	<code>contentsname = <名字></code>
	设置目录标题名 <code>\contentsname</code> 。中文默认为“目录”。
<hr/> <code>listfigurename</code> *	<code>listfigurename = <名字></code>
	设置插图目录标题名 <code>\listfigurename</code> 。中文默认为“插图”。
<hr/> <code>listtablename</code> *	<code>listtablename = <名字></code>
	设置表格目录标题名 <code>\listtablename</code> 。中文默认为“表格”。
<hr/> <code>figurename</code> *	<code>figurename = <名字></code>
	设置图片环境标题名 <code>\figurename</code> 。中文默认为“图”。
<hr/> <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”。
<hr/> <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。
<hr/> <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”。
<hr/> <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”。
<hr/> <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”。
<hr/> <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。
<hr/> <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”。
<hr/> <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”。
<hr/> <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 C_TE_X 的文档类或是用 `ctex` 宏包加载了 `headings` 选项时,会设置整个文档的页面格式(`page style`)为 `headings`,即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

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

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

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

- `\CTEXthechapter`、`\CTEXthesection` 等章节编号(见 7 节)。它们用来代替英文文档类中的 `\thechapter`、`\thesection` 等宏。
- `\leftmark`、`\rightmark`,它们是在使用章节标题命令后,自动设置的宏。它们实际是在与章节标题命令对应的标记命令 `\chaptermark`、`\sectionmark` 中调用 `\markright` 或 `\markboth` 生成的。

有关 L^AT_EX 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{\CTEXifname{\CTEXthesection——}{\}#1}}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

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

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

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

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

例 11

```

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

本例的页眉效果大致如下(有页眉线):

第 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**, **lotskip**, **lofskip**, **indent**, **hang**, **pagestyle**, **break**, **tocline** 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**, **fixskip**, **hang**, **break** 和 **tocline** 这六个格式无效。

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。注意,斜线 / 前后不能有空格或者换行。

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

例如,同时设置 **part** 一级标题的 **pagestyle** 选项, **chapter** 一级标题的 **format** 与 **pagestyle** 选项和 **section** 一级标题的 **name** 与 **number** 选项:

例 12

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

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

numbering = **true**|false
控制是否对章节标题编号。对各级标题的默认值均为 **true**。

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

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

如果没有特别说明，以下将用“...”代表各级章节标题名。

.../name	★
Updated: 2014-03-08	

name = {<前名字>,<后名字>}
name = {<前名字>}

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

例 13

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

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

表 5 name 选项的默认设置

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

`\CTEXifname` `\CTEXifname {⟨有名字时的格式命令⟩} {⟨无名字时的格式命令⟩}`

New: 2016-09-18

`\CTEXifname` 用于根据当前章节的名字的有无设置不同的格式。

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

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

例 15

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

```
.../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 16

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

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

```
.../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 21)。

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

```
.../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 17

```
\ctexset{
  chapter/number = \arabic{chapter},
}
```

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	{}

```
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+ ★

Updated: 2014-03-08

aftername = {<代码>}
aftername+ = {<代码>}

aftername 选项的参数 <代码> 将被插入到章节编号与其后的标题内容之间, 用于控制格式变换。常用于控制章节编号与标题内容之间的距离, 或者控制标题是否另起一行。

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

```
.../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 18

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

上面的代码可以做出类似这样的章标题效果:

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

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

New: 2015-06-19

`aftertitle` 选项的参数 `{代码}` 将被插入到章节标题内容之后。

`aftertitle+` 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
<code>part</code>	<code>\par</code>
<code>chapter</code>	<code>\par</code>
<code>section</code>	<code>\@@par</code>
<code>subsection</code>	<code>\@@par</code>
<code>subsubsection</code>	<code>\@@par</code>
<code>paragraph</code>	<code>{}</code>
<code>(sub3section)</code>	<code>\@@par</code>
<code>(sub4section)</code>	同上
<code>subparagraph</code>	<code>{}</code>
<code>(sub4section)</code>	<code>\@@par</code>

表 12 `runin` 选项的默认设置

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

```
section/runin      ★
subsection/runin   ★
subsubsection/runin ★
paragraph/runin    ★
subparagraph/runin ★
```

New: 2015-06-27

`runin = true|false`

`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 ★
```

New: 2015-06-27

`afterindent = true|false`

`afterindent` 选项用于设置章节标题后首段的缩进。

`book` 和 `report` 类的 `\part` 标题被单独排在一页之上,`afterindent` 选项没有意义。

对于 `\section` 级以下标题,若设置了 `runin` 选项为 `true`,即标题与随后正文排在同一段,`afterindent` 选项也就没有了意义。

表 13 `afterindent` 选项的默认设置

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

```
.../beforeskip ★
```

Updated: 2016-05-10

`beforeskip = {<弹性间距>}`

`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 before skip 选项的默认设置

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

.../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 before skip 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。

chapter/lofskip	★	lofskip = {弹性间距}
chapter/lotskip	★	lotskip = {弹性间距}

New: 2016-10-01

lofskip 选项控制插图目录(.lof)中,章之间的插图标题的距离。

同样,lotskip 选项控制表格目录(.lot)中,章之间的表格标题的距离。

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

.../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 19

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

section/hang	★	hang = true false
subsection/hang	★	
subsubsection/hang	★	
paragraph/hang	★	
subparagraph/hang	★	

New: 2016-10-22

默认情况下,\section 级以下标题具有悬挂缩进的效果(缩进的宽度为名字宽度和 indent 选项设置的宽度之和)。设置 hang 选项为 false 可以取消这一效果。

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

.../break * break = {<格式命令>}

.../break+ * break+= {<格式命令>}

New: 2016-09-19

break 选项用于控制章节标题与之前正文的分隔关系。一般用于设置是否在标题之前分页或者设置行间罚点。

带加号的 `break+` 选项则用于在已有格式之后追加新的格式命令。

`break` 选项的默认值,在 `scheme` 选项的不同取值下相同。

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

例 20

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

表 18 break 选项的默认设置

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

.../tocline * tocline = {<格式定义>}

New: 2016-10-25

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

表 19 tocline 选项的默认设置

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

其中 `\CTEXnumberline` 的意义是若标题 #1 没有名字,则不输出 `\numberline{\CTEXthe#1}` 等编号:
`\CTEXifname{\protect\numberline{\csname CTExthe#1\endcsname}}{}`

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 20 `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 21 `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 21

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

本例的设置效果大致如下:

熟悉 L^AT_EX

第 8 节 实用命令

8.1 字号与间距

`\zihao`

Updated: 2014-03-08

`\zihao {⟨字号⟩}`
用于调整字号大小。其中⟨字号⟩的有效值共有 16 个,如表 22 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 22 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

`\ziju`

Updated: 2014-03-28

`\ziju {⟨中文字符宽度的倍数⟩}`
用于调整相邻汉字之间的间距,即(在正常中文行文中)前一个汉字的右边缘与后一个汉字的左边缘之间的距离。其中参数可以是任意浮点数值;而中文字符宽度指的是实际汉字的宽度,不包含当前字距。
这个命令会影响 `\ccwd` 的值,但不会影响英文字距。

 $\backslash\text{ccwd}$ Updated: 2014-03-27

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

8.2 中文数字转换

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

 $\backslash\text{chinese}$ Updated: 2016-05-01

 $\backslash\text{chinese} \{ \langle \text{counter} \rangle \}$ $\backslash\text{pagenumbering} \{ \text{chinese} \}$

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

 $\backslash\text{zhnumber}$ New: 2014-03-08

 $\backslash\text{zhnumber} \{ \langle \text{number} \rangle \}$

以中文格式输出数字。这里的数字可以是整数、小数和分数。

 $\backslash\text{zhdigits}$ New: 2014-03-08

 $\backslash\text{zhdigits} \{ \langle \text{number} \rangle \}$

将阿拉伯数字转换为中文数字串。

 $\backslash\text{CTEXnumber}$ $\backslash\text{CTEXnumber} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$

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

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

 $\backslash\text{CTEXdigits}$ $\backslash\text{CTEXdigits} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$

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

8.3 杂项

 $\backslash\text{CTeX}$

用于显示 $\text{CT}\text{E}\text{X}$ 标志。

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

在 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 下, $\text{CT}\text{E}\text{X}$ 宏集依赖 $\text{Lua}\text{TeX-j}\text{a}$ 宏包来完成中文支持。该宏包是日本 TEX 社区的北川弘典、前田一贵、八登崇之等人开发的, 设计目的主要是在 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 引擎下实现日本 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 引擎的 (大部分) 功能。它为了兼容 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 的使用习惯, 对 $\text{L}\text{A}\text{T}\text{E}\text{X} 2_{\epsilon}$ 的 NFSS 作了不少修改和扩充。这对于简体中文用户来说不是必要的, 因而 $\text{CT}\text{E}\text{X}$ 禁用了它在 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 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 $\text{x}\text{e}\text{C}\text{J}\text{K}$ 宏包大致相同。

20150420 版以后的 $\text{Lua}\text{TeX-j}\text{a}$ 宏包开始支持竖排, 但 $\text{CT}\text{E}\text{X}$ 暂不支持竖排。

9.1 Lua_{TeX} 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 $\langle family \rangle$ 的时候,同时设置该字体族在字符范围 $\langle character\ range_n \rangle$ 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 $\langle family \rangle$ 在字符范围 $\langle character\ range \rangle$ 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 22

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

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

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

$\langle name \rangle$ 选项可以与 `AlternateFont` 共同使用,但不能与 `CharRange` 一起使用。如果没有给 $\langle name \rangle$ 设置值,则等价于设置 `CharRange= $\langle name \rangle$` ,即只设置 $\langle name \rangle$ 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 23

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

10.2 宏包载入后的配置

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

例 24

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

例 25

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

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

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

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

<hr/> <hr/> hyperref	新版本宏包中总是自动处理对 hyperref 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 hyperref 宏包。 hyperref 选项过时, 因兼容性保留, 功能是在导言区末尾载入 hyperref 宏包。
<hr/> <hr/> fntef	旧版本的 fntef 选项用于统一 CCTfntef 与 CJKfntef 的界面, 新版本 CT_EX 宏集不再支持 CCT , 而是直接载入 CJKfntef 或 xeCJKfntef 宏包并禁用其彩色设置。该选项是过时选项。
<hr/> <hr/> \CTEXunderdot \CTEXunderline \CTEXunderdblline \CTEXunderwave \CTEXsout \CTEXxout \CTEXfilltwosides	在调用 fntef 宏包选项的同时, 旧版本 CT_EX 宏包由于需要支持 CCT 系统, 会将以 <code>\CJK</code> 开头的 <code>\CJKunderline</code> 等宏换名为以 <code>\CTEX</code> 开头的 <code>\CTEXunderline</code> 等宏。此功能在新版本的 CT_EX 宏集中已失去意义。 此外, 在 pdf_TE_X 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。 在新版本中, 上述由 fntef 衍生的相关宏都成为过时命令。
<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	<code>\CTEXsetup[⟨选项⟩]{⟨标题⟩}</code> 相当于设置了 <code>\ctexset{⟨标题⟩ = {⟨选项⟩}}</code> 。过时命令。
<hr/> <hr/> \CTEXoptions	<code>\CTEXoptions[⟨选项⟩]</code> 相当于设置了 <code>\ctexset{⟨选项⟩}</code> 。过时命令。
<hr/> <hr/> \Chinese	<code>\Chinese{⟨counter⟩}</code> 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。
<hr/> <hr/> captiondelimter	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 CT_EX 宏包中失效。 可以使用 caption 宏包的 <code>labelsep</code> 选项来完成同样的功能。

例 26

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

11.3 CT_EX 1.02c 以后的 SVN 开发版

CT_EX 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。

CT_EX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 C_TE_X 2.2 之前的版本

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

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

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

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

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

11.5 C_TE_X 2.4.1 和 2.4.2

part/fixbeforeskip
chapter/fixbeforeskip

这两个选项已经被删除, 相应功能由新的选项 fixskip 提供。

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 pdfL_AT_EX 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项, 其中 zhm-CJK 是可选的。
- xeCJK 宏集, 它还依赖
 - xtemplate 宏包, 它属于 l3packages 宏集。
 - fontspec 宏包。
- environ 宏包, 它还依赖 trimspaces 宏包。
- ➡ 以上是使用 X_YL_AT_EX 编译时的依赖项。

- luatexja 宏包, 它还依赖
 - adobemapping 宏包。
 - lualibs 宏包。
 - luaotfload 宏包。
 - luatexbase 宏包, 它还依赖 ctablestack 宏包。
 - oberdiek 宏集。
 - xkeyval 宏包。
 - etoolbox 宏包。
- fontspec 宏包。
- xunicode-addon 宏包, 属于 xeCJK 宏集, 它还依赖
 - xunicode 宏包, 它还依赖
 - * graphics 宏集。
 - * graphics-cfg 宏包。
 - * graphics-def 宏包。
- ➡ 以上是使用 Lua^AT_EX 编译时的依赖项。
- pxevery sel 宏包, 属于 platex-tools 宏集。
- zhmetrics-uptex 宏包。
- ➡ 以上是使用 up^AT_EX 编译时的依赖项。

出于一些原因, zhmCJK 尚未被收入 T_EX Live 和 MiK_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 . \@currentx } { 9999/99/99 } }
5 <*class>
6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl
7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl
8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl
9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl
10 </class>
11 <*ctex>
12 \msg_new:nnnn { ctex } { subpackage-loaded }
13 { Package~\#1'~can~not~be~loaded~with~`ctex'. }
14 {
15   \#1'~is~actually~a~part~of~`ctex'.\\
16   It~is~not~necessary~to~load~it~separately.
17 }
18 \@ifpackagelated { ctexsize }
19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }
20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }
21 \@ifpackagelated { 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 } { 2017/12/16 } { }
38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
39 <*class>
40 \@ifpackagelater { l3keys2e } { 2015/12/20 } { }
41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
42 </class>
```

\c__ctex_engine_str
\c__ctex_engine_file_str

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

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

```

56 \RequirePackage { ctexhook , ctexpatch }
<|ctexheading> 57 \RequirePackage { fix-cm , everyisel }

```

14.1 内部函数与变量

```

\l__ctex_tmp_tl 临时变量。
\l__ctex_tmp_int
\l__ctex_tmp_box 58 \tl_clear_new:N \l__ctex_tmp_tl
\l__ctex_tmp_dim 59 \int_new:N \l__ctex_tmp_int
60 \box_new:N \l__ctex_tmp_box
<|ctexheading> 61 \dim_new:N \l__ctex_tmp_dim

```

`\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 }

```

`\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} } { } }

```

`\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 }

```

`\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>

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: }

```

`\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 }

```

`\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 }

```

`\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 }

```

`\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 } }

```

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

```

131 \bool_new:N \g__ctex_zhmCJK_bool

```

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

```

132 \tl_new:N \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

```

\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
```

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

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

\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>
```

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 \int_new:N \g__ctex_font_size_flag
153 \int_set:Nn \g__ctex_font_size_flag { -1 }
154 </class>ctex|ctexsize>
```

14.2 宏包选项

```
155 <*class>style>
156 \keys_define:nn { ctex / option }
157 {
158 </class>style>
zihao 159 <*class>ctex|ctexsize>
160     zihao .choice: ,
161     zihao .value_required:n = true ,
162     zihao / 5 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero } ,
163     zihao / -4 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_one } ,
164     zihao / false .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
<ctexsize> 165 }
166 </class>ctex|ctexsize>
167 <*class>ctex>
168     c5size .code:n =
169     {
170         \msg_warning:nnn { ctex } { deprecated-option }
171         { Option~`zihao=5'~ is~ set. }
172         \keys_set:nn { ctex / option } { zihao = 5 }
173     } ,
174     cs4size .code:n =
175     {
176         \msg_warning:nnn { ctex } { deprecated-option }
```

```

177         { Option~ `zihao=-4'~ is~ set. }
178         \keys_set:nn { ctex / option } { zihao = -4 }
179     } ,
180     c5size .value_forbidden:n = true ,
181     cs4size .value_forbidden:n = true ,

```

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

```

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

```

autoindent 自动调整段落的首行缩进功能。

```

185     autoindent .choice: ,
186     autoindent .default:n = { true } ,
187     autoindent / true .code:n =
188     {
189         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
190         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
191     } ,
192     autoindent / false .code:n =
193     {
194         \tl_clear:N \l__ctex_autoindent_tl
195         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
196     } ,
197     autoindent / unknown .code:n =
198     {
199         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
200         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
201     } ,

```

indent 仅为兼容性保留, 已过时。

```

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

```

```

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

```

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

```

226     fontset .tl_gset:N = \g__ctex_fontset_tl ,

```



```

227     nofonts      .code:n =
228     {
229         \msg_warning:nnn { ctex } { deprecated-option }
230         {
231             Option~`fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
232             fontset~ option.
233         }
234         \keys_set:nn { ctex / option } { fontset = none }
235     } ,
236     adobefonts .code:n =
237     {
238         \msg_warning:nnn { ctex } { deprecated-option }
239         {
240             Option~`fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
241             fontset~ option.
242         }
243         \keys_set:nn { ctex / option } { fontset = adobe }
244     } ,
245     winfonts     .code:n =
246     {
247         \msg_warning:nnn { ctex } { deprecated-option }
248         {
249             Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
250             fontset~ option.
251         }
252         \keys_set:nn { ctex / option } { fontset = windows }
253     } ,
254     nofonts      .value_forbidden:n = true ,
255     winfonts     .value_forbidden:n = true ,
256     adobefonts .value_forbidden:n = true ,

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

```

punct 设置标点符号输出格式。

```

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

```

```

288     \keys_set:nn { ctex / option } { punct = plain }
289   } ,
290   nopunct    .value_forbidden:n = true ,

space 291   space .choices:nn =
292     { true , auto , false }
293     {
294       \exp_args:Nx \ctex_at_end:n
295       { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
296     } ,
297   space .default:n = { true } ,
298   nospace .code:n =
299     {
300       \msg_warning:nnn { ctex } { deprecated-option }
301       { Option~ `space=false' ~ is~ set. }
302       \keys_set:nn { ctex / option } { space = false }
303     } ,
304   nospace .value_forbidden:n = true ,

heading 305   heading .bool_set:N = \l__ctex_heading_bool ,

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

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

scheme 316   scheme .tl_set:N = \l__ctex_scheme_tl ,
317 <*ctexheading>
318   scheme .default:n = { plain } ,
319   scheme .initial:n = { plain }
320 }
321 </ctexheading>
322 <!*ctexheading>
323   scheme .default:n = { chinese } ,
324   scheme .initial:n = { chinese } ,
325 </!ctexheading>

326 </class|ctex|ctexheading>
327 <*class|ctex>

328   cap .code:n =
329     {
330       \msg_warning:nnn { ctex } { deprecated-option }
331       { Option~ `scheme = chinese' ~ is~ set. }
332       \keys_set:nn { ctex / option } { scheme = chinese }
333     } ,
334   nocap .code:n =
335     {
336       \msg_warning:nnn { ctex } { deprecated-option }
337       { Option~ `scheme = plain' ~ is~ set. }
338       \keys_set:nn { ctex / option } { scheme = plain }
339     } ,
340   cap .value_forbidden:n = true ,
341   nocap .value_forbidden:n = true ,

```

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 342 fntef .code:n =

```

343     {
344       \msg_warning:nnn { ctex } { deprecated-option }
345       { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
346     } ,
347     fancyhdr .code:n =
348     {
349       \msg_warning:nnn { ctex } { deprecated-option }
350       { `fancyhdr'~ package~ is~ loaded. }
351       \RequirePackage { fancyhdr }
352     } ,
353     hyperref .code:n =
354     {
355       \msg_warning:nnn { ctex } { deprecated-option }
356       { `hyperref'~ package~ will~ be~ loaded. }
357       \ctex_at_end:n
358       {
359         \cs_if_exist:NF \hypersetup
360         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
361       }
362       \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
363     } ,
364   }

365 </class>ctex>
366 <*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

```

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

```

将未知选项传给标准文档类。

```

391 <*class>
392 \keys_define:nn { ctex / option }
393 {
394   unknown .code:n =
395   { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
396 }
397 </class>

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

```

```

400 <*class|style>
401 \ProcessKeysOptions { ctex / option }
402 </class|style>
403 <*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。

```

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

```

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

```

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

```

14.3 特定引擎支持与设置

14.3.1 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。

```

430 <*pdfTeX|upTeX|aptex>
431 \cs_new_protected_nopar:Npn \ctex_set_zhmap:n #1
432 {
433   \AtBeginDvi {#1}
434   \ctex_at_end_package:nn { atbegshi }
435   { \AtBeginShipoutFirst {#1} }
436 }
437 \@onlypreamble \ctex_set_zhmap:n
438 </pdfTeX|upTeX|aptex>
439 <*pdfTeX>

```

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

```

440 \seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
441 { C19 , C10 , C00 , C09 , C40 , C60 }

```

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

```

442 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2

```

```

443 {
444   \cs_if_free:cF { #1 + #2 }
445   {
446     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
447     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
448   }
449 }
450 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
451 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn

```

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

```

452 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
453 {
454   \cs_if_free:NF \CJK@plane
455   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
456 }
457 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
458 {
459   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
460   #1
461 }
462 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
463 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
464 {
465   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
466   \tex_immediate:D \pdfTeX{pdfobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
467   \cs_new_protected_nopar:Npx #1
468   {
469     \exp_not:N \pdfTeX{pdffontattr:D \exp_not:N \tex_font:D
470     { /ToUnicode ~ \int_use:N \pdfTeX{pdflastobj:D \c_space_tl 0 ~ R }
471   }
472 }

```

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

```

473 \group_begin:
474 \char_set_catcode_other:N \#
475 \sys_if_output_pdf:TF
476 {
477   \group_end:
478   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
479   { \CTEX@Family@CMap {#1} {#2} }
480   {
481     \ctex_at_end_package:nn { cmap }
482     { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
483   }
484   { \ctex_patch_failure:N \DeclareFontFamily }
485 }
486 { \group_end: }

```

首先检查选项,决定是否载入 `zhmCJK` 宏包。

```

487 \if_bool:N \g__ctex_zhmCJK_bool
488 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
489 \RequirePackage { zhmCJK }

```

不载入 `zhmCJK` 宏包时直接调用 `CJK` 及相关宏包。

```

490 \else:
491   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
492   { \RequirePackage { CJK } }
493   { \RequirePackage { CJKutf8 } }
494   \RequirePackage { CJKpunct , CJKspace }

```

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

```

495 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4

```

```

496 {
497   \tl_set:Nn \CJKrmdefault {#1}
498   \tl_set:Nn \CJKsfdefault {#2}
499   \tl_set:Nn \CJKttdefault {#3}
500   \ctex_set_zhmap:n { \ctex_zhmap_input:n {#4} }
501 }
502 \@onlypreamble \ctex_load_zhmap:nnnn

```

`\ctex_zhmap_input:n` 载入字体映射文件时,确认 % 和 \ 的 `\catcode`。

```

503 \cs_new_protected_nopar:Npn \ctex_zhmap_input:n #1
504 {
505   \ctex_file_wrapper:nnn
506   {
507     \char_set_catcode_comment:n { 37 } % %
508     \char_set_catcode_escape:n { 92 } % \
509   }
510   { \file_input:n {#1} }
511   {
512     \char_set_catcode:nn { 37 } { \char_value_catcode:n { 37 } }
513     \char_set_catcode:nn { 92 } { \char_value_catcode:n { 92 } }
514   }
515 }

516 \tl_if_exist:NF \CJKfamilydefault
517 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
518 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
519 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
520 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
521 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
522 { \CJKfamily { \CJKrmdefault } }
523 { }
524 { \ctex_patch_failure:N \rmfamily }
525 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
526 { \CJKfamily { \CJKsfdefault } }
527 { }
528 { \ctex_patch_failure:N \sffamily }
529 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
530 { \CJKfamily { \CJKttdefault } }
531 { }
532 { \ctex_patch_failure:N \ttfamily }
533 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
534 { \CJKfamily { \CJKfamilydefault } }
535 { \cs_set_eq:NN \reset@font \normalfont }
536 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```

537 \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。

```

538 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
539 {
540   \ctex_file_wrapper:nnn
541   {
542     \char_set_catcode_other:n { 60 } % <
543     \char_set_catcode_math_superscript:n { 94 } % ^
544     \int_set:Nn \tex_endlinechar:D { -1 }
545   }
546   { \file_input:n {#1} }
547   {
548     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
549     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }

```

```

550      \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
551    }
552  }
553 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

```

\ctex_plane_to_utfxvibe:Nn
\CJK@surr

```

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

```

554 \if_cs_exist:N \CJK@surr
555   \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
556   {
557     \tl_set:Nx \l__ctex_tmp_tl {#2}
558     \int_set:Nn \l__ctex_tmp_int
559     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
560     \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
561     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
562     {
563       \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
564       \tl_gset:Nx #1
565       {
566         \int_to_Hex:n
567         { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
568         \int_to_Hex:n
569         { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
570       }
571     }
572   }
573 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
574 \fi:

```

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

```

575 \AtBeginDocument
576 {
577   \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
578   { \punctstyle { \l__ctex_punct_tl } }
579 }

```

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

```

580 \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。

```

581 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
582   \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
583   {
584     \CJK@makeActive
585     \CJK@loadBinding { standard }
586   }

```

在导言区结束时调用 \CJK@envStart 启用完整的中文功能。

\CJK@envStart 的定义是

```

\def\CJK@envStart#1#2#3{
  \CJK@upperReset
  \ifCJK@lowercase@
    \CJK@lowerReset
  \fi%
  \CJK@makeActive%
}

```

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


```

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

```

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

```

587 \exp_args:Nx \ctex_at_end_preamble:n
588 {
589   \exp_not:N \CJK@envStart
590   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
591   \exp_not:N \CJKtilde
592 }

```

zhmCJK 判断结束。

```
593 \fi:
```

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

```

594 <@@=>
595 \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。

```
596 <@@=ctex>
```

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

```

597 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
598 {
599   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
600   { \cs_set_eq:NN \CJK@ignorespaces #1 }
601   \ctex_set_ignorespaces:
602 }
603 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
604 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }

```

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

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

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

```

606 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
607 {
608   \clist_map_inline:Nn \c__ctex_punct_family_clist
609   {
610     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
611     {

```

```

612         \cs_set_eq:cc
613         { CJKpunct@ ##1 @spaces }
614         { c__ctex_ #1 ##1 _punct_spaces_tl }
615     }
616 }
617 }
618 \clist_const:Nn \c__ctex_punct_family_clist
619 {
620     zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
621     zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
622 }

```

\ctex_punct_map_family:nn

CJK 族 #1 使用族 #2 的边界信息。

```

623 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
624 {
625     \cs_if_free:cF { CJKpunct@ #2 @spaces }
626     { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
627 }

```

\ctex_punct_map_bfseries:nn

CJK 族 #1 的 \bfseries 使用族 #2 的边界信息。

```

628 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
629 {
630     \clist_map_inline:nn {#1}
631     {
632         \ctex_punct_map_series:nnn { ##1 } { b } {#2}
633         \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
634     }
635 }
636 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
637 {
638     \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
639     \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
640     \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
641     \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
642     \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
643     \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
644 }

```

\ctex_punct_map_itshape:nn

CJK 族 #1 的 \itshape 使用族 #2 的边界信息。

```

645 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
646 {
647     \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
648     \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
649     \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
650     \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
651     \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
652     \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
653 }

```

\ctex_punct_space:nn
\ctexspadef

定义标点的边界信息。

```

654 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
655 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
656 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

657 \ctex_file_input:n { ctexspa.def }
658 </pdfTeX>

```

14.3.2 ctex-engine-xetex.def

```

659 <*xetex>

```

```

660 \RequirePackage { xeCJK }
661 \exp_args:Nx \xeCJKsetup
662 {
663     LoadFandol    = false ,
664     PunctStyle    = \l__ctex_punct_tl
665 }

```

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`。

```

666 \@ifpackagelater { fontspec } { 2014/05/25 } { }
667 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
668 </xetex>

```

14.3.3 ctex-engine-luatex.def

```

669 <*/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。

```

670 \msg_new:nnn { ctex } { luatexja-loaded }
671 {
672     Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
673     Loading~file~`#1'~will~abort!
674 }
675 \@ifpackageloaded { luatexja }
676 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
677 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
678 \RequirePackage { luatexja }
679 \@ifpackagelater { luatexja } { 2015/09/21 } { }
680 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
681 \RequirePackage { fontspec }
682 \@ifpackagelater { fontspec } { 2014/05/25 } { }
683 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

14.3.3.1 LuaTeX-jā 的默认设置

```

684 \ExplSyntaxOff

```

以下设置抄录自 `lltjdefs.sty`, 略有改动。

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

```

685 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF }
686 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF }
687 \ltjdefcharrange{3}{ %
688     "2000-"206F, "2070-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF,
689     "2B00-"2BFF }
690 \ltjdefcharrange{4}{ %
691     "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
692     "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
693     "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF } % non-Japanese
694 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF }
695 \ltjdefcharrange{6}{ %
696     "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
697     "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF }
698 \ltjdefcharrange{7}{ %
699     "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF,
700     "A830-"A83F, "AC00-"D7FF }
701 \ltjdefcharrange{8}{ "A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7 }

```

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

```

702 \ltjdefcharrange{9}{ %
703     "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A }

```

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

```

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

```

以下设置抄录自 ltj-latex.sty。

```

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

```

14.3.3.2 LuaTeX-ja 的补丁

```

717 <@@=ctex_ltj>

```

在 L^AT_EX 下, LuaTeX-ja 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。

```

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

```

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

```

726 \cs_set_protected:Npn \fontspec_visible_space:
727 {
728   \etex_iffontchar:D \tex_font:D "2423 \exp_stop_f:
729   \ltjalchar "2423 \exp_stop_f:
730   \else:
731     \fontspec_visible_space_fallback:
732   \fi:
733 }

```

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。

```

734 \ctex_at_end_package:nn { listings }
735 {
736   \use:x
737   {
738     \exp_not:N \RequirePackage { lltpj-listings }
739     \tl_set:Nn \exp_not:N \lstlistingname
740     { \exp_not:o { \lstlistingname } }
741     \tl_set:Nn \exp_not:N \lstlistlistingname
742     { \exp_not:o { \lstlistlistingname } }
743   }
744 }

```

14.3.3.3 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。

```

745 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
746 {
747   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }

```

```

748     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
749   }
750 \tl_new:N \CJK@family
751 \tl_new:N \l__ctex_ltj_current_font_tl
752 \tl_set:Nn \l__ctex_ltj_current_font_tl
753   { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

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

```

754 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
755   {
756     \group_begin:
757     \tl_set_eq:NN \f@encoding \CJK@encoding
758     \tl_set_eq:NN \f@family \CJK@family
759     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
760     \ctex_ltj_pickup_font:
761     \group_end:
762     \font@name
763     \__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。

```

764     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
765     { \__ctex_ltj_select_font_aux: }
766   }
767 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
768   {
769     \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
770     \cs_gset_nopar:Npx \font@name {#1}
771   }
772 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
773   { \cs_gset_eq:NN \font@name \__ctex_ltj_save_fontname:w }

```

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

```

774 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
775   {
776     \exp_after:wN \cs_if_exist:NF \font@name
777     {
778       \group_begin:
779       \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
780       \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
781       \define@newfont
782       \group_end:
783     }
784   }
785 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

%%%% \jfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX(false, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\protected\def\globaljfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX(true, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\def\ltj@@jfont{\directlua{luatexja.jfont.jfontdefY()}}

```

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

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

```

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

```
786 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
787 {
788   \get@external@font
789   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
790   {
791     \tl_set:Nx \external@font
792     { \exp_after:wN __ctex_ltj_patch_external_font:w \external@font }
793   }
794   \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。

```
795   \font@name
796   \lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
797   \use:c { \f@encoding + \f@family }
798   \use:c { \curr@fontshape }
799 }

```

\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。

```
800 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
801 {
802   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
803   {
804     \group_begin:
805     \tl_set_eq:NN \CJK@family \f@family
806     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
807     {
808       \cs_gset_protected_nopar:Npx \subst@correction
809       {
810         \cs_new_eq:NN
811         \exp_not:c { \l__ctex_ltj_current_font_tl }
812         \font@name
813       }
814       \group_insert_after:N \group_insert_after:N
815       \group_insert_after:N \subst@correction
816     }
817     \group_end:
818   }
819 }

```

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

```
820 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
821 {

```

```

822 \lua_now_x:n { luatexja.jfont.does_alt_set ('lua_escape_x:n {#1}') }
823 \prg_return_true: \else: \prg_return_false: \fi:
824 }

```

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。

```

825 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
826 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

在 `\selectfont` 中更新替代字体。

```

827 \cs_new_protected_nopar:Npn \ctex_ltj_select_alterfont:
828 {
829   \ctex_ltj_if_alterfont_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
830   {
831     \lua_now_x:n
832     {
833       luatexja.jfont.output_alt_font_cmd
834       ('y', 'lua_escape_x:n { \l__ctex_ltj_current_shape_tl }')
835     }
836     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a ('f@size') }
837   }
838 }
839 \tl_new:N \l__ctex_ltj_current_shape_tl
840 \tl_set:Nn \l__ctex_ltj_current_shape_tl
841 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

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

```

842 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
843 {
844   \cs_if_exist:cF { #1/f@size }
845   {
846     \group_begin:
847     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
848     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
849     \ctex_ltj_pickup_font:
850     \group_end:
851     \__ctex_ltj_pop_fontname:
852   }
853 }
854 <@=@>

```

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。

```

855 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
856 {
857   \ltj@@getjfontnumber #1
858   \lua_now_x:n
859   {
860     luatexja.jfont.pickup_alt_font_b
861     ( \int_use:N \ltj@tempcntc, 'lua_escape_x:n {#2}' )
862   }
863 }

```


14.3.3.4 数学字体族

864 `<@@=ctex_ljtj>`

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_ljtj_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`。

```
865 \cs_new:Npn \ctex_ljtj_if_jfont:nTF #1
866 {
867   \lua_now_x:n
868   { \luaescape_x:n {string.match('\lua_escape_x:n {#1}', '[~/]+') } }
869   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
870 }
```

`\ctex_ljtj_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。

```
871 \cs_new:Npn \ctex_ljtj_if_jfont_math:NTF #1
872 { \exp_after:wN \__ctex_ljtj_if_jfont_math:w \token_to_str:N #1 \q_stop }
873 \group_begin:
874 \char_set_catcode_other:N M
875 \cs_new:Npn \__ctex_ljtj_if_jfont_math:w #1 M #2#3 \q_stop
876 { \ctex_ljtj_if_jfont:nTF {#3} }
877 \group_end:
878 <@@= >
```

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

`\ctex_ljtj_get_and_define_fonts:nN`

```
879 \cs_new_protected_nopar:Npn \ctex_ljtj_get_and_define_fonts:nN #1#2
880 {
881   \ctex_ljtj_if_jfont:nTF { \token_to_str:N #2 }
882   { \ctex_ljtj_get_and_define_fonts_ja:nN }
883   { \ctex_ljtj_get_and_define_fonts_al:nN }
884   {#1} #2
885 }
886 \cs_new_eq:NN \ctex_ljtj_get_and_define_fonts_al:nN \getanddefine@fonts
887 \cs_set_eq:NN \getanddefine@fonts \ctex_ljtj_get_and_define_fonts:nN
888 \cs_new_protected_nopar:Npn \ctex_ljtj_get_and_define_fonts_ja:nN #1#2
889 {
890   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
891   \ctex_ljtj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
892   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
893   \ctex_ljtj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
894   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
895   \ctex_ljtj_pickup_font:
896   \tl_put_right:Nx \math@fonts
897   {
898     \ltj@setpar@global
899     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
900     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
901     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
902   }
903 }
```

904 `<@@=ctex_ljtj>`

`\use@mathgroup`
`\ctex_ljtj_use_math_group:Nn`

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

```
905 \cs_new_protected_nopar:Npn \ctex_ljtj_use_math_group:Nn #1#2
906 {
```

```

907 \mode_if_math:T
908 {
909   \math@bgroup
910   \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
911   \ctex_ltj_math_group_hook:
912   \ctex_ltj_if_jfont_math:NTF #1
913   { \jfam } { \mathgroup } #2 \scan_stop:
914   \math@egroup
915 }
916 }
917 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
918 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

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

```

919 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
920 {
921   \group_begin:
922   \cs_set_protected:Npn #1 ##1##2##3
923     { \ltjsetmathletter { ##1 } }
924   #2
925   \group_end:
926 }
927 \ctex_at_end_package:nn { unicode-math }
928 {
929   \cs_if_exist:NTF \um_input_math_symbol_table:
930   {
931     \ctex_ltj_set_math_letter:NN
932     \um_sym:nnn
933     \um_input_math_symbol_table:
934   }
935   {
936     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
937     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
938       { \__um_switchto_literal: }
939     \ctex_ltj_set_math_letter:NN
940     \__um_sym:nnn
941     \__um_input_math_symbol_table:
942   }
943 }

```

14.3.3.5 字体族的定义与使用

```

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

```

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

```

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 }

```

```

\CJK@encoding
\__ctex_ltj_change_encoding:

```

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

```

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.jfont.add_kyenc_list('\CJK@encoding') }
961 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
962 { \tl_set_eq:NN \g_fontspec_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

```

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_fontname_wrap: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_fontname_wrap: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_alterate_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_alterate_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 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1004 \clist_new:N \l__ctex_ltj_font_options_clist
1005 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1006 {
1007   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1008   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1009 }

```

分别保存 `fontspec` 设置的字体族名、字体名称和字体选项。

```

1010 \prop_new:N \g__ctex_ltj_family_name_prop

```

```

\g__ctex_ltj_family_name_prop
\g__ctex_ltj_family_font_name_prop
\g__ctex_ltj_family_font_options_prop

```

```

1011 \prop_new:N \g__ctex_ltj_family_font_name_prop
1012 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

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

```

1013 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1014 {
1015   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1016   {
1017     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1018     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1019     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1020     {
1021       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1022       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1023       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1024       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1025     }
1026     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1027   }
1028 }
1029 \tl_new:N \l__ctex_ltj_tmp_tl
1030 \msg_new:nnn { ctex } { redefine-family }
1031 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#2). }

```

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

```

1032 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1033 {
1034   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }
1035   {
1036     \group_begin:
1037     \__ctex_ltj_change_encoding:
1038     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1039     \exp_not:n { \cs_set_eq:NN \CTEX@fontfamily \use_none:n }
1040     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1041     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1042     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1043     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1044     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1045     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1046     \__ctex_ltj_set_alternate_family:n {#1}
1047     \group_end:
1048   }
1049 }
1050 \tl_new:N \l__ctex_ltj_base_family_tl
1051 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1052 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1053 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1054 {
1055   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1056   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1057   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1058   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1059 }
1060 \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

\CJKfamily 切换字体。

```

1061 \NewDocumentCommand \CJKfamily { m }
1062 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1063 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1064 {
1065   \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1066   {
1067     \tl_set:Nn \l__ctex_ltj_family_tl {#1}

```

```

1068     \selectfont
1069   }
1070   { \_ctex_ltj_family_unknown_warning:n {#1} }
1071 }
1072 \tl_new:N \l_ctex_ltj_family_tl
1073 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

判断 CJK 字体族 #1 是否存在,若存在则把实际族名保存到 #2 中。

```

1074 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
1075 {
1076   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1077   { \prg_return_true: }
1078   {
1079     \cs_if_exist_use:cTF { \_ctex_ltj_family_csname:n {#1} }
1080     {
1081       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1082       \prg_return_true:
1083     }
1084     { \prg_return_false: }
1085   }
1086 }
1087 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { x } { T , F , TF }

```

```

1088 \cs_new_protected_nopar:Npn \_ctex_ltj_family_unknown_warning:n #1
1089 {
1090   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1091   {
1092     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1093     {
1094       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1095       \msg_warning:nnn { ctex } { family-unknown } {#1}
1096     }
1097   }
1098 }
1099 \seq_new:N \g__ctex_ltj_unknown_family_seq
1100 \msg_new:nnn { ctex } { family-unknown }
1101 {
1102   Unknown~CJK~family~\_ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1103   Try~to~use~\_ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1104 }
1105 \cs_new_nopar:Npn \_ctex_ltj_msg_def_family_map:n #1
1106 {
1107   \str_case_x:nnF {#1}
1108   {
1109     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1110     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1111     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1112   }
1113   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1114   [...] \{ ... \}
1115 }
1116 \cs_new_nopar:Npn \_ctex_ltj_msg_family_map:n #1
1117 {
1118   \str_case_x:nnF {#1}
1119   {
1120     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1121     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1122     \CJKttdefault { \token_to_str:N \CJKttdefault }
1123   }
1124   {#1}
1125 }

```

```

1126 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1127 {
1128   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1129   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl

```

```

1130     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1131     {
1132         \int_gincr:N \g__ctex_ltj_family_int
1133         \__ctex_ltj_fontspec:xnn
1134         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1135         {#1} {#2}
1136     }
1137 }
1138 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1139 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1140 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1141 {
1142     \bool_if:NT \l__ctex_ltj_add_alterate_bool
1143     {
1144         \cs_if_free:cF
1145         { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1146         {
1147             \cs_gset_eq:cc
1148             { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1149             { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1150             \cs_gset_eq:cc
1151             { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1152             { \__ctex_ltj_alterate_cs:n { clear / \l_ctex_ltj_family_tl } }
1153         }
1154         \bool_set_false:N \l__ctex_ltj_add_alterate_bool
1155     }
1156     \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1157     \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1158     \ctex_ltj_switch_family:n {#1}
1159 }
1160 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1161 \prop_new:N \g__ctex_ltj_fontspec_prop

\ctex_ltj_add_font_features:n 1162 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1163 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1164 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1165 {
1166     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1167     {#1} \l__ctex_ltj_tmp_tl
1168     {
1169         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1170         {#1} \l__ctex_ltj_font_options_clist
1171         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1172         \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1173         \ctex_ltj_fontspec:xx
1174         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1175         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1176     }
1177     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1178 }
1179 \bool_new:N \l__ctex_ltj_add_alterate_bool
1180 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1181 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1182 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1183 {
1184     \token_to_str:N \addCJKfontfeature (s)~ignored.\
1185     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1186 }

```

为了支持字体属性可选项在前在后两种语法,给出两个辅助工具,自带展开功能。

```

1187 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1188 {
1189     \IfNoValueTF {#2}
1190     { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1191     {
1192         \use:x { #1 {#2} {#3} }

```

```

1193         #4
1194     }
1195 }
1196 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1197 {
1198     \use:x { #1 {#4} {#2} }
1199     #3
1200 }

```

```

\setCJKfamilyfont 1201 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 1202 {
    \CJKfontspec 1203     \__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 1204     { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1205     { }
1206 }
1207 \NewDocumentCommand \newCJKfontfamily { o m o m }
1208 {
1209     \tl_set:Nx \l__ctex_ltj_tmp_tl
1210     { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1211     \cs_new_protected_nopar:Npx #2
1212     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1213     \__ctex_ltj_pass_args:nnnn
1214     { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1215     { }
1216 }
1217 \NewDocumentCommand \CJKfontspec { o m }
1218 {
1219     \__ctex_ltj_pass_args:nnnn
1220     { \ctex_ltj_fontspec:nn } {#1} {#2}
1221     { \tex_ignorespaces:D }
1222 }
1223 \NewDocumentCommand \addCJKfontfeatures { m }
1224 {
1225     \ctex_ltj_add_font_features:x {#1}
1226     \tex_ignorespaces:D
1227 }
1228 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

```

\setCJKmainfont 1229 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1230 {
\setCJKmonofont 1231     \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1232     { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1233     { \normalfont }
1234 }
1235 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1236 \NewDocumentCommand \setCJKsansfont { o m }
1237 {
1238     \__ctex_ltj_pass_args:nnnn
1239     { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1240     { \normalfont }
1241 }
1242 \NewDocumentCommand \setCJKmonofont { o m }
1243 {
1244     \__ctex_ltj_pass_args:nnnn
1245     { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1246     { \normalfont }
1247 }
1248 \NewDocumentCommand \setCJKmathfont { o m }
1249 {
1250     \__ctex_ltj_pass_args:nnnn
1251     { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1252     { }
1253 }
1254 \NewDocumentCommand \defaultCJKfontfeatures { m }
1255 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1256 \clist_new:N \g__ctex_ltj_default_features_clist

```



```

1257 \onlypreamble \setCJKmainfont
1258 \onlypreamble \setCJKsansfont
1259 \onlypreamble \setCJKmonofont
1260 \onlypreamble \setCJKmathfont
1261 \onlypreamble \setCJKromanfont
1262 \onlypreamble \defaultCJKfontfeatures

1263 \tl_if_exist:NF \CJKfamilydefault
1264 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1265 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1266 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1267 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }

```

\fontfamily
\CTEX@fontfamily

修改 \fontfamily, 使主要 CJK 字体族能随西文主要字体更新。

```

1268 \RenewDocumentCommand \fontfamily { m }
1269 {
1270   \tl_set:Nx \f@family {#1}
1271   \CTEX@fontfamily {#1}
1272 }
1273 \cs_new_protected_nopar:Npn \CTEX@fontfamily #1
1274 {
1275   \str_if_eq:nnTF {#1} { \familydefault }
1276   { \CJKfamily { \CJKfamilydefault } }
1277   { \__ctex_ltj_update_family_aux: }
1278 }
1279 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_aux:
1280 {
1281   \str_case:x:nn { \f@family }
1282   {
1283     { \rmdefault } { \CJKfamily { \CJKrmdefault } }
1284     { \sfdefault } { \CJKfamily { \CJKsfdefault } }
1285     { \ttdefault } { \CJKfamily { \CJKttdefault } }
1286     { \familydefault } { \CJKfamily { \CJKfamilydefault } }
1287   }
1288 }

```

在导言区末尾更新 \CJKfamilydefault。

```

1289 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

\ctex_ltj_ensure_default_family:

在导言区结束确认 \CJKfamilydefault 确实存在。

```

1290 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1291 {
1292   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1293   {
1294     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1295     {
1296       \str_if_eq:x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1297       { \use:n }
1298       {
1299         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1300         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1301         { \use:n }
1302       }
1303     }
1304     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1305     {
1306       \prop_map_break:n
1307       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1308     }
1309   }
1310 }
1311 \normalfont
1312 \ctex_ltj_update_mathfont:
1313 }
1314 }

```

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

```

1315 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1316 {
1317   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1318   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1319   {
1320     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1321     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1322   }
1323 }
1324 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1325 {
1326   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1327   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1328   { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1329   \cs_if_free:CTF
1330   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1331   {
1332     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1333     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1334   }
1335   {
1336     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1337     { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1338   }
1339 }

```

14.3.3.6 替代字体的设置

AlternateFont
CharRange 设置替代字体的选项。

```

1340 \keys_define:nn { ctex_ltj / fontspec }
1341 {
1342   AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1343   AlternateFont .value_required:n = true ,
1344   CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1345   CharRange .value_required:n = true
1346 }

```

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

```

1347 \group_begin:
1348   \char_set_catcode_other:N \l
1349   \cs_set:Npn \__ctex_ltj_tmp:w #1
1350   {
1351     \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1352     {
1353       \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1354       {
1355         \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1356         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1357         \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1358         \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1359         { ! \tl_if_blank_p:n { ###1 } }
1360         \seq_concat:NNN \l__ctex_ltj_alternate_seq
1361         \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1362       }
1363     }
1364   }
1365   \char_set_catcode_active:N \l
1366   \__ctex_ltj_tmp:w { || }
1367 \group_end:
1368 \seq_new:N \l__ctex_ltj_tmp_seq
1369 \seq_new:N \l__ctex_ltj_alternate_seq

```

\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。

```

1370 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1371 {
1372   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1373   {
1374     \__ctex_ltj_check_family:n {#1}
1375     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1376     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1377     #3
1378   }
1379   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1380 }

```

\ctex_ltj_save_alternate_seq:Nn
\ctex_ltj_save_alternate_seq:Nnnwn

保存由 AlternateFont 设置的替代字体序列。

```

1381 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1382 {
1383   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1384   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1385 }
1386 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1387 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1388 { m m m +0{ } } m u{ \q_stop } }
1389 {
1390   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1391   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1392   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1393   \tl_if_blank:nTF {#5}
1394   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1395   {
1396     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1397     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1398   }
1399   \use:x
1400   {
1401     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1402     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1403     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1404     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1405   }
1406 }
1407 \clist_new:N \l__ctex_ltj_alternate_options_clist

```

\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。

```

1408 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1409 {
1410   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1411   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1412   \ctex_ltj_set_alternate_family:coonn
1413   { \__ctex_ltj_alternate_cs:n {#1} }
1414   { \l__ctex_ltj_char_range_clist }
1415   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1416 }
1417 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1418 {
1419   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1420   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1421   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1422 }
1423 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

\ctex_ltj_save_alternate_family:Nnnn

保存替代字体序列的定义, 以备定义主字体时使用。

```

1424 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4

```

```

1425 {
1426   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1427   \cs_gset_protected_nopar:Npx #1
1428     { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1429 }

```

\ctex_ltj_set_alternate_family:nnn

实际定义替代字体族。

```

1430 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1431 {
1432   \group_begin:
1433   \__ctex_ltj_change_encoding:
1434   \cs_set_eq:NN \CJKfamily \use_none:n
1435   \ctex_ltj_swap_cs:NN
1436   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1437   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1438   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1439   \group_end:
1440 }
1441 \tl_new:N \l__ctex_ltj_alternate_family_tl

```

\ctex_ltj_swap_cs:NN

交换两个控制序列的意义。

```

1442 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1443 {
1444   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1445   \cs_set_eq:NN #1 #2
1446   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1447   \cs_undefine:N \__ctex_ltj_tmp:w
1448 }

```

LTJFONTUID

__ctex_ltj_update_family_uid:N

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

```

1449 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1450 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1451 {
1452   \int_gincr:N \g__ctex_ltj_family_int
1453   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1454 }
1455 \int_new:N \g__ctex_ltj_family_int

```

\ctex_ltj_declare_alternate_shape:nnnnnn

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

```

1456 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1457 {
1458   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1459   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1460     { \l__ctex_ltj_base_family_tl } {#3} {#4}
1461     { \l_fontspec_family_tl } {#3} {#4}
1462 }

```

\ctex_ltj_set_alternate_shape:Nnnnnnn

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

```

1463 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1464 {
1465   \clist_map_inline:Nn #1
1466   {
1467     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1468     {
1469       \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1470       \l__ctex_ltj_char_range_tl

```

```

1471     }
1472     { \ctex_ltj_set_altername_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1473   }
1474   \__ctex_ltj_save_altername_shape:cnn
1475   { \__ctex_ltj_altername_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1476   { luatexja.jfont.clear_alt_font_latex }
1477   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1478 }

```

\ctex_ltj_set_altername_shape:nnn

我们使用 `->` 而不是像 **LuaTeX-j**a 一样使用 `-` 作为区间的分隔符。**LuaTeX-j**a 支持使用负数来引用由 **JFM** 设置的字符类。如果使用 `-` 作为分隔符,那么负数单独使用时,就需要把它放在两层花括号之内(例如 `{{-1}}`),或者使用类似 `{{-1}}-{{-1}}` 的形式才不会解释错误。

```

1479 \NewDocumentCommand \ctex_ltj_set_altername_shape:nnn
1480 { m m > { \SplitArgument { \c_one } { -> } } m }
1481 { \ctex_ltj_set_altername_shape:nnnn {#1} {#2} #3 }
1482 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:nnnn #1#2#3#4
1483 {
1484   \ctex_ltj_set_altername_shape:n
1485   {
1486     \IfNoValueTF {#4}
1487     { \int_eval:n {#3} , \int_eval:n {#3} , }
1488     {
1489       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1490       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1491     }
1492     '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1493     '\lua_escape_x:n { \CJK@encoding/#1 }'
1494   }
1495 }
1496 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:n #1
1497 {
1498   \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1499   \__ctex_ltj_save_altername_shape:cnn
1500   { \__ctex_ltj_altername_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1501   { luatexja.jfont.set_alt_font_latex } {#1}
1502 }

```

\ctex_ltj_set_altername_shape:nnN

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

```

1503 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:nnN #1#2#3
1504 {
1505   \tl_map_inline:Nn #3
1506   {
1507     \ctex_ltj_set_altername_shape:n
1508     {
1509       ##1 ,
1510       '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1511       '\lua_escape_x:n { \CJK@encoding/#1 }'
1512     }
1513   }
1514 }

```

__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。

```

1515 \cs_new_protected_nopar:Npn \__ctex_ltj_save_altername_shape:Nnn #1#2#3
1516 {
1517   \group_begin:
1518   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1519   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1520   \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1521   \cs_gset_protected_nopar:Npx #1
1522   { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }

```

```

1523 \group_end:
1524 }
1525 \cs_generate_variant:Nn \__ctex_ltj_save_altername_shape:Nnn { c }

```

clearalternamefont
resetalternamefont

清除和重置操作总是全局的。

```

1526 \keys_define:nn { ctex }
1527 {
1528   clearalternamefont .code:n =
1529   { \clist_map_function:xN {#1} \ctex_ltj_clear_altername_font:n } ,
1530   resetalternamefont .code:n =
1531   { \clist_map_function:xN {#1} \ctex_ltj_reset_altername_font:n } ,
1532   clearalternamefont .default:n = \l_ctex_ltj_family_tl ,
1533   resetalternamefont .default:n = \l_ctex_ltj_family_tl
1534 }
1535 \cs_new_protected_nopar:Npn \ctex_ltj_clear_altername_font:n #1
1536 {
1537   \group_begin:
1538   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1539   {
1540     \cs_if_exist_use:cT { \__ctex_ltj_altername_cs:n { clear / #1 } }
1541     {
1542       \prop_gput:Nno \g__ctex_ltj_reset_altername_prop
1543       {#1} { \l__ctex_ltj_base_family_tl }
1544       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1545       \selectfont
1546     }
1547   }
1548   { \__ctex_ltj_family_unknown_warning:n {#1} }
1549   \group_end:
1550 }
1551 \cs_new_protected_nopar:Npn \ctex_ltj_reset_altername_font:n #1
1552 {
1553   \group_begin:
1554   \prop_gpop:NnNT \g__ctex_ltj_reset_altername_prop {#1} \CJK@family
1555   {
1556     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1557     \use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1558     \selectfont
1559   }
1560   \group_end:
1561 }
1562 \prop_new:N \g__ctex_ltj_reset_altername_prop
1563 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

declarecharrange

预先声明字符范围。

```

1564 \keys_define:nn { ctex }
1565 {
1566   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1567   declarecharrange .value_required:n = true
1568 }
1569 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1570 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1571 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1572 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1573 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

#1 是名字, **#2** 是范围。

```

1574 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1575 {
1576   \tl_clear:N \l__ctex_ltj_char_range_tl
1577   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1578   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1579   \ctex_ltj_def_char_range_key:n {#1}
1580   \tl_clear:N \l__ctex_ltj_char_range_tl

```

\ctex_ltj_declare_char_range:nn
\g__ctex_ltj_char_range_prop

```

1581 }
1582 \tl_new:N \l__ctex_ltj_char_range_tl
1583 \prop_new:N \g__ctex_ltj_char_range_prop

```

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

```

1584 \NewDocumentCommand \ctex_ltj_save_char_range:n
1585 { > { \SplitArgument { \c_one } { -> } } m }
1586 { \ctex_ltj_save_char_range:nn #1 }
1587 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1588 {
1589   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1590   { {
1591     \IfNoValueTF {#2}
1592     { \int_eval:n {#1} , \int_eval:n {#1} }
1593     {
1594       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1595       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1596     }
1597   } }
1598 }

```

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

```

1599 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1600 {
1601   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1602   {
1603     \keys_define:nn { ctex_ltj / fontspec }
1604     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1605   }
1606 }

```

\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。

```

1607 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1608 {
1609   \tl_if_blank:nTF {#2}
1610   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1611   {
1612     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1613     {
1614       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1615       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1616     }
1617   }
1618 }

```

__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 [simsun.ttc] 设置为 range 的主字体,就需要使用 range={{[simsun.ttc]}} 或者 []{[simsun.ttc]}。下面的目的是,支持直接使用 [simsun.ttc] 和 [...] [simsun.ttc]。

```

1619 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1620 { +o o u { \q_stop } }
1621 {
1622   \exp_args:NNf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1623   {
1624     \IfNoValueTF {#1} { {#3} }
1625     {
1626       \IfNoValueTF {#2}
1627       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ [ #1 ] ] {#3} } }
1628       { [ {#1} ] { [ #2 ] } }
1629     }
1630   }
1631   \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1632 }

```

14.3.3.7 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
}

1633 \AtBeginDocument
1634 {
1635   \ctex_appto_cmd:NnnTF \verbatim@font
1636   { \char_set_catcode_letter:n { 64 } }
1637   { \CTEX@verbatim@font@hook }
1638   { }
1639   { \ctex_patch_failure:N \verbatim@font }
1640 }
1641 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1642 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
```

`_ctex_ltjitaliccorr` L^AT_EX 的倾斜校正也要重新定义。

```
1643 <@@@=>
1644 \cs_set_eq:NN \@@italiccorr \/
1645 <@@=-ctex>
```

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

```
1646 \cs_new_protected_nopar:Npn \ctex_ltj_set_kanjiskip:N
1647 { \ltj@setpar@global \ltjsetkanjiskip }
1648 \cs_new_protected_nopar:Npn \ctex_ltj_set_xkanjiskip:N
1649 { \ltj@setpar@global \ltjsetxkanjiskip }

1650 </luatex>
```

14.3.4 ctex-engine-uptex.def

```
1651 <*uptex|aptex>
```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```
1652 \tl_set:Nn \CJKrmdefault { rm }
1653 \tl_set:Nn \CJKsfdefault { sf }
1654 \tl_set:Nn \CJKttdefault { tt }
1655 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1656 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1657 { \kanjifamily { \CJKrmdefault } }
1658 { }
1659 { \ctex_patch_failure:N \rmfamily }
1660 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1661 { \kanjifamily { \CJKsfdefault } }
1662 { }
1663 { \ctex_patch_failure:N \sffamily }
1664 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1665 { \kanjifamily { \CJKttdefault } }
1666 { }
```



```

1667 { \ctex_patch_failure:N \ttfamily }
1668 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1669 { \kanjifamily { \CJKfamilydefault } }
1670 { \cs_set_eq:NN \reset@font \normalfont }
1671 { \ctex_patch_failure:N \normalfont }
1672 \NewDocumentCommand \CJKfamily { m }
1673 {
1674   \kanjifamily { #1 }
1675   \selectfont
1676 }
1677 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```

在导言区末尾更新 `\CJKfamilydefault`。

```
1678 \ctex_at_end_preamble:n { \ctex_update_default_family: }
```

使修改立刻生效,保证导言区字体族正确。

```
1679 \normalfont
```

`\em` 取消 `upLATEX` 对 `\em` 使用 `\mcfamily`、`\gtfamily` 命令的重定义,恢复 `LATEX 2ε` 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。`upLATEX 2016/05/07u00` 的定义有所变化,这一行为可以由用户通过 `platexrelease` 包改变,需要分支处理。

```

1680 \ctex_patch_cmd_once:NnnnTF \em
1681 { \ExplSyntaxOff }
1682 { \eminnershape \else \gtfamily \itshape }
1683 { \eminnershape \else \itshape }
1684 { }
1685 {
1686   \ctex_patch_cmd:Nnn \em
1687     { \mcfamily \upshape \else \gtfamily \itshape }
1688     { \eminnershape \else \itshape }
1689 }
1690 \cs_set_nopar:Npn \eminnershape { \upshape }

```

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

```

1691 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1692 {
1693   \DeclareKanjiFamily{JY2}{#1}{}
1694   \DeclareKanjiFamily{JT2}{#1}{}
1695   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1696   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1697   \tl_if_empty:nF { #3 }
1698   {
1699     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1700     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1701   }
1702 }
1703 \@onlypreamble \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。

```

1704 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1705 {
1706   \ctex_set_zhmap:n
1707   {
1708     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1709     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1710     \tl_if_empty:nF { #3 }
1711     {
1712       \special{ pdf:mapline~ #1b-h~ UniGB-UTF16-H~ #3 }

```

```

1713         \special{ pdf:mapline~ #1b-v~ UniGB-UTF16-V~ #3 }
1714     }
1715 }
1716 }
1717 \onlypreamble \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。

```

1718 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1719 {
1720     \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1721     \ctex_set_upmap:nnn { upserifit } { #3 } {}
1722     \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1723     \ctex_set_upmap:nnn { upmono } { #6 } {}
1724 }
1725 \onlypreamble \ctex_set_upfonts:nnnnnn

```

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

```

1726 \RequirePackage { pxeveryysel }
1727 </uptex|aptex>

```

14.3.5 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。

```

1728 <*pdf|luatex|uptex|aptex>
1729 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1730 {
1731     \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1732     {
1733         \group_begin:
1734         \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1735         \tl_gset:Nx \CJKfamilydefault
1736         {
1737             \str_case:onF { \familydefault }
1738             {
1739                 { \rmdefault } { \exp_not:N \CJKrmdefault }
1740                 { \sfdefault } { \exp_not:N \CJKsfdefault }
1741                 { \ttdefault } { \exp_not:N \CJKttdefault }
1742             }
1743             { \CJKfamilydefault }
1744         }
1745         \group_end:
1746     }

```

使用 **LuaLaTeX** 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 **upLaTeX** 时, `C19rm`、`JY2rm` 等总是有定义的, 不需要确认。

```

1747 <*luatex>
1748     \ctex_ljtj_ensure_default_family:
1749 </luatex>
1750 }

```

\l__ctex_family_default_init_tl

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

```

1751 \tl_new:N \l__ctex_family_default_init_tl
1752 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1753 \tl_set:Nx \l__ctex_family_default_init_tl
1754 {
1755     \exp_not:N \__ctex_family_default_wrap:n

```

```

1756     { \exp_not:o { \CJKfamilydefault } }
1757   }
1758   \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1759   </pdf|luatex|uptex|aptex>

```

14.3.6 操作系统的判断

`\ctex_detected_platform:` 在 LuaTeX 下直接用调用 `os.name` 来判断。

```

1760 <*luatex>
1761 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1762 {
1763   \tl_gset:Nx \g__ctex_fontset_tl
1764   {
1765     \lua_now_x:n
1766     {
1767       if ~ os.name == 'windows' then ~
1768         tex.sprint ( 'windows' )
1769       elseif ~ os.name == 'macosx' then ~
1770         tex.sprint ( 'mac' )
1771       else ~
1772         tex.sprint ( 'fandol' )
1773       end
1774     }
1775   }
1776 }
1777 </luatex>

```

ApTeX 可以使用 `\ngostype` 来判断。

```

1778 <*aptex>
1779 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1780 {
1781   \tl_gset:Nx \g__ctex_fontset_tl
1782   {
1783     \str_case:onF { \ngostype }
1784     {
1785       { Win32 } { windows }
1786       { Win64 } { windows }
1787       { Darwin } { mac }
1788     }
1789     { fandol }
1790   }
1791 }
1792 </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!

```

1793 <*xetex|pdf|uptex>
1794 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1795 {
1796   \file_if_exist:nTF { /dev/null }
1797   {
1798     \file_if_exist:nTF { nul: }
1799     {
1800       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1801       { \ctex_if_macosx:TF { mac } { windows } }
1802       { \ctex_if_macosx:TF { mac } { fandol } }
1803     }
1804     { \ctex_if_macosx:TF { mac } { fandol } }
1805   }
1806   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1807 }
1808 \str_const:Nx \c__ctex_upper_case_file_str

```

```

1809 { \exp_args:No \str_upper_case:n { \g_file_curr_name_str } }

\ctex_if_macosx:TF 以 /Library/Fonts/Songti.ttc 为特征文件判断 Mac OS X。
1810 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1811 {
1812   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1813   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1814   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1815 }
1816 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1817 </xetex|pdfTeX|uptex>

```

14.3.7 hyperref 兼容性处理

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

```

\ctex_hypersetup:n 如果已经载入 hyperref 宏包, 则直接使用其定义设置选项; 否则 \ctex_hypersetup:n 的效果
                    与 \PassOptionsToPackage 一致, 只传递宏包参数。如果用户不载入 hyperref 宏包, 相关
                    参数即被丢弃。
1818 <*class|ctex>
1819 \@ifpackageloaded { hyperref }
1820 {
1821   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1822   { \hypersetup {#1} }
1823 }
1824 {
1825   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1826   { \PassOptionsToPackage {#1} { hyperref } }
1827 }
1828 </class|ctex>

```

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。

```

1829 <*pdfTeX>
1830 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1831 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1832 {
1833   \ctex_hypersetup:n { CJKbookmarks = true }
1834   \sys_if_output_pdf:TF
1835   { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1836   {
1837     \ctex_at_end_package:nn { hyperref }
1838     {
1839       \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1840       {
1841         \AtBeginShipoutFirst
1842         { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1843       }
1844       { \RequirePackage { xCJK2uni } }
1845     }
1846   }
1847 }
1848 { \ctex_hypersetup:n { pdfencoding = unicode } }
1849 </pdfTeX>

```

在 XeTeX 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

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

驱动文件 `hxdetex.def` 中设置了 `\Hy@unicodetrue`, 从而书签总是会被 `\HyPsd@ConvertToUnicode` 转化成 UTF-16BE 编码的形式(抄录自 `\pdfstringdef` 的定义):

```
\ifHy@unicode
\HyPsd@ConvertToUnicode#1%
\ifx\HyPsd@pdfencoding\HyPsd@pdfencoding@auto
\ltx@ifundefined{StringEncodingConvertTest}{%
}{%
\edef\unescapestring\HyPsd@temp#1%
\ifxetex
\let\HyPsd@UnescapedString\HyPsd@temp
\StringEncodingConvertTest\HyPsd@temp\HyPsd@temp
{utf16be}{ascii-print}{%
\edef\escapestring\HyPsd@temp\HyPsd@temp
\global\let#1\HyPsd@temp
\HyPsd@EscapeTeX#1%
\Hy@unicodetrue
}{%
\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_YTeX 的维护者 Khaled Hosny 已经注意到了这个问题¹⁹。需要注意的是, `hxdetex.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 `CTYTeX` 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 X_YTeX 下更保险的 `pdfencoding` 选项来设置。为了与 X_YTeX 下的行为一致(使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 LuaTeX 下也启用这个选项。

```
1850 <*xetex|luatex>
1851 \ctex_hypersetup:n { pdfencoding = unicode }
1852 </xetex|luatex>
```

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

```
1853 <*uptex|aptex>
1854 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1855 \ctex_at_end_package:nn { hyperref }
1856 { \AtBeginShipoutFirst { \special { pdf:tounicode~ UTF8-UTF16 } } }
1857 </uptex|aptex>
1858 <*pdfTeX|xetex|luatex|uptex|aptex>
```

14.3.8 CJKfntef、xeCJKfntef 相关设置

CT_YTeX 宏集对 pdfTeX 与 X_YTeX 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。

```
1859 <*pdfTeX>
1860 \RequirePackage { CJKfntef }
1861 \normalem
1862 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
```

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

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

```

1863 { \tl_clear:c { CJK#1color } }
1864 </pdfTeX>
1865 <*xetex>
1866 \RequirePackage { xeCJKfntef }
1867 \ifpackageclater { xeCJKfntef } { 2014/11/04 }
1868 {
1869   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1870   { \xeCJKsetup { #1 / format = { } } }
1871 }
1872 {
1873   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1874   { \tl_clear:c { CJK#1color } }
1875 }
1876 </xetex>
1877 <*luatex|uptex|aptex>
1878 \msg_new:nnn { ctex } { fntef-not-available }
<luatex> 1879 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
<uptex> 1880 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
<aptex> 1881 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1882 \msg_warning:nn { ctex } { fntef-not-available }
1883 </luatex|uptex|aptex>
1884 \clist_map_inline:nn
1885 { underdot , underline , underdblline , underwave , sout , xout }
1886 <*pdfTeX|xetex>
1887 {
1888   \__ctex_clear_fntef_color:n {#1}
1889   \cs_new_protected_nopar:cpx { CTEX#1 }
1890   {
1891     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1892     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1893     \exp_not:c { CJK#1 }
1894   }
1895 }
1896 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1897 {
1898   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1899   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1900   \CJKfilltwosides
1901 }
1902 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1903 </pdfTeX|xetex>
1904 <*luatex|uptex|aptex>
1905 { \cs_new_eq:cN { CTEX#1 } \use:n }
1906 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1907 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1908 </luatex|uptex|aptex>
1909 <*pdfTeX>
1910 \clist_map_inline:nn
1911 {
1912   underdotbasesep , underdotsep , underlinebasesep ,
1913   underlinesep , underdbllinesep , underdbllinebasesep ,
1914   underwavebasesep , underwavesep , southeight ,
1915   underdotcolor , underwavecolor , underlinecolor ,
1916   underdbllinecolor , soutcolor , xoutcolor
1917 }
1918 {
1919   \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1920   \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1921 }
1922 </pdfTeX>

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1923 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1924 <*pdfTeX|xetex>
1925 {

```

```

1926 \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1927 \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1928 }
1929 </pdfTeX|xetex>
1930 <*luatex>
1931 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1932 </luatex>
1933 <*uptex|aptex>
1934 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1935 </uptex|aptex>
1936 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1937 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1938 <*pdfTeX|xetex>
1939 {
1940   \cs_set_protected_nopar:Npn \CJKglue
1941     { \skip_horizontal:N \l__ctex_ccglue_skip }
1942 }
1943 </pdfTeX|xetex>
1944 <*luatex>
1945 { \ctex_ljt_set_kanjiskip:N \l__ctex_ccglue_skip }
1946 </luatex>
1947 <*uptex|aptex>
1948 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1949 </uptex|aptex>
1950 \skip_new:N \l__ctex_ccglue_skip

```

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

\ctex_if_ccglue_touched: **TF**

```

1951 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1952 {
1953 <*pdfTeX|xetex>
1954   \if_meaning:w \CJKglue \__ctex_ccglue:
1955     \prg_return_false: \else: \prg_return_true: \fi:
1956 </pdfTeX|xetex>
1957 <*luatex>
1958   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1959     { \prg_return_false: } { \prg_return_true: }
1960 </luatex>
1961 <*uptex|aptex>
1962   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1963     { \prg_return_false: } { \prg_return_true: }
1964 </uptex|aptex>
1965 }

```

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

```

1966 <*pdfTeX|xetex>
1967 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1968 </pdfTeX|xetex>

```

\ctex_update_em_unit: 将当前汉字的宽度保存到 \ccwd 中备用。不采用 1em, 因为这时的 1em 实际上来自西文字体的信息, 未必等于汉字的宽度, 这似乎在传统的 .tfm 字体上表现更明显。在 pdfTeX 和 XeTeX 下, 直接使用 \f@size\p@ 作为汉字的宽度, 这应该对大多数汉字字体都成立, 但不适用于诸如“方正兰亭黑长”之类的特殊字体。在 XeTeX 可以用 \fontcharwd 来改进。而在 pdfTeX 下, 若使用 zhmetrics 技术, 所有的汉字共享同一个 .tfm, \fontcharwd 也就没有意义。在 LuaTeX 下, LuaTeX-ja 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 \zw 作为汉字宽度。upTeX 可以直接使用原生的长度单位 zw。

```

1969 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
<pdfTeX|xetex> 1970 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1971 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1972 { \dim_set:Nn \ccwd { 1zw } }

```

14.3.10 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。 `pxeveryysel` 宏包取消了检查, 但也需要恢复定义, 避免重复使用钩子。

```

1973 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1974 {
1975   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1976   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1977 }
1978 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1979 \if_cs_exist:N \@EverySelectfont@Init
1980 \group_begin:
1981   \cs_set:Npn \__ctex_tmp:N #1
1982   {
1983     \tl_set:Nn \l__ctex_tmp_tl {#1}
1984     \cs_new_eq:NN \CTEX@selectfont@save #1
1985     \cs_new_protected_nopar:Npn \__ctex_restore_selectfont:
1986     {
1987       \tl_put_left:Nn \@EverySelectfont@Init
1988       { \let #1 \CTEX@selectfont@save }
1989       \cs_undefine:N \__ctex_restore_selectfont:
1990     }
1991   }
1992   \ctex_parse_name:NN \__ctex_tmp:N \selectfont
1993   \exp_last_unbraced:NNo \group_end:
1994   \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
1995   { \ExplSyntaxOff }
1996   { \size@update }
1997   { \CTEX@selectfont@hook \size@update }
1998   { \__ctex_restore_selectfont: }
1999   { \ctex_patch_failure:N \selectfont }
2000 \fi:

```

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

```

2001 <*pdfTeX>
2002 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
2003 </pdfTeX>
2004 <*xetex|luatex|uptex|aptex>
2005 \EverySelectfont { \CTEX@selectfont@hook }
2006 </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。

```

2007 <*luatex>
2008 \ctex_add_to_selectfont:n
2009 {
2010   \ctex_ltj_select_font:
2011   \ctex_ltj_select_alternate_font:
2012 }
2013 \tl_set:Nn \CJK@family { song } \selectfont
2014 \tl_clear:N \CJK@family
2015 </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。

```

2016 <*luatex|uptex|aptex>

```



```

2017 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
2018 {
2019   \skip_if_eq:nnT
<luatex> 2020   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
<uptex|aptex> 2021   { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2022   {
2023     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
<luatex> 2024     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
<uptex|aptex> 2025     \skip_set_eq:NN \ptex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2026   }
2027 }
2028 \tl_new:N \l__ctex_xkanjiskip_tl
2029 \tl_set:Nn \l__ctex_xkanjiskip_tl
<luatex> 2030 { .25\zw plus 1pt minus 1pt }
<uptex|aptex> 2031 { .25zw plus 1pt minus 1pt }
2032 \skip_new:N \l__ctex_xkanjiskip_skip
2033 \skip_set:Nn \l__ctex_xkanjiskip_skip
<luatex> 2034 { \ltjgetparameter { xkanjiskip } }
<uptex|aptex> 2035 { \ptex_xkanjiskip:D }

2036 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2037 </luatex|uptex|aptex>

```

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

```

\cdp
\cwg
2038 <*luatex>
2039 \dim_new:N \cht
2040 \dim_new:N \cdp
2041 \dim_new:N \cwg
2042 \newluafunction \g__ctex_kanjisize_func
2043 \group_begin:
2044 \char_set_catcode_space:n { 32 }
2045 \lua_now_x:n
2046 {
2047   local nulltable = { }
2048   local t = lua.get_functions_table()
2049   local fmt = luatexja.jfont.font_metric_table
2050   local getattribute = tex.getattribute
2051   local setdimen = tex.setdimen
2052   t[\int_use:N \g__ctex_kanjisize_func] = function ()
2053     local ft = fmt[getattribute('ltj@curjfont')] or nulltable
2054     local ft = ft and ft.char_type or nulltable
2055     local fk = ft and ft[0] or nulltable
2056     setdimen('cht', fk.height or 0)
2057     setdimen('cdp', fk.depth or 0)
2058     setdimen('cwg', fk.width or ft.zw or 0)
2059   end
2060 }
2061 \group_end:
2062 \cs_new_protected_nopar:Npn \ctex_update_kanjisize:
2063 { \luafunction:D \g__ctex_kanjisize_func }
2064 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2065 </luatex>

```

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

```

2066 \keys_define:nn { ctex }
2067 {
2068   <*pdf|xtex>
2069     space .choice: ,
2070     space / true .code:n =
<pdf|xtex> 2071     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<xtex> 2072     { \xeCJKsetup { CJKspace = true } } ,
2073     space / auto .code:n =
<pdf|xtex> 2074     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
<xtex> 2075     { \xeCJKsetup { CJKspace = false } } ,

```

```

2076     space / false .code:n =
<pdf> 2077     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
<xetex> 2078     { \xeCJKsetup { CJKspace = false } } ,
2079     space .default:n = { true } ,
2080     space .initial:n = { auto }
2081 </pdf><xetex>
2082 <*luatex|uptex|aptex>
2083     space .code:n =
2084     { \msg_warning:nn { ctex } { invalid-option } }
2085 </luatex|uptex|aptex>
2086 }

```

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

```

2087 \keys_define:nn { ctex }
2088 {
2089     punct .code:n =
2090     {
2091         \tl_set:Nx \l__ctex_punct_tl { #1 }
<pdf> 2092         \punctstyle { \l__ctex_punct_tl }
<xetex> 2093         \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
<luatex> 2094         \ctex_mono_jfm:o { \l__ctex_punct_tl }
<uptex|aptex> 2095         \msg_warning:nn { ctex } { invalid-option }
2096     } ,
2097     punct .default:n = { quanjiao } ,
2098 }

```

X_YL_AT_EX、LuaL_AT_EX 和 upL_AT_EX 总是使用 UTF8 编码。

```

2099 <*xetex|luatex|uptex|aptex>
2100 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2101 </xetex|luatex|uptex|aptex>
2102 </pdf><xetex|luatex|uptex|aptex>

```

14.3.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

<class|ctex> 2103 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.4 用户设置接口

```

<ctexset> 2104 <*class|ctex|ctexheading>
2105 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
2106 </class|ctex|ctexheading>

```

\CTEXsetup 过时命令。出于历史原因,\CTEXoptions 需要在 p_xe_ver_ysel 宏包之后定义。

```

<CTEXoptions> 2107 <*class|ctex>
2108 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
2109 {
2110     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
2111     { \ctexset~ {~ #2~ } = {~ #1~ }~ }~ is~ set. }
2112     \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
2113 }
2114 \NewDocumentCommand \CTEXoptions { +o }
2115 {
2116     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
2117     { \ctexset~ {~ #1~ }~ is~ set. }
2118     \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
2119 }

```

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。

```

2120 \keys_define:nn { ctex }
2121 {
2122   autoindent .choice: ,
2123   autoindent .default:n = { true } ,
2124   autoindent / true .code:n =
2125   {
2126     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2127     \ctex_select_size:
2128   } ,
2129   autoindent / false .code:n =
2130   { \tl_clear:N \l__ctex_autoindent_tl } ,
2131   autoindent / unknown .code:n =
2132   {
2133     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2134     \ctex_select_size:
2135   }
2136 }

```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2137 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2138 \cs_new_protected_nopar:Npn \ctex_select_size:
2139 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

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

```

2140 \cs_new_protected_nopar:Npn \ctex_update_size:
2141 {
2142   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2143   {
2144     \ctex_update_stretch:
2145     \ctex_update_parindent:
2146   }
2147   { \ctex_update_ziju: }
2148 }
2149 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2150 \tl_new:N \l__ctex_ziju_tl
2151 \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。

```

2152 \ctex_add_to_selectfont:n
2153 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

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

```

2154 \keys_define:nn { ctex }
2155 {
2156   linestretch .code:n =
2157   {
2158     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2159     \ctex_select_size:
2160   } ,
2161   linestretch .value_required:n = true
2162 }
2163 \tl_new:N \l__ctex_line_stretch_tl
2164 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

`\ctex_update_stretch:` 首先计算一行上汉字的字数, `\CJKglue` 相当于将 `\linewidth` 与汉字总宽度之差均匀地填充到汉字之间。 ϵ - \TeX 的除法是四舍五入, 而我们这里应该用截断。由于没有可展性的要求, 直接用原语 `\tex_divide:D` 要比 `\int_div_truncate:nn` 快一些。下面的算法还兼顾到了 `\linewidth` 不为汉字字宽的整数倍的情况。若用户禁用 `linestretch` 并且修改过 `\CJKglue`, 则只更新 `\ccwd`, 否则设置伸展量为 0.08 倍 `\baselineskip`。注意 `everysel` 的钩子位于 `\size@update` 之前, `\baselineskip` 还未更新, 不能直接使用它。

```

2165 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2166 {
2167   \ctex_update_em_unit:
2168   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2169   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2170   {
2171     \ctex_if_ccglue_touched:TF
2172     { \ctex_update_ccwd: }
2173     {
2174       \dim_set:Nn \l__ctex_tmp_dim
2175       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2176       \skip_set:Nn \l__ctex_ccglue_skip
2177       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2178       \ctex_update_ccglue:
2179     }
2180   }
2181   {
2182     \int_set:Nn \l__ctex_tmp_int
2183     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2184     \tex_divide:D \l__ctex_tmp_int \ccwd
2185     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2186     {
2187       \skip_set:Nn \l__ctex_ccglue_skip
2188       {
2189         \c_zero_dim plus \dim_eval:n
2190         {
2191           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2192           \l__ctex_tmp_int
2193         }
2194       }
2195     }
2196     { \skip_zero:N \l__ctex_ccglue_skip }
2197     \ctex_update_ccglue:
2198   }
2199 }

```

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

```

2200 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2201 {
2202   \tl_if_empty:NF \l__ctex_autoindent_tl
2203   {
2204     \dim_compare:nNnF \parindent = \c_zero_dim
2205     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2206   }
2207 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2208 \NewDocumentCommand \ziju { m }
2209 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2210 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2211 {
2212   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2213   \ctex_select_size:
2214 }

```

`\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。

```
2215 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2216 {
2217   \ctex_update_em_unit:
2218   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2219   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2220   \dim_compare:nNnTF \ccwd > \c_zero_dim
```

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

```
2221 {
2222   \dim_set:Nn \l__ctex_tmp_dim
2223     { \linewidth - \ccwd + \l__ctex_ziju_dim }
2224   \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2225   \tex_divide:D \l__ctex_tmp_int \ccwd
2226   \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。

```
2227   \dim_compare:nNnF \parindent = \c_zero_dim
2228   {
2229     \int_compare:nNnF \l__ctex_tmp_int < \c_three
2230     { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2231   }
2232   \skip_set:Nn \l__ctex_ccglue_skip
2233   {
2234     \l__ctex_ziju_dim
2235     plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2236     minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2237     { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2238   }
2239 }
2240 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2241 \ctex_update_ccglue:
```

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

```
2242   \ctex_update_parindent:
2243 }
2244 \dim_new:N \l__ctex_ziju_dim
```

`\CTEXindent`
`\CTEXnoindent`

过时命令。

```
2245 \NewDocumentCommand \CTEXindent { }
2246 {
2247   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2248   { \parindent is~ set~ to~ 2\ccwd. }
2249   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2250 }
2251 \NewDocumentCommand \CTEXnoindent { }
2252 {
2253   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2254   { \parindent is~ set~ to~ 0pt. }
2255   \dim_zero:N \parindent
2256 }
```

14.6 中文数字与日期

```
2257 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2258 \RequirePackage { zhnumber }
```

```

\chinese 2259 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2260 \cs_new_eq:NN \@chinese \@zhnum
2261 \cs_new_eq:NN \Chinese \chinese
2262 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

```

2263 \ctex_at_end_package:nn { enumitem }
2264 {
2265   \cs_if_free:NF \AddEnumerateCounter
2266   {
2267     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2268     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2269     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2270   }
2271 }

\CTEXnumber 2272 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 2273 { \protected@edef #1 { \zhnumber {#2} } }
2274 \NewDocumentCommand \CTEXdigits { m m }
2275 { \protected@edef #1 { \zhdigits {#2} } }

today 2276 \cs_set_eq:NN \CTEX@todayold \today
2277 \keys_define:nn { ctex }
2278 {
2279   today .choice: ,
2280   today / old .code:n =
2281     { \cs_set_eq:NN \today \CTEX@todayold } ,
2282   today / small .code:n =
2283     {
2284       \cs_set_eq:NN \today \zhtoday
2285       \zhnumsetup { time = Arabic }
2286     } ,
2287   today / big .code:n =
2288     {
2289       \cs_set_eq:NN \today \zhtoday
2290       \zhnumsetup { time = Chinese }
2291     } ,
2292   today / unknown .code:n =
2293     { \msg_error:nxx { ctex } { today-undef } {#1} }
2294 }
2295 \msg_new:nnnn { ctex } { today-undef }
2296 { Today~format~`#1'~is~undefined. }
2297 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

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。

```

2298 \tl_if_exist:NF \proofname
2299 {
2300   \tl_new:N \proofname
2301   \tl_set:Nn \proofname { Proof }
2302 }

2303 \keys_define:nn { ctex }
2304 {
2305   contentsname .tl_set:N = \contentsname ,
2306   listfigurename .tl_set:N = \listfigurename ,
2307   listtablename .tl_set:N = \listtablename ,
2308   figurename .tl_set:N = \figurename ,
2309   tablename .tl_set:N = \tablename ,
2310   abstractname .tl_set:N = \abstractname ,
2311   indexname .tl_set:N = \indexname ,
2312   appendixname .tl_set:N = \appendixname ,
2313   proofname .tl_set:N = \proofname ,
2314   bibname .tl_set:N = \refname

```

```

2315 bibname .tl_set:N = \bibname
2316 <*beamer>
2317 algorithmname .tl_set:N = \algorithmname ,
2318 bibname .tl_set:N = \bibname ,
2319 refname .tl_set:N = \refname ,
2320 continuation .tl_set:N = \insertcontinuationtext
2321 </beamer>
2322 }

2323 <*ctex>
2324 \msg_new:nnn { ctex } { ctexbibname }
2325 {
2326   Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.\\
2327   The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2328 }
2329 \tl_if_exist:NTF \insertcontinuationtext
2330 {
2331   \keys_define:nn { ctex }
2332   {
2333     algorithmname .tl_set:N = \algorithmname ,
2334     bibname .tl_set:N = \bibname ,
2335     refname .tl_set:N = \refname ,
2336     continuation .tl_set:N = \insertcontinuationtext
2337   }
2338 }
2339 {
2340   \tl_if_exist:NTF \bibname
2341   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2342   {
2343     \tl_if_exist:NTF \refname
2344     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2345     {
2346       \msg_warning:nn { ctex } { ctexbibname }
2347       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2348     }
2349   }
2350 }
2351 </ctex>
2352 </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。

```
2353 <*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。

```

2354 <*article|book|report>
2355 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
2356 { section , subsection , subsubsection , paragraph , subparagraph }
2357 </article|book|report>

\c__ctex_headings_seq 2358 <*article|book|report>
2359 \seq_new:N \c__ctex_headings_seq
2360 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2361 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2362 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2363 </article|book|report>
2364 <*beamer>
2365 \seq_const_from_clist:Nn \c__ctex_headings_seq
2366 { part , section , subsection }
2367 </beamer>

```

```

__ctex_initial_heading:n 2368 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2369 {
2370   \tl_new:c { CTEX@pre#1 }
2371   \tl_new:c { CTEX@post#1 }
2372   \tl_const:cx { CTEXthe#1 }
2373   {
2374     \exp_not:c { CTEX@pre#1 }
2375     \exp_not:c { CTEX@the#1 }
2376     \exp_not:c { CTEX@post#1 }
2377   }
2378   \tl_const:cx { CTEX@#1name }
2379   {
2380     \group_begin:
2381     \exp_not:c { CTEX@#1nameformat }
2382     {
2383       \exp_not:c { CTEX@pre#1 }
2384       \exp_not:N \tl_if_empty:NTF
2385       \exp_not:c { CTEX@#1numberformat }
2386       { \exp_not:c { CTEX@the#1 } }
2387       {
2388         \group_begin:
2389         \exp_not:c { CTEX@#1numberformat }
2390         \exp_not:c { CTEX@the#1 }
2391         \group_end:
2392       }
2393       \exp_not:c { CTEX@post#1 }
2394     }
2395   }
2396 }
2397 }

__ctex_def_heading_keys:n 2398 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2399 {
2400   \tl_put_right:Nx \l__ctex_tmp_tl
2401   {
2402     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2403     #1 / name .code:n =
2404     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2405     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2406     #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1beforekip } ,
2407     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1afterskip } ,
2408     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1indent } ,
2409     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1numbering } ,
2410     #1 / numbering .initial:n = true ,
2411     #1 / beforekip .initial:n = \c_zero_skip ,
2412     #1 / afterskip .initial:n = \c_zero_skip ,
2413     #1 / indent .initial:n = \c_zero_dim ,
2414     #1 / beforekip .value_required:n = true ,
2415     #1 / afterskip .value_required:n = true ,
2416     #1 / indent .value_required:n = true ,
2417     < *article|book|report >
2418     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1afterindent } ,
2419     #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1fixskip } ,
2420     #1 / hang .bool_set:N = \exp_not:c { CTEX@#1hang } ,
2421     #1 / hang .initial:n = true ,
2422     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1runin } ,
2423     #1 / tocline .code:n =
2424     {
2425       \cs_set:Npn \exp_not:c { CTEX@#1tocline }
2426       \exp_not:n { #####1#####2 } { #####1 }
2427     } ,
2428     \__ctex_plus_key_aux:nn {#1} { break } ,
2429     < /article|book|report >
2430     \__ctex_plus_key_aux:nn {#1} { format } ,
2431     \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2432     \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2433     \__ctex_plus_key_aux:nn {#1} { titleformat } ,

```



```

2434     \__ctex_plus_key_aux:nn {#1} { aftername } ,
2435     \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2436   }
2437 }
2438 \cs_new_nopar:Npn \__ctex_plus_key_aux:nn #1#2
2439 {
2440   #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2441   #1 / #2 + .code:n =
2442   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2443   #1 / #2 ~ + .code:n =
2444   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2445 }

```

```

\ctex_assign_heading_name:nn
\__ctex_assign_heading_name:nnn

```

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

```

2446 \NewDocumentCommand \ctex_assign_heading_name:nn
2447 { m > { \SplitArgument { \c_one } { , } } +m }
2448 { \__ctex_assign_heading_name:nnn {#1} #2 }
2449 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2450 {
2451   \tl_set:cn { CTEX@pre#1 } {#2}
2452   \IfNoValueTF {#3}
2453   { \tl_clear:c { CTEX@post#1 } }
2454   { \tl_set:cn { CTEX@post#1 } {#3} }
2455 }

```

```

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

```

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2456 \group_begin:
2457 <*book|report>
2458 \tl_set:Nn \l__ctex_tmp_tl
2459 {
2460   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2461   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2462   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2463   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2464   chapter / lofskip .initial:n = \c_zero_skip ,
2465   chapter / lotskip .initial:n = \c_zero_skip ,
2466   chapter / lofskip .value_required:n = true ,
2467   chapter / lotskip .value_required:n = true ,
2468 }
2469 </book|report>
2470 <*article|beamer>
2471 \tl_clear:N \l__ctex_tmp_tl
2472 </article|beamer>

```

定义标题键值选项。

```

2473 \seq_map_inline:Nn \c__ctex_headings_seq
2474 {
2475   \__ctex_initial_heading:n {#1}
2476   \__ctex_def_heading_keys:n {#1}
2477 }
2478 \use:x
2479 {
2480   \group_end:
2481   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2482 }
2483 <*article|book|report>

```

14.8.2 标准标题命令的修改

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

```

2484 <*book|report>
2485 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2486 {
2487   \CTEX@fixheadingskip
2488   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2489     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2490 }
2491 </book|report>

```

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

```

2492 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2493 {
2494   \par
2495   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2496   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2497 }
2498 \skip_new:N \l__ctex_heading_skip
2499 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2500 { \skip_set:Nn \l__ctex_heading_skip }
2501 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\partmark` 提供 `\partmark`。

```

2502 \ProvideDocumentCommand \partmark { m }
2503 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2504 \cs_new_eq:NN \CTEXifname \use_ii:nn
2505 \cs_new_protected_nopar:Npn \CTEX@ifnametrue
2506 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2507 \cs_new_protected_nopar:Npn \CTEX@ifnamefalse
2508 { \cs_set_eq:NN \CTEXifname \use_ii:nn }

```

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

```

2509 <*book|report>
2510 \cs_new_protected_nopar:Npn \CTEX@addloflotskip #1
2511 {
2512   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2513   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2514   {
2515     \addtocontents { lof }
2516     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2517   }
2518   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2519   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2520   {
2521     \addtocontents { lot }
2522     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2523   }
2524 }
2525 </book|report>

```

```

\CTEX@addtocline
2526 \cs_new_protected:Npn \CTEX@addtocline #1#2
2527 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

2528 <@@=>
\part 2529 <*article>
2530 \renewcommand\part{%
2531 \ifnoskipsec \leavevmode \fi
2532 \par
2533 \CTEX@part@break
2534 % \addvspace{4ex}%
2535 \CTEX@setheadingskip \CTEX@part@beforeskip
2536 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2537 \addvspace \CTEX@headingskip
2538 \ifodd \CTEX@part@afterindent
2539 \@afterindenttrue
2540 \else
2541 \@afterindentfalse
2542 \fi
2543 \secdef\@part\@spart}
2544 </article>
2545 <*book|report>
2546 \renewcommand\part{%
2547 % \if@openright
2548 % \cleardoublepage
2549 % \else
2550 % \clearpage
2551 % \fi
2552 \CTEX@part@break
2553 % \thispagestyle{plain}%
2554 \thispagestyle{\CTEX@part@pagestyle}%
2555 \if@twocolumn
2556 \onecolumn
2557 \@tempwattrue
2558 \else
2559 \@tempwafalse
2560 \fi
2561 % \null\vfil
2562 \CTEX@setheadingskip \CTEX@part@beforeskip
2563 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2564 \vspace*{\CTEX@headingskip}%
2565 \secdef\@part\@spart}
2566 </book|report>

\@part 2567 <*article>
2568 \def\@part[#1]#2{%
2569 \ifnum \c@secnumdepth >\m@ne
2570 \ifodd \CTEX@part@numbering
2571 \CTEX@ifnametrue
2572 \refstepcounter{part}%
2573 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2574 \else
2575 \CTEX@ifnamefalse
2576 \CTEX@makeanchor{part*}%
2577 % \addcontentsline{toc}{part}{#1}%
2578 \fi
2579 \else
2580 \CTEX@ifnamefalse
2581 \CTEX@makeanchor{part*}%
2582 % \addcontentsline{toc}{part}{#1}%
2583 \fi
2584 \CTEX@addtocline{part}{#1}%
2585 {\interlinepenalty \@M
2586 % \normalfont \parindent \z@ \raggedright
2587 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2588 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2589 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2590 \CTEX@partname \CTEX@part@aftername
2591 \fi \fi

```

```

2592 % \huge\bfseries #2%
2593 \CTEX@part@titleformat{#2}%
2594 % \markboth{}{}%
2595 \partmark{#1}%
2596 \CTEX@part@aftertitle}%
2597 \nobreak
2598 % \vskip 3ex
2599 \CTEX@setheadingskip \CTEX@part@afterskip
2600 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2601 \vskip \CTEX@headingskip
2602 \@afterheading}
2603 </article>
2604 <*book|report>
2605 \def\@part[#1]#2{%
2606 \ifnum \c@secnumdepth >-2\relax
2607 \ifodd \CTEX@part@numbering
2608 \CTEX@ifnametrue
2609 \refstepcounter{part}%
2610 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2611 \else
2612 \CTEX@ifnamefalse
2613 \CTEX@makeanchor{part*}%
2614 % \addcontentsline{toc}{part}{#1}%
2615 \fi
2616 \else
2617 \CTEX@ifnamefalse
2618 \CTEX@makeanchor{part*}%
2619 % \addcontentsline{toc}{part}{#1}%
2620 \fi
2621 \CTEX@addtoctline{part}{#1}%
2622 % \markboth{}{}%
2623 \partmark{#1}%
2624 {\interlinepenalty \@M
2625 % \normalfont \centering
2626 \normalfont \CTEX@part@format
2627 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2628 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2629 \CTEX@partname \CTEX@part@afterskip
2630 \fi \fi
2631 % \Huge\bfseries #2\par}%
2632 \CTEX@part@titleformat{#2}%
2633 \CTEX@part@aftertitle}%
2634 \@endpart}
2635 </book|report>

\@spart 2636 <*article>
2637 \def\@spart#1{%
2638 \CTEX@ifnamefalse
2639 \CTEX@makeanchor@spart{part*}%
2640 {\interlinepenalty \@M
2641 % \normalfont \parindent \z@ \raggedright
2642 \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2643 \CTEX@part@format
2644 % \huge \bfseries #1\par}%
2645 \CTEX@part@titleformat{#1}%
2646 \CTEX@part@aftertitle}%
2647 \nobreak
2648 % \vskip 3ex
2649 \CTEX@setheadingskip \CTEX@part@afterskip
2650 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2651 \vskip \CTEX@headingskip
2652 \@afterheading}
2653 </article>
2654 <*book|report>
2655 \def\@spart#1{%
2656 \CTEX@ifnamefalse
2657 \CTEX@makeanchor@spart{part*}%

```

```

2658      {\interlinepenalty \@M
2659 %      \normalfont \centering
2660      \normalfont \CTEX@part@format
2661 %      \Huge \bfseries #1\par}%
2662      \CTEX@part@titleformat{#1}%
2663      \CTEX@part@aftertitle}%
2664      \@endpart}
2665 </book|report>

\@endpart 2666 <*book|report>
2667 \def\@endpart{%
2668 %      \vfil
2669      \CTEX@setheadingskip \CTEX@part@afterskip
2670      \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2671      \vskip \CTEX@headingskip
2672      \newpage
2673      \if@twoside
2674      \if@openright
2675      \null
2676      \thispagestyle{empty}%
2677      \newpage
2678      \fi
2679      \fi
2680      \if@tempwa
2681      \twocolumn
2682      \fi}
2683 </book|report>

14.8.2.2 chapter 的标题

2684 <*book|report>

\chapter 2685 \renewcommand\chapter{%
2686 %      \if@openright\cleardoublepage\else\clearpage\fi
2687 %      \thispagestyle{plain}%
2688      \CTEX@chapter@break
2689      \thispagestyle{\CTEX@chapter@pagestyle}%
2690      \global\@topnum\z@
2691 %      \@afterindentfalse
2692      \ifodd \CTEX@chapter@afterindent
2693      \@afterindenttrue
2694      \else
2695      \@afterindentfalse
2696      \fi
2697      \secdef\@chapter\@schapter}

\@chapter 2698 \def\@chapter[#1]#2{%
2699   \ifnum \c@secnumdepth >\m@ne
2700   <*book>
2701   \if@mainmatter
2702   </book>
2703   \ifodd \CTEX@chapter@numbering
2704   \CTEX@ifnametrue
2705   \refstepcounter{chapter}%
2706 %   \typeout{\@chapapp\space\thechapter.}%
2707   \typeout{\CTEXthechapter}%
2708 %   \addcontentsline{toc}{chapter}
2709 %   {\protect\numberline{\thechapter}#1}%
2710   \else
2711   \CTEX@ifnamefalse
2712   \CTEX@makeanchor{\Hy@chapapp*}%
2713 %   \addcontentsline{toc}{chapter}{#1}%
2714   \fi
2715   <*book>
2716   \else
2717   \CTEX@ifnamefalse
2718   \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2719 %   \addcontentsline{toc}{chapter}{#1}%

```

```

2720     \fi
2721 </book>
2722 \else
2723     \CTEX@ifnamefalse
2724     \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2725 % \addcontentsline{toc}{chapter}{#1}%
2726 \fi
2727 \CTEX@addtocline{chapter}{#1}%
2728 \chaptermark{#1}%
2729 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2730 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2731 \CTEX@addloflotskip{chapter}%
2732 \if@twocolumn
2733     \@topnewpage[\@makechapterhead{#2}]%
2734 \else
2735     \@makechapterhead{#2}%
2736 \afterheading
2737 \fi}

\@schapter 2738 \def\@schapter#1{%
2739     \CTEX@ifnamefalse
2740     \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2741     \if@twocolumn
2742         \@topnewpage[\@makeschapterhead{#1}]%
2743     \else
2744         \@makeschapterhead{#1}%
2745     \afterheading
2746 \fi}

\@makechapterhead 2747 \def\@makechapterhead#1{%
2748 % \vspace*{50\p@}%
2749 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2750 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2751 \vspace*{\CTEX@headingskip}%
2752 % {\normalfont \parindent \z@ \raggedright
2753 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2754 \CTEX@chapter@format
2755 % \ifnum \c@secnumdepth >\m@ne
2756 % \if@mainmatter
2757 % \huge \bfseries \@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2758 % \fi
2759 % \fi
2760 \CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}%
2761 \interlinepenalty\@M
2762 % \Huge \bfseries #1\par\nobreak
2763 \CTEX@chapter@titleformat{#1}%
2764 \CTEX@chapter@aftertitle
2765 \nobreak
2766 % \vskip 40\p@
2767 \CTEX@setheadingskip \CTEX@chapter@afterskip
2768 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2769 \vskip \CTEX@headingskip
2770 }}

\@makeschapterhead 2771 \def\@makeschapterhead#1{%
2772 % \vspace*{50\p@}%
2773 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2774 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2775 \vspace*{\CTEX@headingskip}%
2776 % {\normalfont \parindent \z@ \raggedright
2777 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2778 \CTEX@chapter@format
2779 \interlinepenalty\@M
2780 % \Huge \bfseries #1\par\nobreak
2781 \CTEX@chapter@titleformat{#1}
2782 \CTEX@chapter@aftertitle
2783 \nobreak

```

```

2784 % \vskip 40\p@
2785 \CTEX@setheadingskip \CTEX@chapter@afterskip
2786 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2787 \vskip \CTEX@headingskip
2788 }}
2789 </book|report>

```

14.8.2.3 section 类的标题

```

\@startsection 2790 \def\@startsection#1#2#3#4#5#6{%
2791 \if@noskipsec \leavevmode \fi
2792 \par
2793 % \@tempskipa #4\relax
2794 % \@afterindenttrue
2795 % \ifdim \@tempskipa <\z@
2796 % \@tempskipa -\@tempskipa \@afterindentfalse
2797 % \fi
2798 \CTEX@update@sectionformat@n{#1}%
2799 \ifodd \CTEX@afterindent
2800 \@afterindenttrue
2801 \else
2802 \@afterindentfalse
2803 \fi
2804 \if@nobreak
2805 \everypar{}%
2806 \else
2807 % \addpenalty\@secpenalty\addvspace\@tempskipa
2808 \csname CTEX@#1@break\endcsname
2809 \CTEX@setheadingskip{#4}%
2810 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2811 \addvspace \CTEX@headingskip
2812 \fi
2813 \@ifstar
2814 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2815 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

\@secntformat 2816 \def\@secntformat#1{%
2817 % \csname the#1\endcsname\quad}%
2818 \csname CTEX@#1name\endcsname
2819 \csname CTEX@#1@aftername\endcsname}

\@sect 2820 \def\@sect#1#2#3#4#5#6[#7]#8{%
2821 \ifnum #2>\c@secnumdepth
2822 \CTEX@ifnamefalse
2823 \CTEX@makeanchor@sect{#1*}%
2824 \let\@svsec\@empty
2825 \else
2826 \ifodd \csname CTEX@#1@numbering\endcsname
2827 \CTEX@ifnametrue
2828 \refstepcounter{#1}%
2829 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2830 \else
2831 \CTEX@ifnamefalse
2832 \CTEX@makeanchor{#1*}%
2833 \let\@svsec\@empty
2834 \fi
2835 \fi
2836 % \@tempskipa #5\relax
2837 % \ifdim \@tempskipa>\z@
2838 \unless \ifodd \CTEX@runin
2839 \begingroup
2840 #6{%
2841 \CTEX@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2842 % \interlinepenalty \@M #8\@@par}%
2843 \interlinepenalty \@M
2844 \csname CTEX@#1@titleformat\endcsname{#8}%
2845 \csname CTEX@#1@aftertitle\endcsname}%

```

```

2846 \endgroup
2847 \csname #1mark\endcsname{#7}%
2848 % \addcontentsline{toc}{#1}{%
2849 % \ifnum #2>\c@secnumdepth \else
2850 % \protect\numberline{\csname the#1\endcsname}%
2851 % \fi
2852 % #7}%
2853 \CTEX@addtocline{#1}{#7}%
2854 \else
2855 \def\@svsechd{%
2856 #6{\hskip\glueexpr #3\relax
2857 % \@svsec #8}%
2858 \@svsec
2859 \csname CTEX@#1@titleformat\endcsname{#8}%
2860 \csname CTEX@#1@aftertitle\endcsname}%
2861 \csname #1mark\endcsname{#7}%
2862 % \addcontentsline{toc}{#1}{%
2863 % \ifnum #2>\c@secnumdepth \else
2864 % \protect\numberline{\csname the#1\endcsname}%
2865 % \fi
2866 % #7}%
2867 \CTEX@addtocline{#1}{#7}}%
2868 \fi
2869 \@xsect{#5}}

\@sssect 2870 \def\@sssect#1#2#3#4#5{%
2871 \CTEX@ifnamefalse
2872 % \@tempskipa #3\relax
2873 % \ifdim \@tempskipa>\z@
2874 \unless \ifodd \CTEX@runin
2875 \begingroup
2876 #4{%
2877 \CTEX@hangfrom{\hskip\glueexpr #1\relax}%
2878 % \interlinepenalty \@M #5\@par}%
2879 \interlinepenalty \@M
2880 \CTEX@titleformatn{#5}%
2881 \CTEX@aftertitle}%
2882 \endgroup
2883 \else
2884 % \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2885 \def\@svsechd{#4{\hskip\glueexpr #1\relax
2886 \CTEX@titleformatn{#5}\CTEX@aftertitle}}%
2887 \fi
2888 \@xsect{#3}}

\@xsect 2889 \def\@xsect#1{%
2890 % \@tempskipa #1\relax
2891 % \ifdim \@tempskipa>\z@
2892 \unless \ifodd \CTEX@runin
2893 \par \nobreak
2894 % \vskip \@tempskipa
2895 \CTEX@setheadingskip{#1}%
2896 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2897 \vskip \CTEX@headingskip
2898 \@afterheading
2899 \else
2900 \@nobreakfalse
2901 \global\@noskipsectrue
2902 \everypar{%
2903 \if@noskipsec
2904 \global\@noskipsecfalse
2905 {\setbox\z@\lastbox}%
2906 \clubpenalty\@M
2907 \begingroup \@svsechd \endgroup
2908 \unskip
2909 % \@tempskipa #1\relax
2910 % \hskip -\@tempskipa

```



```

2911         \hskip\glueexpr #1\relax
2912     \else
2913         \clubpenalty \@clubpenalty
2914         \everypar{}\%
2915     \fi}%
2916 \fi
2917 \ignorespaces}

2918 <@@=ctex>

```

\CTEX@hangfrom **hang** 选项控制是否采用悬挂缩进。

```

2919 \cs_new_protected_nopar:Npn \CTEX@hangfrom
2920 {
2921     \bool_if:NTF \CTEX@hang
2922     { \@hangfrom }
2923     { \noindent \use:n }
2924 }

```

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

```

2925 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2926 {
2927     \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2928     \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2929     \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2930     \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2931     \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2932     \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2933 }
2934 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2935 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2936 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2937 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2938 \cs_new_eq:NN \CTEX@hang \c_true_bool
2939 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

\CTEX@part@tocline
\CTEX@chapter@tocline

```

2940 \cs_new:Npn \CTEX@part@tocline #1#2
2941 {
2942     \CTEXifname
2943     { \CTEXthepart \hspace { 1em } }
2944     { }
2945     #2
2946 }
2947 <*book|report>
2948 \cs_new:Npn \CTEX@chapter@tocline #1#2
2949 {
2950     \CTEXifname
2951     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2952     { }
2953     #2
2954 }
2955 </book|report>

```

\CTEXnumberline

```

2956 \cs_new_nopar:Npn \CTEXnumberline #1
2957 {
2958     \CTEXifname
2959     { \protect \numberline { \use:c { CTEXthe #1 } } }
2960     { }
2961 }

2962 \int_zero:N \l__ctex_tmp_int
2963 \seq_map_inline:Nn \c__ctex_section_headings_seq
2964 {
2965     \int_incr:N \l__ctex_tmp_int
2966     \cs_gset_protected_nopar:cpx {#1}
2967     {
2968         \exp_not:N \@startsection {#1}

```

```

2969 { \int_use:N \l__ctex_tmp_int }
2970 { \exp_not:c { CTEX@#1@indent } }
2971 { \exp_not:c { CTEX@#1@beforeskip } }
2972 { \exp_not:c { CTEX@#1@afterskip } }
2973 { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2974 }
2975 \cs_new:cpn { CTEX@#1@tocline } ##1##2
2976 { \CTEXnumberline { ##1 } ##2 }
2977 }

```

14.8.2.4 附录标题

```

appendix/name 2978 \keys_define:nn { ctex }
appendix/number 2979 {
appendix/numbering 2980   appendix .meta:nn = { ctex / appendix } {#1} ,
2981   appendix / name .code:n =
2982   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2983   appendix / number .tl_set:N = \CTEX@appendix@number ,
2984   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2985   appendix / numbering .initial:n = true
2986 }
2987 \tl_new:N \CTEX@preappendix
2988 \tl_new:N \CTEX@postappendix

\appendix 2989 \cs_new_eq:NN \CTEX@save@appendix \appendix
2990 \cs_gset_protected_nopar:Npn \appendix
2991 {
2992   \CTEX@save@appendix
2993   < *article >
2994   \gdef \CTEX@presection { \CTEX@preappendix }
2995   \gdef \CTEX@thesection { \CTEX@appendix@number }
2996   \gdef \CTEX@postsection { \CTEX@postappendix }
2997   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2998   < /article >
2999   < *book|report >
3000   \gdef \CTEX@prechapter { \CTEX@preappendix }
3001   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3002   \gdef \CTEX@postchapter { \CTEX@postappendix }
3003   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3004   < /book|report >
3005 }

```

14.8.2.5 设置 hyperref 宏包的标题锚点

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

```

3006 \cs_new_protected_nopar:Npn \CTEX@makeanchor #1
3007 { }

```

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

```

3008 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
< article > 3009 { part , spart , sect , ssect }
< book|report > 3010 { part , spart , chapter , schapter , sect , ssect }
3011 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3012 {
3013   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3014   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3015 }

```

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

```

3016 \cs_new_protected_nopar:Npn \CTEX@hyperheadinghook

```

```

3017 {
3018   \group_begin:
3019   \ifHy@implicit
3020     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3021     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3022     {
3023       \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3024       {
3025         \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3026         \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3027       }
3028     }
3029   \else:
3030     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3031     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3032   \fi:
3033   \group_end:
3034 }

3035 \ctex_at_end_package:nn { hyperref }
3036 {
3037   \cs_gset_protected_nopar:Npn \CTEX@makeanchor #1
3038   {
3039     \Hy@MakeCurrentHrefAuto {#1}
3040     \Hy@raisedlink
3041     {
3042       \hyper@anchorstart { \@currentHref }
3043       \hyper@anchorend
3044     }
3045   }
3046   \CTEX@hyperheadinghook
3047 }

```

14.8.2.6 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 `\ttl@<section>` 的宏中备用, 它的内容是

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

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

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

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

```

3048 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
3049 {
3050   \ifpackagewith { titlesec } { explicit }
3051   {
3052     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3053     \__ctex_titlesec_format_explicit:Nn

```

```

3054     }
3055     { }
3056     \clist_map_inline:nn
3057     { indentafter , noindentafter , indentfirst , nonindentfirst }
3058     {
3059         \@ifpackagewith { titlesec } { ##1 }
3060         {
3061             \clist_map_break:n
3062             { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3063         }
3064     { }
3065     }
3066     \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3067 }
3068 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
3069 {
3070     \__ctex_titlesec_format:n {#1}
3071     \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3072 }
3073 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
3074 {
3075     \cs_if_free:cF { ttlf@#1 }
3076     { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3077 }
3078 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
3079 {
3080     \tl_set:Nx #1
3081     {
3082         \bool_if:cTF { CTEX@#2@runin }
3083         { \exp_not:N \ttlh@runin }
3084         { \exp_not:N \ttlh@hang }
3085         \tl_tail:N #1
3086     }
3087 }
3088 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3089 {
3090     \cs_set_nopar:Npx #1 ##1
3091     {
3092         \bool_if:cTF { CTEX@#2@runin }
3093         { \exp_not:N \ttlh@runin }
3094         { \exp_not:N \ttlh@hang }
3095         \exp_args:No \tl_tail:n { #1 { } }
3096     }
3097 }
3098 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
3099 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3100 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3101 {
3102     \exp_not:n { {#1} {#2} {#3} {#4} }
3103     { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3104 }
3105 \@ifpackageloaded { titlesec }
3106 { }
3107 {
3108     \ctex_at_end_package:nn { titlesec }
3109     {
3110         \@ifpackagewith { titlesec } { loadonly }
3111         { }
3112         { \ctex_titlesec_hook: }
3113     }
3114 }

```

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

```

3115 \ctex_at_end_package:nn { titlesec }
3116 {
3117     <*book|report>

```

```

3118 \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3119 </book|report>
3120 \cs_if_free:NF \ttl@tocpart
3121 {
3122   \cs_set_protected_nopar:Npn \ttl@tocpart
3123   { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3124 }
3125 \seq_map_inline:Nn \c__ctex_headings_seq
3126 {
3127   \cs_if_exist:cF { ttl@toc #1 }
3128   {
3129     \cs_new_protected_nopar:cpx { ttl@toc #1 }
3130     {
3131       \tl_set:Nn \exp_not:N \ttl@a
3132       {
3133         \exp_not:N \protect
3134         \exp_not:N \numberline { \exp_not:c { CTExthe #1 } }
3135       }
3136     }
3137   }
3138 }
3139 }

```

14.8.2.7 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。

```

3140 \group_begin:
3141 \char_set_catcode_other:N \#
3142 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
3143 {
3144   \ctex_patch_cmd:Nnn \ttl@settopmark
3145   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3146   {
3147     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3148     \CTEX@titlepslabel@set {#1}
3149   }
3150   \ctex_patch_cmd:Nnn \ttl@setsubmark
3151   { \protect \@namedef { the#1 } { } }
3152   {
3153     \protect \@namedef { the#1 } { }
3154     \CTEX@titlepslabel@clear {#1}
3155   }
3156   \ctex_patch_cmd:Nnn \ttl@setsubmark
3157   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3158   {
3159     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3160     \CTEX@titlepslabel@set {#2}
3161   }
3162 }
3163 \group_end:

```

`\CTEX@titlepslabel@set`
`\CTEX@titlepslabel@clear` 这两个函数要在随后被 `\xdef` 展开来获得 `\CTEXtheXXX` 的内容, 不应该用 `\protected` 来定义。

```

3164 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
3165 {
3166   \cs_if_free:cF { CTExthe#1 }
3167   { \protect \@namedef { CTExthe#1 } { \@nameuse { CTExthe#1 } } }

```

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

```

3168 }
3169 \cs_new_nopar:Npn \CTEX@titleslabel@clear #1
3170 {
3171   \cs_if_free:cF { CTEXthe#1 }
3172   { \protect \@namedef { CTEXthe#1 } { } }
3173 }

```

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

```

3174 \ctex_at_end_package:nn { titlesec }
3175 { \cs_if_free:NF \ttl@settopmark { \ctex_titles_hook: } }
3176 \ctex_at_end_package:nn { titles } { \ctex_titles_hook: }

```

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

```

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

```

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

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

```

3177 \ctex_at_end_package:nn { titles }
3178 {
3179   \cs_set_protected_nopar:Npn \ttl@setifthe #1
3180   {
3181     \exp_args:Nco \cs_set_nopar:Npn { ifthe #1 }
3182     {
3183       \CTEXifname
3184       { \protect \@firstoftwo }
3185       { \protect \@secondoftwo }
3186     }
3187   }
3188   \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3189 }

```

14.8.3 目录标签的宽度

```

\numberline 3190 \cs_new_protected:Npn \CTEX@toc@width@n #1
3191 {
3192   \hbox_set:Nn \l__ctex_tmp_box {#1}
3193   \dim_set:Nn \@tempdima
3194   {
3195     \dim_max:nn { \@tempdima }
3196     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
3197   }
3198 }
3199 \group_begin:
3200 \char_set_catcode_other:N \#
3201 \use:n
3202 {
3203   \group_end:
3204   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3205   { \CTEX@toc@width@n {#1} }
3206   { }
3207   { \ctex_patch_failure:N \numberline }
3208   \@ifpackageloaded { tocloft }
3209   { }
3210   {
3211     \ctex_at_end_package:nn { tocloft }
3212     {
3213       \ctex_preto_cmd:NnnTF \numberline
3214       { \char_set_catcode_letter:n { 64 } }
3215       { \CTEX@toc@width@n {#1} }
3216       { }
3217       { \ctex_patch_failure:N \numberline }

```

```

3218     }
3219   }
3220 }

```

14.8.4 页眉信息的修改

```

\ps@headings 3221 <*article>
3222 \if@twoside
3223   \ctex_patch_cmd:Nnn \ps@headings
3224     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3225     { \CTEXifname { \CTEXthesection \quad } { } }
3226   \ctex_patch_cmd:Nnn \ps@headings
3227     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3228     { \CTEXifname { \CTEXthesubsection \quad } { } }
3229 \else:

```

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

```

3230   \ctex_patch_cmd:Nnn \ps@headings
3231     { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3232     { \CTEXifname { \CTEXthesection \quad } { } }
3233 \fi:
3234 </article>
3235 <*book|report>
3236 \ctex_patch_cmd:Nnn \ps@headings
3237   {
3238     \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3239     \ifnum \c@secnumdepth > \m@ne
3240       \@chapapp \thechapter . ~ \ %
3241     \fi
3242     \fi \fi
3243   }
3244   { \CTEXifname { \CTEXthechapter \quad } { } }
3245 \if@twoside
3246   \ctex_patch_cmd:Nnn \ps@headings
3247     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3248     { \CTEXifname { \CTEXthesection \quad } { } }
3249 \fi:
3250 </book|report>

```

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

```

3251 \ctex_at_end_package:nn { fancyhdr }
3252 {
3253 <*article>
3254   \ctex_patch_cmd:Nnn \ps@fancy
3255     { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3256     { \CTEXifname { \CTEXthesection \quad } { } }
3257   \ctex_patch_cmd:Nnn \ps@fancy
3258     { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3259     { \CTEXifname { \CTEXthesubsection \quad } { } }
3260 </article>
3261 <*book|report>
3262   \ctex_patch_cmd:Nnn \ps@fancy
3263     { \ifnum \c@secnumdepth > \m@ne \@chapapp \thechapter . ~ \ \fi }
3264     { \CTEXifname { \CTEXthechapter \quad } { } }
3265   \ctex_patch_cmd:Nnn \ps@fancy
3266     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3267     { \CTEXifname { \CTEXthesection \quad } { } }
3268 </book|report>
3269 }
3270 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

3271 <*beamer>

3272 \ExplSyntaxOff

对应 \partpage。

```

3273 \defbeamertertemplate*{part page}{CTEX}[1][]{
3274   \begingroup
3275   % \centering
3276   % {\usebeamerfont{part name}%
3277   % \usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3278   % \vskip1em\par
3279   \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3280   \parindent \dimexpr \CTEX@part@indent \relax
3281   \CTEX@part@format
3282   \ifodd \CTEX@part@numbering
3283     \CTEX@partname \CTEX@part@aftername
3284   \fi
3285   \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3286   % \usebeamerfont{part title}\insertpart\par
3287     \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3288   \end{beamercolorbox}
3289   \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3290 \endgroup
3291 }
```

对应 \sectionpage。

```

3292 \defbeamertertemplate*{section page}{CTEX}[1][]{
3293   \begingroup
3294   % \centering
3295   % {\usebeamerfont{section name}%
3296   % \usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}
3297   % \vskip1em\par
3298   \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3299   \parindent \dimexpr \CTEX@section@indent \relax
3300   \CTEX@section@format
3301   \ifodd \CTEX@section@numbering
3302     \CTEX@sectionname \CTEX@section@aftername
3303   \fi
3304   \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3305   % \usebeamerfont{section title}\insertsection\par
3306     \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3307   \end{beamercolorbox}
3308   \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3309 \endgroup
3310 }
```

对应 \subsectionpage。

```

3311 \defbeamertertemplate*{subsection page}{CTEX}[1][]{
3312   \begingroup
3313   % \centering
3314   % {\usebeamerfont{subsection name}%
3315   % \usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
3316   % \vskip1em\par
3317   \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3318   \parindent \dimexpr \CTEX@subsection@indent \relax
3319   \CTEX@subsection@format
3320   \ifodd \CTEX@subsection@numbering
3321     \CTEX@subsectionname \CTEX@subsection@aftername
3322   \fi
3323   \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3324   % \usebeamerfont{subsection title}\insertsubsection\par
3325     \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3326   \end{beamercolorbox}
3327   \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3328 \endgroup
```



```
3329 }
```

将 **beamer** 的默认模板重定向为 CTEX 模板。

```
3330 \defbeamerplatformalias{part page}{default}{CTEX}
3331 \defbeamerplatformalias{section page}{default}{CTEX}
3332 \defbeamerplatformalias{subsection page}{default}{CTEX}
3333 \ExplSyntaxOn
3334 </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。

```
3335 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
3336 {
3337   \protected@edef \@currentlabel
3338   {
3339     \cs_if_exist:cTF { CTEX@the#1 }
3340     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3341     { \exp_not:o { \@currentlabel } }
3342   }
3343 }
```

`\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。

```
3344 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3345 {
3346   \seq_map_inline:Nn \c__ctex_headings_seq
3347   { \ctex_fix_varioref_label:n { ##1 } }
3348 }
```

`__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。

```
3349 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3350 {
3351   \tl_if_empty:cT { p@#1 }
3352   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3353 }
```

如果 **varioref** 已经被载入,则使用它来设置。

```
3354 \ifpackageloaded { varioref }
3355 { \ctex_varioref_hook: }
3356 {
3357   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3358   \RenewDocumentCommand \refstepcounter { m }
3359   {
3360     \CTEX@save@refstepcounter {#1}
3361     \CTEX@setcurrentlabel@n {#1}
3362   }
3363   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3364 }
```

14.8.7 载入 $\langle scheme \rangle$ 文件

```
3365 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3366 </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 \startsection 设定的章节标题格式, chinese 汉化的标题格式。

```
3367 <*scheme&(article|book|report|beamer)>

3368 \keys_set:nn { ctex / part }
3369 {
3370     aftertitle = \par ,
3371 <*plain>
3372     name       = \partname \space ,
3373 <*article|book|report>
3374     number     = \thepart ,
3375 </article|book|report>
3376 <*beamer>
3377     number     = \insertromanpartnumber ,
3378 </beamer>
3379 </plain>
3380 <*chinese>
3381     number     = \chinese { part } ,
3382 </chinese>
3383 <*article>
3384     beforeskip = 4ex ,
3385     afterskip  = 3ex ,
3386 <*plain>
3387     format     = \raggedright ,
3388     nameformat = \Large \bfseries ,
3389     aftername   = \par \nobreak ,
3390     titleformat = \huge \bfseries ,
3391     afterindent = false
3392 </plain>
3393 <*chinese>
3394     format     = \Large \bfseries \centering ,
3395     aftername   = \quad ,
3396     afterindent = true
3397 </chinese>
3398 </article>
3399 <*book|report>
3400     aftername   = \par \vskip 20 \p@ ,
3401     beforeskip  = Opt \@plus 1fil ,
3402     afterskip   = Opt \@plus 1fil ,
3403     pagestyle   = plain ,
3404     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3405 <*plain>
3406     format     = \centering ,
3407     nameformat = \huge \bfseries ,
3408     titleformat = \Huge \bfseries
3409 </plain>
3410 <*chinese>
3411     format     = \huge \bfseries \centering
3412 </chinese>
3413 </book|report>
3414 <*beamer>
3415     format     = \centering ,
3416     nameformat = \usebeamerfont { part ~ name }
3417                 \usebeamercolor [fg] { part ~ name } ,
3418     aftername   = \vskip 1em \par ,
3419     titleformat = \usebeamerfont { part ~ title }
3420 </beamer>
3421 }
3422 <*book|report>
```

```

3423 \keys_set:nn { ctex / chapter }
3424 {
3425     pagestyle     = plain ,
3426     aftertitle    = \par ,
3427     beforeskip    = 50 \p@ ,
3428     afterskip     = 40 \p@ ,
3429     lofskip       = 10 \p@ ,
3430     lotskip       = 10 \p@ ,
3431     break         = \if@openright \cleardoublepage \else \clearpage \fi ,
3432 <*plain>
3433     name          = \chaptername \space ,
3434     number        = \thechapter ,
3435     format        = \raggedright ,
3436     nameformat    = \huge \bfseries ,
3437     aftername     = \par \nobreak \vskip 20 \p@ ,
3438     titleformat   = \Huge \bfseries ,
3439     afterindent   = false ,
3440     tocline       = \CTEXnumberline {#1} #2
3441 </plain>
3442 <*chinese>
3443     number        = \chinese { chapter } ,
3444     format        = \huge \bfseries \centering ,
3445     aftername     = \quad ,
3446     afterindent   = true
3447 </chinese>
3448 }
3449 </book|report>

3450 <@@= >

3451 \keys_set:nn { ctex / section }
3452 {
3453 <*article|book|report>
3454     number        = \thesection ,
3455     aftername     = \quad ,
3456     aftertitle    = \@@par ,
3457     beforeskip    = 3.5ex \@plus 1ex \@minus .2ex ,
3458     afterskip     = 2.3ex \@plus .2ex ,
3459     runin         = false ,
3460     break         = \addpenalty \@secpenalty ,
3461 <*plain>
3462     format        = \Large \bfseries ,
3463     afterindent   = false
3464 </plain>
3465 <*chinese>
3466     format        = \Large \bfseries \centering ,
3467     afterindent   = true
3468 </chinese>
3469 </article|book|report>
3470 <*beamer>
3471 <*plain>
3472     name          = \sectionname \space ,
3473 </plain>
3474     format        = \centering ,
3475     number        = \insertsectionnumber ,
3476     nameformat    = \usebeamerfont { section ~ name }
3477                  \usebeamerfont { section ~ title } ,
3478     aftername     = \vskip 1em \par ,
3479     titleformat   = \usebeamerfont { section ~ title } ,
3480     aftertitle    = \par
3481 </beamer>
3482 }

3483 \keys_set:nn { ctex / subsection }
3484 {
3485 <*article|book|report>
3486     number        = \thesubsection ,
3487     format        = \large \bfseries ,

```

```

3488     aftername      = \quad ,
3489     aftertitle     = \@@par ,
3490     beforeskip     = 3.25ex \@plus 1ex \@minus .2ex ,
3491     afterskip      = 1.5ex \@plus .2ex ,
3492     runin          = false ,
3493     break          = \addpenalty \@secpenalty ,
3494 <*plain>
3495     afterindent    = false
3496 </plain>
3497 <*chinese>
3498     afterindent    = true
3499 </chinese>
3500 </article|book|report>
3501 <*beamer>
3502 <*plain>
3503     name           = \subsectionname \space ,
3504     number         = \insertsubsectionnumber ,
3505 </plain>
3506 <*chinese>
3507     number         = \arabic { section } . \arabic { subsection } ,
3508 </chinese>
3509     format         = \centering ,
3510     nameformat     = \usebeamerfont { subsection ~ name }
3511                   \usebeamerfont { subsection ~ name } ,
3512     aftername      = \vskip 1em \par ,
3513     titleformat    = \usebeamerfont { subsection ~ title } ,
3514     aftertitle     = \par
3515 </beamer>
3516 }

3517 <*article|book|report>

3518 \keys_set:nn { ctex / subsubsection }
3519 {
3520     number         = \thesubsubsection ,
3521     format         = \normalsize \bfseries ,
3522     aftername      = \quad ,
3523     aftertitle     = \@@par ,
3524     beforeskip     = 3.25ex \@plus 1ex \@minus .2ex ,
3525     afterskip      = 1.5ex \@plus .2ex ,
3526     runin          = false ,
3527     break          = \addpenalty \@secpenalty ,
3528 <*plain>
3529     afterindent    = false
3530 </plain>
3531 <*chinese>
3532     afterindent    = true
3533 </chinese>
3534 }

3535 \keys_set:nn { ctex / paragraph }
3536 {
3537     number         = \theparagraph ,
3538     format         = \normalsize \bfseries ,
3539     aftername      = \quad ,
3540     beforeskip     = 3.25ex \@plus 1ex \@minus .2ex ,
3541     break          = \addpenalty \@secpenalty ,
3542 <*plain>
3543     afterindent    = false
3544 </plain>
3545 <*chinese>
3546     afterindent    = true
3547 </chinese>
3548 }

3549 \keys_set:nn { ctex / subparagraph }
3550 {
3551     number         = \thesubparagraph ,
3552     format         = \normalsize \bfseries ,

```

```

3553     aftername      = \quad ,
3554     beforeskip     = 3.25ex \@plus 1ex \@minus .2ex ,
3555     break          = \addpenalty \@secpenalty ,
3556 <*plain>
3557     afterindent    = false
3558 </plain>
3559 <*chinese>
3560     afterindent    = true
3561 </chinese>
3562 }

```

处理 sub3section 与 sub4section 的格式。

```

3563 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3564 {
3565   \keys_set:nn { ctex / paragraph }
3566   {
3567     aftertitle     = \@@par ,
3568     afterskip      = 1ex \@plus .2ex ,
3569     runin          = false
3570   }
3571 }
3572 {
3573   \keys_set:nn { ctex / paragraph }
3574   {
3575     afterskip      = 1em ,
3576     runin          = true
3577   }
3578 }
3579 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3580 {
3581   \keys_set:nn { ctex / subparagraph }
3582   {
3583     aftertitle     = \@@par ,
3584     afterskip      = 1ex \@plus .2ex ,
3585     runin          = false
3586   }
3587 }
3588 {
3589   \keys_set:nn { ctex / subparagraph }
3590   {
3591     afterskip      = 1em ,
3592     runin          = true
3593   }
3594 }
3595 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3596 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3597 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3598 <@@=ctex>

```

处理附录的格式。

```

3599 \keys_set:nn { ctex / appendix }
3600 <*article>
3601 { number          = \@Alph \c@section }
3602 </article>
3603 <*book|report>
3604 {
3605   name            = \appendixname \space ,
3606   number          = \@Alph \c@chapter
3607 }
3608 </book|report>
3609 </article|book|report>
3610 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

3611 `<ctex|ctexheading>`

`\c__ctex_std_class_tl` 用于记录被引入的标准文档类。

```
3612 \clist_map_inline:nn { article , book , report , beamer }
3613 {
3614   \@ifclassloaded {#1}
3615     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3616     { }
3617 }
```

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`。

```
3618 \msg_new:nnn { ctex } { not-standard-class }
3619 {
3620   None~of~the~standard~document~classes~was~loaded.\\
3621   Heading~`#1'~is~selected.\\
3622   ctex~may~not~work~as~expected.
3623 }
<ctex> 3624 \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 3625 \use:n
3626 {
3627   \tl_if_exist:NTF \c__ctex_std_class_tl
3628     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3629     {
3630       \cs_if_exist:NTF \chapter
3631         {
3632           \cs_if_exist:NF \if@mainmatter
3633             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3634             \tl_const:Nn \c__ctex_class_tl { book }
3635         }
3636         { \tl_const:Nn \c__ctex_class_tl { article } }
3637       \msg_warning:nx { ctex } { not-standard-class } { \c__ctex_class_tl }
3638     }
3639   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3640 }
<ctex> 3641 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3642 </ctex|ctexheading>
```

14.8.10 标题配置文件

3643 `<*name>`

```
3644 \keys_set_known:nn { ctex }
3645 {
3646   contentsname = 目录 ,
3647   listfigurename = 插图 ,
3648   listtablename = 表格 ,
3649   figurename = 图 ,
3650   tablename = 表 ,
3651   abstractname = 摘要 ,
3652   indexname = 索引 ,
3653   bibname = 参考文献 ,
3654   appendixname = 附录 ,
3655   proofname = 证明 ,
3656   algorithmname = 算法 ,
3657   refname = 参考文献 ,
3658   continuation = (续) ,
3659   part / name = { 第 , 部分 } ,
3660   chapter / name = { 第 , 章 }
3661 }
```

3662 `</name>`

14.9 chinese 方案的其他设置

3663 `<*scheme&chinese>`

chinese 在标准文档类下的页面格式总采用 headings。

`<article|book|report>` 3664 `\pagestyle { headings }`

日期格式。

3665 `\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。

3666 `\ctex_if_autoindent_touched:F`

3667 `{ \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。

3668 `<!generic>`

3669 `\str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }`

3670 `<*beamer>`

3671 `{`

3672 `\uselanguage { ChineseGBK }`

3673 `\languagealias { chinese } { ChineseGBK }`

3674 `\ctex_file_input:n { ctex-name-gbk.cfg }`

3675 `}`

3676 `{`

3677 `\uselanguage { ChineseUTF8 }`

3678 `\languagealias { chinese } { ChineseUTF8 }`

3679 `\ctex_file_input:n { ctex-name-utf8.cfg }`

3680 `}`

让 translator 包优先查找中文翻译。

3681 `\clist_put_left:Nn \trans@languagepath { chinese }`

3682 `</beamer>`

3683 `<!beamer>`

3684 `{ \ctex_file_input:n { ctex-name-gbk.cfg } }`

3685 `{ \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。

3686 `\int_compare:nNf \g__ctex_font_size_flag > { -1 }`

3687 `{ \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。

3688 `\fp_compare:nNf { \l__ctex_line_spread_fp } ? { \c_zero_fp }`

3689 `{ \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }`

3690 `</!beamer>`

3691 `</!generic>`

不使用标题定义时的通用设置。

3692 `<*generic>`

3693 `\tl_set:Nn \l__ctex_tmp_tl { beamer }`

3694 `\tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl`

3695 `{`

3696 `\str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }`

3697 `{`

3698 `\uselanguage { ChineseGBK }`

3699 `\languagealias { chinese } { ChineseGBK }`

3700 `\ctex_file_input:n { ctex-name-gbk.cfg }`

3701 `}`

3702 `{`

3703 `\uselanguage { ChineseUTF8 }`

3704 `\languagealias { chinese } { ChineseUTF8 }`

3705 `\ctex_file_input:n { ctex-name-utf8.cfg }`

3706 `}`

```

3707 \clist_put_left:Nn \trans@languagepath { chinese }
3708 }
3709 {
3710 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3711 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3712 { \ctex_file_input:n { ctex-name-utf8.cfg } }
3713 \int_compare:nNnF \g__ctex_font_size_flag > { -1 }
3714 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3715 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3716 { \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。

```

3717 \tl_if_exist:NT \c__ctex_std_class_tl
3718 { \RequirePackage { indentfirst } }
3719 }
3720 </generic>
3721 </scheme&chinese>

```

14.10 中文字号

```

3722 <*class|ctex|ctexsize>

```

```

\zihao 3723 \NewDocumentCommand \zihao { m }
3724 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

```

\ctex_zihao:n 3725 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3726 {
3727 \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3728 { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3729 { \msg_error:nnn { ctex } { fontsize } {#1} }
3730 }
3731 \msg_new:nnnn { ctex } { fontsize }
3732 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3733 {
3734 The~old~font~size~is~used~if~you~continue.\\
3735 The~available~font~sizes~are~listed~as~follow.\\
3736 \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3737 }

```

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。

```

3738 \prop_new:N \c__ctex_font_size_prop
3739 \seq_new:N \c__ctex_font_size_seq
3740 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3741 {
3742 \use:x
3743 {
3744 \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3745 {
3746 { \dim_to_decimal:n {#2} }
3747 { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3748 }
3749 }
3750 \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3751 }
3752 \clist_map_inline:nn
3753 {
3754 { 8 } { 5 bp } ,
3755 { 7 } { 5.5 bp } ,
3756 { -6 } { 6.5 bp } ,

```

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


```

3757 { 6 } { 7.5 bp } ,
3758 { -5 } { 9 bp } ,
3759 { 5 } { 10.5 bp } ,
3760 { -4 } { 12 bp } ,
3761 { 4 } { 14 bp } ,
3762 { -3 } { 15 bp } ,
3763 { 3 } { 16 bp } ,
3764 { -2 } { 18 bp } ,
3765 { 2 } { 22 bp } ,
3766 { -1 } { 24 bp } ,
3767 { 1 } { 26 bp } ,
3768 { -0 } { 36 bp } ,
3769 { 0 } { 42 bp }
3770 }
3771 { \__ctex_save_font_size:nn #1 }

```

```

\ctex_declare_math_sizes:nnnn 3772 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3773 {
3774   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3775   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3776 }

```

```

\__ctex_get_font_sizes:Nn 3777 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3778 {
3779   \tl_clear:N #1
3780   \tl_map_inline:nn {#2}
3781   {
3782     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3783     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3784     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3785   }
3786 }

3787 \clist_map_inline:nn
3788 {
3789   { 8 } { 8 } { 5pt } { 5pt } ,
3790   { 7 } { 7 } { 5pt } { 5pt } ,
3791   { -6 } { -6 } { 5pt } { 5pt } ,
3792   { 6 } { 6 } { 5pt } { 5pt } ,
3793   { -5 } { -5 } { 6pt } { 5pt } ,
3794   { 5 } { 5 } { 7pt } { 5pt } ,
3795   { -4 } { -4 } { 8pt } { 6pt } ,
3796   { 4 } { 4 } { 5 } { 6 } ,
3797   { -3 } { -3 } { -4 } { -5 } ,
3798   { 3 } { 3 } { 4 } { 5 } ,
3799   { -2 } { -2 } { -3 } { -4 } ,
3800   { 2 } { 2 } { 3 } { 4 } ,
3801   { -1 } { -1 } { -2 } { -3 } ,
3802   { 1 } { 1 } { 2 } { 3 } ,
3803   { -0 } { -0 } { -1 } { -2 } ,
3804   { 0 } { 0 } { 1 } { 2 }
3805 }
3806 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3807 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3808 {
3809   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3810   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3811   { \msg_error:nnn { ctex } { fontsize } {#2} }
3812 }
3813 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3814 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

3815 \if_case:w \g__ctex_font_size_flag

```

```

3816 \ctex_file_input:n { ctex-c5size.clo }
3817 \or:
3818 \ctex_file_input:n { ctex-cs4size.clo }
3819 \fi:
3820 </class|ctex|ctexsize>
3821 <*c5size>
3822 \ctex_set_font_size:Nnn \normalsize { 5 }
3823 {
3824   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3825   \abovedisplayshortskip \z@ \@plus3\p@
3826   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3827   \belowdisplayskip \abovedisplayskip
3828   \let\@listi\@listI
3829 }
3830 \ctex_set_font_size:Nnn \small { -5 }
3831 {
3832   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3833   \abovedisplayshortskip \z@ \@plus2\p@
3834   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3835   \def\@listi{\leftmargin\leftmarginI
3836             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3837             \parsep 2\p@ \@plus\p@ \@minus\p@
3838             \itemsep \parsep}
3839   \belowdisplayskip \abovedisplayskip
3840 }
3841 \ctex_set_font_size:Nnn \footnotesize { 6 }
3842 {
3843   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3844   \abovedisplayshortskip \z@ \@plus\p@
3845   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3846   \def\@listi{\leftmargin\leftmarginI
3847             \topsep 3\p@ \@plus\p@ \@minus\p@
3848             \parsep 2\p@ \@plus\p@ \@minus\p@
3849             \itemsep \parsep}
3850   \belowdisplayskip \abovedisplayskip
3851 }
3852 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3853 \ctex_set_font_size:Nnn \tiny { 7 } { }
3854 \ctex_set_font_size:Nnn \large { -4 } { }
3855 \ctex_set_font_size:Nnn \Large { -3 } { }
3856 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3857 \ctex_set_font_size:Nnn \huge { 2 } { }
3858 \ctex_set_font_size:Nnn \Huge { 1 } { }
3859 </c5size>
3860 <*cs4size>
3861 \ctex_set_font_size:Nnn \normalsize { -4 }
3862 {
3863   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3864   \abovedisplayshortskip \z@ \@plus3\p@
3865   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3866   \belowdisplayskip \abovedisplayskip
3867   \let\@listi\@listI
3868 }
3869 \ctex_set_font_size:Nnn \small { 5 }
3870 {
3871   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3872   \abovedisplayshortskip \z@ \@plus3\p@
3873   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3874   \def\@listi{\leftmargin\leftmarginI
3875             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3876             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3877             \itemsep \parsep}
3878   \belowdisplayskip \abovedisplayskip
3879 }
3880 \ctex_set_font_size:Nnn \footnotesize { -5 }
3881 {

```

```

3882 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3883 \abovedisplayshortskip \z@ \@plus3\p@
3884 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3885 \def\@listi{\leftmargin\leftmargini
3886 \topsep 6\p@ \@plus2\p@ \@minus2\p@
3887 \parsep 3\p@ \@plus2\p@ \@minus\p@
3888 \itemsep \parsep}
3889 \belowdisplayskip \abovedisplayskip
3890 }
3891 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3892 \ctex_set_font_size:Nnn \tiny { -6 } { }
3893 \ctex_set_font_size:Nnn \large { -3 } { }
3894 \ctex_set_font_size:Nnn \Large { -2 } { }
3895 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3896 \ctex_set_font_size:Nnn \huge { -1 } { }
3897 \ctex_set_font_size:Nnn \Huge { 1 } { }
3898 </cs4size>
<ctexsize> 3899 \normalsize
3900 <*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`。

```

3901 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3902 {
3903 \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。

```

3904 \group_begin: \footnotesize \exp_args:NNNo \group_end:
3905 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3906 }

```

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

```

3907 \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。`CTeX` 可以直接用在 PDF 书签中。

```

3908 \NewDocumentCommand \CTeX { }
3909 { C \TeX }
3910 \ctex_at_end_package:nn { hyperref }
3911 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

`captiondelimitter` 过时选项。

```

3912 \keys_define:nn { ctex }
3913 {
3914 captiondelimitter .code:n =
3915 {
3916 \msg_warning:nnn { ctex } { deprecated-option }
3917 { You~can~load~the~package~`caption'~to~get~its~functionality. }
3918 }
3919 }

```

3920 `</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`

```

3921 <*scheme&chinese&(article|book|report)>
3922 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3923 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3924 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3925 </scheme&chinese&(article|book|report)>
3926 <*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。

```

3927 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3928 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3929 \msg_new:nnn { ctex } { fontset-unavailable }
3930 { CTex~fontset~`#1'~is~unavailable~in~current~mode. }

```

`\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。

```

3931 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3932 {
3933   \tl_if_empty:NTF \g__ctex_fontset_tl
3934   {
3935     \ctex_detected_platform:
3936     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3937   }
3938   {
3939     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3940     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3941     {
3942       \use:x
3943       {
3944         \ctex_detected_platform:
3945         \msg_error:nnxx { ctex } { fontset-not-found }
3946         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3947       }
3948       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3949     }
3950   }
3951 }
3952 \@onlypreamble \ctex_load_fontset:
3953 \msg_new:nnnn { ctex } { fontset-not-found }
3954 {
3955   CTex~fontset~`#1'~could~not~be~found.\\
3956   Fontset~`#2'~will~be~used~instead.
3957 }
3958 { You~may~run~`mktexlsr'~firstly. }

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

3959 \keys_define:nn { ctex }
3960 {
3961   fontset .code:n =
3962   {
3963     \ctex_if_preamble:TF
3964     {
3965       \str_if_eq_x:nnTF {#1} { none }
3966       { \msg_warning:nnn { ctex } { invalid-value } {#1} }

```

```

3967         {
3968             \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3969             {
3970                 \tl_gset:Nx \g__ctex_fontset_tl {#1}
3971                 \ctex_load_fontset:
3972             }
3973             {
3974                 \msg_error:nxxx { ctex } { fontset-loaded }
3975                 { \g__ctex_fontset_tl } {#1}
3976             }
3977         }
3978     }
3979     { \msg_error:nn { ctex } { fontset-only-preamble } }
3980 }
3981 }
3982 \msg_new:nnnn { ctex } { fontset-loaded }
3983 {
3984     CTex~fontset~`#1'~has~been~loaded.
3985     \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
3986 }
3987 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3988 \msg_new:nnn { ctex } { fontset-only-preamble }
3989 {
3990     The~`fontset'~option~can~be~used~only~in~preamble.
3991 }

```

载入中文字库。

```

3992 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3993 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

3994 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3995 </class|ctex>
3996 <*config>
3997 %%
3998 </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。

```

3999 <*ctexopts>
4000 %%
4001 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
4002 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

4003 <*c19|c70>
4004 %%
4005 %% Chinese characters
4006 %%
<c19> 4007 %% character set: GBK (extension of GB 2312)
<c70> 4008 %% character set: Unicode
4009 %% font encoding: Unicode
4010 %%
4011 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm&c19> 4012 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm&c70> 4013 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf&c19> 4014 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 4015 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 4016 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt&c70> 4017 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

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

```

4018 <*uptex|aptex>
4019 \DeclareKanjiFamily{JY2}{rm}{-}{}
4020 \DeclareKanjiFamily{JY2}{sf}{-}{}
4021 \DeclareKanjiFamily{JY2}{tt}{-}{}
4022 \DeclareKanjiFamily{JT2}{rm}{-}{}
4023 \DeclareKanjiFamily{JT2}{sf}{-}{}
4024 \DeclareKanjiFamily{JT2}{tt}{-}{}
4025 </uptex|aptex>

4026 <*rm>
4027 <*c19>
4028 \DeclareFontShape{C19}{rm}{m}{n}{-}{} CJK * gbksong{\CJKnormal}
4029 \DeclareFontShape{C19}{rm}{b}{n}{-}{} CJK * gbkhei{\CJKnormal}
4030 \DeclareFontShape{C19}{rm}{bx}{n}{-}{} CJK * gbkhei{\CJKnormal}
4031 \DeclareFontShape{C19}{rm}{m}{sl}{-}{} CJK * gbksongsl{\CJKnormal}
4032 \DeclareFontShape{C19}{rm}{b}{sl}{-}{} CJK * gbkheisl{\CJKnormal}
4033 \DeclareFontShape{C19}{rm}{bx}{sl}{-}{} CJK * gbkheisl{\CJKnormal}
4034 \DeclareFontShape{C19}{rm}{m}{it}{-}{} CJK * gbkkai{\CJKnormal}
4035 \DeclareFontShape{C19}{rm}{b}{it}{-}{} CJKb * gbkkai{\CJKbold}
4036 \DeclareFontShape{C19}{rm}{bx}{it}{-}{} CJKb * gbkkai{\CJKbold}
4037 </c19>
4038 <*c70>
4039 \DeclareFontShape{C70}{rm}{m}{n}{-}{} CJK * unisong{\CJKnormal}
4040 \DeclareFontShape{C70}{rm}{b}{n}{-}{} CJK * unihei{\CJKnormal}
4041 \DeclareFontShape{C70}{rm}{bx}{n}{-}{} CJK * unihei{\CJKnormal}
4042 \DeclareFontShape{C70}{rm}{m}{sl}{-}{} CJK * unisongsl{\CJKnormal}
4043 \DeclareFontShape{C70}{rm}{b}{sl}{-}{} CJK * uniheisl{\CJKnormal}
4044 \DeclareFontShape{C70}{rm}{bx}{sl}{-}{} CJK * uniheisl{\CJKnormal}
4045 \DeclareFontShape{C70}{rm}{m}{it}{-}{} CJK * unikai{\CJKnormal}
4046 \DeclareFontShape{C70}{rm}{b}{it}{-}{} CJKb * unikai{\CJKbold}
4047 \DeclareFontShape{C70}{rm}{bx}{it}{-}{} CJKb * unikai{\CJKbold}
4048 </c70>
4049 </rm>
4050 <*uptex|aptex>
4051 \DeclareFontShape{JY2}{rm}{m}{n}{-}{} upzhserif-h{}
4052 \DeclareFontShape{JY2}{rm}{m}{it}{-}{} upzhserifit-h{}
4053 \DeclareFontShape{JY2}{rm}{bx}{n}{-}{} upzhserifb-h{}
4054 \DeclareFontShape{JT2}{rm}{m}{n}{-}{} upzhserif-v{}
4055 \DeclareFontShape{JT2}{rm}{m}{it}{-}{} upzhserifit-v{}
4056 \DeclareFontShape{JT2}{rm}{bx}{n}{-}{} upzhserifb-v{}
4057 </uptex|aptex>

4058 <*sf>
4059 <*c19>
4060 \DeclareFontShape{C19}{sf}{m}{n}{-}{} CJK * gbkyou{\CJKnormal}
4061 \DeclareFontShape{C19}{sf}{b}{n}{-}{} CJKb * gbkyou{\CJKbold}
4062 \DeclareFontShape{C19}{sf}{bx}{n}{-}{} CJKb * gbkyou{\CJKbold}
4063 \DeclareFontShape{C19}{sf}{m}{sl}{-}{} CJK * gbkyousl{\CJKnormal}
4064 \DeclareFontShape{C19}{sf}{b}{sl}{-}{} CJKb * gbkyousl{\CJKbold}
4065 \DeclareFontShape{C19}{sf}{bx}{sl}{-}{} CJKb * gbkyousl{\CJKbold}
4066 \DeclareFontShape{C19}{sf}{m}{it}{-}{} CJK * gbkyou{\CJKnormal}
4067 \DeclareFontShape{C19}{sf}{b}{it}{-}{} CJKb * gbkyou{\CJKbold}
4068 \DeclareFontShape{C19}{sf}{bx}{it}{-}{} CJKb * gbkyou{\CJKbold}
4069 </c19>
4070 <*c70>
4071 \DeclareFontShape{C70}{sf}{m}{n}{-}{} CJK * uniyou{\CJKnormal}
4072 \DeclareFontShape{C70}{sf}{b}{n}{-}{} CJKb * uniyou{\CJKbold}

```

```

4073 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4074 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4075 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4076 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4077 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4078 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4079 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4080 </c70>
4081 </sf>
4082 <*uptex|aptex>
4083 \DeclareFontShape{JY2}{sf}{m}{n}{<-> upzhsans-h}{ }
4084 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{ }
4085 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{ }
4086 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{ }
4087 </uptex|aptex>
4088 <*tt>
4089 <*c19>
4090 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4091 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4092 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4093 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4094 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4095 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4096 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4097 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4098 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4099 </c19>
4100 <*c70>
4101 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4102 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4103 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4104 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4105 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4106 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4107 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4108 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4109 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4110 </c70>
4111 </tt>
4112 <*uptex|aptex>
4113 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{ }
4114 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{ }
4115 </uptex|aptex>
4116 <*fontset>

```

14.15.2 ctex-fontset-windows.def 等

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。)

```

4117 <*windows>
4118 \file_if_exist:nTF { C:/bootfont.bin }
4119 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
4120 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
4121 </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。

```

4122 <*windowsnew>
4123 \tl_new:N \l__ctex_msyh_suffix_tl
4124 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
4125 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }

```

```

4126 {
4127     \file_if_exist:nF { msyh.ttc }
4128     { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
4129 }
4130 </windowsnew>
4131 <*windowsnew|windowsold>
4132 \sys_if_engine_pdftex:TF
4133 {
4134     \ctex_zhmap_case:nnn
4135     {
4136         \ctex_punct_set:n { windows }
4137         \setCJKmainfont
4138         [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
4139 <*windowsold>
4140         \setCJKsansfont { simhei.ttf }
4141         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4142 </windowsold>
4143 <*windowsnew>
4144         \setCJKsansfont
4145         [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4146         \setCJKfamilyfont { zhyahei }
4147         [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4148         \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4149         \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
4150 </windowsnew>
4151         \setCJKmonofont { simfang.ttf }
4152         \setCJKfamilyfont { zhkai } { simkai.ttf }
4153         \setCJKfamilyfont { zhfs } { simfang.ttf }
4154         \setCJKfamilyfont { zhsong } { simsun.ttc }
4155         \setCJKfamilyfont { zhhei } { simhei.ttf }
4156         \setCJKfamilyfont { zhli } { simli.ttf }
4157         \setCJKfamilyfont { zhyou } { simyou.ttf }
4158         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4159         \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4160         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4161         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4162     }
4163     {
4164         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
4165         \ctex_punct_set:n { windows }
4166         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4167         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4168         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4169     }
4170     {
4171         \tl_set:Nn \CJKrmdefault { rm }
4172         \tl_set:Nn \CJKsfdefault { sf }
4173         \tl_set:Nn \CJKttdefault { tt }
4174     }
4175 }
4176 {
4177     \sys_if_engine_uptex:TF
4178     {
4179         \ctex_set_upfonts:nnnnnn
4180         {simsun.ttc} {simhei.ttf} {simkai.ttf}
4181 <windowsnew> {msyh\l__ctex_msyh_suffix_tl} {msyhbd\l__ctex_msyh_suffix_tl}
4182 <windowsold> {simhei.ttf} {simhei.ttf}
4183         {simfang.ttf}
4184         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4185         \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4186         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4187         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4188 <windowsnew> \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4189         \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4190         \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4191         \ctex_set_upfamily:nnn { zhyou } { upscht } {}
4192         \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}

```



```

4193     }
4194     {
4195 <*windowsold>
4196         \setCJKmainfont
4197             [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
4198         \setCJKsansfont { SimHei }
4199         \setCJKmonofont { FangSong_GB2312 }
4200         \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
4201         \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
4202 </windowsold>
4203 <*windowsnew>
4204         \setCJKmainfont
4205             [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
4206         \setCJKsansfont
4207             [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4208         \setCJKmonofont { FangSong }
4209         \setCJKfamilyfont { zhkai } { KaiTi }
4210         \setCJKfamilyfont { zhfs } { FangSong }
4211 </windowsnew>
4212         \setCJKfamilyfont { zhsong } { SimSun }
4213         \setCJKfamilyfont { zhhei } { SimHei }
4214         \setCJKfamilyfont { zhli } { LiSu }
4215         \setCJKfamilyfont { zhyou } { YouYuan }
4216 <*windowsnew>
4217         \setCJKfamilyfont { zhyahei }
4218             [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4219 </windowsnew>
4220     }
4221 }
4222 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```

4223 <*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。

```

4224 \sys_if_engine_pdftex:TF
4225 {
4226     \sys_if_output_pdf:TF
4227     { \ctex_fontset_error:n { adobe } }
4228     {
4229         \ctex_zhmap_case:nnn
4230         {
4231             \setCJKmainfont
4232             [
4233                 cmap = UniGB-UTF16-H ,
4234                 BoldFont = AdobeHeitiStd-Regular.otf ,
4235                 ItalicFont = AdobeKaitiStd-Regular.otf
4236             ] { AdobeSongStd-Light.otf }
4237             \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4238             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4239             \setCJKfamilyfont { zhsong }
4240                 [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
4241             \setCJKfamilyfont { zhhei }
4242                 [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4243             \setCJKfamilyfont { zhkai }
4244                 [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
4245             \setCJKfamilyfont { zhfs }
4246                 [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4247             \ctex_punct_set:n { adobe }
4248             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4249             \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4250             \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4251             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4252             \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }

```

```

4253     }
4254     {
4255         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
4256         \ctex_punct_set:n { adobe }
4257         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4258         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4259         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4260     }
4261     { \ctex_fontset_error:n { adobe } }
4262 }
4263 }
4264 {
4265     \sys_if_engine_uptex:TF
4266     {
4267         \ctex_set_upfonts:nnnnnn
4268         {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
4269         {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
4270         {AdobeFangsongStd-Regular.otf}
4271         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4272         \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4273         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4274         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4275     }
4276     {
4277         \setCJKmainfont
4278         [
4279             BoldFont = AdobeHeitiStd-Regular ,
4280             ItalicFont = AdobeKaitiStd-Regular
4281         ] { AdobeSongStd-Light }
4282         \setCJKsansfont { AdobeHeitiStd-Regular}
4283         \setCJKmonofont { AdobeFangsongStd-Regular}
4284         \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4285         \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4286         \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4287         \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4288     }
4289 }
4290 </adobe>
4291 <*/fandol>

```

14.15.4 ctex-fontset-fandol.def

```

4292 \sys_if_engine_pdftex:TF
4293 {
4294     \sys_if_output_pdf:TF
4295     { \ctex_fontset_error:n { fandol } }
4296     {
4297         \ctex_zhmap_case:nnn
4298         {
4299             \setCJKmainfont
4300             [
4301                 cmap = UniGB-UTF16-H ,
4302                 BoldFont = FandolSong-Bold.otf ,
4303                 ItalicFont = FandolKai-Regular.otf
4304             ] { FandolSong-Regular.otf }
4305             \setCJKsansfont
4306             [
4307                 cmap = UniGB-UTF16-H ,
4308                 BoldFont = FandolHei-Bold.otf
4309             ] { FandolHei-Regular.otf }
4310             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4311             \setCJKfamilyfont { zhsong }
4312             [
4313                 cmap = UniGB-UTF16-H ,
4314                 BoldFont = FandolSong-Bold.otf
4315             ] { FandolSong-Regular.otf }
4316             \setCJKfamilyfont { zhhei }

```

```

4317         [
4318             cmap = UniGB-UTF16-H ,
4319             BoldFont = FandolHei-Bold.otf
4320         ] { FandolHei-Regular.otf }
4321     \setCJKfamilyfont { zhfs }
4322     [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4323     \setCJKfamilyfont { zhkai }
4324     [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4325     \ctex_punct_set:n { fandol }
4326     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4327     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4328     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4329     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4330     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4331     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4332 }
4333 {
4334     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4335     \ctex_punct_set:n { fandol }
4336     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4337     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4338     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4339 }
4340 { \ctex_fontset_error:n { fandol } }
4341 }
4342 }
4343 {
4344     \sys_if_engine_uptex:TF
4345     {
4346         \ctex_set_upfonts:nnnnnn
4347         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4348         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4349         {FandolFang-Regular.otf}
4350         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4351         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4352         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4353         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4354     }
4355     {
4356         \setCJKmainfont
4357         [
4358             Extension = .otf ,
4359             BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular
4360         ]
4361         { FandolSong-Regular }
4362         \setCJKsansfont
4363         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4364         \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4365         \setCJKfamilyfont { zhsong }
4366         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4367         \setCJKfamilyfont { zhhei }
4368         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4369         \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4370         \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4371     }
4372 }
4373 </fandol>

```

14.15.5 ctex-fontset-mac.def

按 [Issue 351](#) 的讨论，以 El Capitan 为分界，分别设置 macold (El Capitan 之前) 和 macnew (El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。

```

4374 <*mac>
4375 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
4376 { \ctex_file_input:n { ctex-fontset-macnew.def } }

```

```
4377 { \ctex_file_input:n { ctex-fontset-macold.def } }
4378 </mac>
```

14.15.6 ctex-fontset-macold.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

```
** 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...
```

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

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。

```

4379 <*macold>
4380 \sys_if_engine_pdftex:TF
4381 { \ctex_fontset_error:n { mac } }
4382 {
4383   \sys_if_engine_uptex:TF
4384   { \ctex_fontset_error:n { mac } }
4385   {
4386     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4387     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4388     \setCJKmonofont { STFangsong }
4389     \setCJKfamilyfont { zhsong } { STSong }
4390     \setCJKfamilyfont { zhhei } { STHeiti }
4391     \setCJKfamilyfont { zhfs } { STFangsong }
4392     \setCJKfamilyfont { zhkai } { STKaiti }
4393   }
4394 }
4395 </macold>

```

14.15.7 ctex-fontset-macnew.def

macnew 在默认字体设置方面,引入了多字重的宋体作为罗马字族,以及引入了苹方黑体作为无衬线字族。

```

4396 <*macnew>
4397 \sys_if_engine_pdftex:TF
4398 { \ctex_fontset_error:n { mac } }
4399 {
4400   \sys_if_engine_uptex:TF
4401   { \ctex_fontset_error:n { mac } }
4402   {
4403     \setCJKmainfont
4404     [
4405       UprightFont = Songti~SC~Light,
4406       ItalicFont = Kaiti~SC,
4407       BoldItalicFont = Kaiti~SC~Bold
4408     ] { Songti~SC }
4409     \setCJKsansfont { PingFang~SC }
4410     \setCJKmonofont { STFangsong }
4411     \setCJKfamilyfont { zhsong }
4412     [
4413       UprightFont = Songti~SC~Light,
4414     ] { Songti~SC }
4415     \setCJKfamilyfont { zhhei } { Heiti~SC }
4416     \setCJKfamilyfont { zhpfs } { PingFang~SC }
4417     \setCJKfamilyfont { zhfs } { STFangsong }
4418     \setCJKfamilyfont { zhkai } { Kaiti~SC }
4419   }
4420 }
4421 </macnew>
4422 <*founder>

```

14.15.8 ctex-fontset-founder.def

```

4423 \sys_if_engine_pdftex:TF
4424 {
4425   \ctex_zhmap_case:nnn
4426   {
4427     \setCJKmainfont

```

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

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

```

4428     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4429     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4430     \setCJKmonofont { FZFSK.TTF }
4431     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4432     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4433     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4434     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4435     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4436     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4437     \ctex_punct_set:n { founder }
4438     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4439     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4440     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4441     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4442     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4443     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4444     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4445 }
4446 {
4447     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4448     \ctex_punct_set:n { founder }
4449     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4450     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4451     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4452 }
4453 { \ctex_fontset_error:n { founder } }
4454 }
4455 {
4456     \sys_if_engine_uptex:TF
4457     {
4458         \ctex_set_upfonts:nnnnnn
4459         { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4460         { FZXH1K.TTF } { FZHTK.TTF }
4461         { FZFSK.TTF }
4462         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4463         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4464         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4465         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4466         \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4467         \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4468         \ctex_set_upfamily:nnn { zhyou } { upschg } {}
4469         \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4470     }
4471     {
4472         \setCJKmainfont
4473         [ 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), \LaTeX 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。

```

4474     \setCJKsansfont [ BoldFont = FZHei-B01 ]
4475     { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4476     \setCJKmonofont { FZFangSong-Z02 }
4477     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4478     \setCJKfamilyfont { zhhei } { FZHei-B01 }
4479     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4480     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4481     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4482     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4483 }
4484 }

```

```
4485 </founder>
4486 <*ubuntu>
```

14.15.9 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。

```
4487 \sys_if_engine_pdftex:TF
4488 {
4489   \ctex_zhmap_case:nnn
4490   {
4491     \setCJKmainfont
4492       [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4493     \setCJKsansfont { wqy-zenhei.ttc }
4494     \setCJKmonofont { uming.ttc }
4495     \setCJKfamilyfont { zhsong } { uming.ttc }
4496     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4497     \setCJKfamilyfont { zhkai } { ukai.ttc }
4498     \ctex_punct_set:n { ubuntu }
4499     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4500     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4501     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4502     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4503     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4504   }
4505   {
4506     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4507     \ctex_punct_set:n { ubuntu }
4508     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4509     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4510     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4511   }
4512   { \ctex_fontset_error:n { ubuntu } }
4513 }
4514 {
4515   \sys_if_engine_uptex:TF
4516   {
4517     \ctex_set_upfonts:nnnnnn
4518       { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4519       { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4520       { uming.ttc }
4521     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4522     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4523     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4524   }
4525   {
4526     \setCJKmainfont
4527       [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4528     \setCJKsansfont { WenQuanYi~Zen~Hei }
4529     \setCJKmonofont { AR~PL~UMing~CN }
4530     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4531     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4532     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4533   }
4534 }
4535 </ubuntu>
```

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

14.15.10 中文字体命令

```

4536 <!*windows&!mac>
4537 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4538 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
<lubuntu> 4539 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4540 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4541 <*windowsnew|windowsold|founder>
4542 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4543 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4544 </windowsnew|windowsold|founder>
<windowsnew> 4545 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
<macnew> 4546 \NewDocumentCommand \yahei { } { \CJKfamily { zhpj } }
<macnew> 4547 \NewDocumentCommand \pingfang { } { \CJKfamily { zhpj } }

4548 <!/windows&!mac>
4549 </fontset>
4550 <*zhmap>

```

14.15.11 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4551 \begingroup\catcode61\catcode48\catcode32=10\relax%
4552 \catcode 35=6 % #
4553 \catcode123=1 % {
4554 \catcode125=2 % }
4555 \toks0{\endlinechar=\the\endlinechar\relax}%
4556 \toks2{\endlinechar=13}%
4557 \def\x#1 #2 {%
4558   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4559   \toks2\expandafter{\the\toks2 \catcode#1=#2}%
4560   \x 13 5 % carriage return
4561   \x 32 10 % space
4562   \x 35 6 % #
4563   \x 40 12 % (
4564   \x 41 12 % )
4565   \x 45 12 % -
4566   \x 46 12 % .
4567   \x 47 12 % /
4568   \x 58 12 % :
4569   \x 60 12 % <
4570   \x 61 12 % =
4571   \x 64 11 % @
4572   \x 91 12 % [
4573   \x 93 12 % ]
4574   \x 123 1 % {
4575   \x 125 2 % }
4576 \edef\x#1{\endgroup%
4577   \edef\noexpand#1{%
4578     \the\toks0 %
4579     \let\noexpand\noexpand\noexpand#1%
4580     \noexpand\noexpand\noexpand\noexpand\undefined%
4581     \noexpand\noexpand\noexpand\noexpand\endinput}%
4582     \the\toks2}%
4583 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4584 \begingroup\expandafter\endgroup
4585 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4586   \expandafter\ifx\csname ifctexpdf\endcsname\relax
4587     \expandafter\ifx\csname pdfoutput\endcsname\relax
4588       iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4589   \else ifctexpdf\fi
4590 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4591 \begingroup

```



```

4592 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4593   \long\def\x#1\ProvidesFile#2[#3]{%
4594     #1%
4595     \immediate\write-1{File: #2 #3}%
4596     \expandafter\xdef\csname ver@#2\endcsname{#3}}
4597 \expandafter\x%
4598 \fi
4599 \endgroup

```

14.15.11.1 zhwindowsfonts.tex

```

4600 < *windows >
4601 \ProvidesFile{zhwindowsfonts.tex}%
4602 [2018/05/01 v2.4.14 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4603
4604 \ifzhmappdf
4605   \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4606   \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4607   \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4608   \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4609   \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4610   \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4611   \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4612
4613   \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4614   \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4615   \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4616   \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4617   \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4618   \pdfmapline{=unili@Unicode@ <simli.ttf}
4619   \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4620
4621   \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4622   \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4623   \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4624   \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4625   \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4626   \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4627
4628   \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4629   \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4630   \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4631   \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4632   \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4633   \pdfmapline{=uniyouisl@Unicode@ <simyou.ttf}
4634
4635 \else
4636   \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4637   \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4638   \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4639   \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4640   \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4641   \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4642   \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4643
4644   \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4645   \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4646   \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4647   \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4648   \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4649   \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4650   \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4651
4652   \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4653   \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4654   \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4655   \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}

```

```

4656 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4657 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4658
4659 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4660 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4661 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4662 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4663 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4664 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4665
4666 </windows>

```

14.15.11.2 zhadobefonts.tex

```

4667 <*adobe>
4668 \ProvidesFile{zhadobefonts.tex}%
4669 [2018/05/01 v2.4.14 Adobe font map loader for DVIPDFMx (CTEX)]
4670
4671 \ifzhmappdf
4672 %% pdfTeX does not support OTF fonts
4673
4674 \else
4675 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4676 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4677 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4678 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4679 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4680
4681 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4682 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4683 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4684 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4685 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4686
4687 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4688 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4689 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4690 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4691
4692 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4693 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4694 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4695 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4696
4697 </adobe>

```

14.15.11.3 zhfandolfonts.tex

```

4698 <*fandol>
4699 \ProvidesFile{zhfandolfonts.tex}%
4700 [2018/05/01 v2.4.14 Fandol font map loader for DVIPDFMx (CTEX)]
4701
4702 \ifzhmappdf
4703 %% pdfTeX does not support OTF fonts
4704
4705 \else
4706 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4707 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4708 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4709 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4710 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4711
4712 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4713 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4714 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4715 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4716 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4717

```

```

4718 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4719 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4720 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4721 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4722
4723 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4724 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4725 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4726 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4727
4728 </fandol>

```

14.15.11.4 zhfounderfonts.tex

```

4729 <*founder>
4730 \ProvidesFile{zhfounderfonts.tex}%
4731 [2018/05/01 v2.4.14 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4732
4733 \ifzhmappdf
4734 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4735 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4736 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4737 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4738 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4739 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4740 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4741
4742 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4743 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4744 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4745 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4746 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4747 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4748 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4749
4750 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4751 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4752 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4753 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4754 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4755 \pdfmapline{=gbkyoussl@UGBK@ <FZY1K.TTF}
4756
4757 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4758 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4759 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4760 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4761 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4762 \pdfmapline{=uniyoussl@Unicode@ <FZY1K.TTF}
4763
4764 \else
4765 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4766 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4767 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4768 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4769 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4770 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4771 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4772
4773 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4774 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4775 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4776 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4777 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4778 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4779 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4780
4781 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4782 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}

```

```

4783 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4784 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4785 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4786 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4787
4788 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4789 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4790 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4791 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4792 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4793 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4794
4795 </founder>

```

14.15.11.5 zhubuntufonts.tex

```

4796 <*ubuntu>
4797 \ProvidesFile{zhubuntufonts.tex}%
4798 [2018/05/01 v2.4.14 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4799
4800 \ifzhmappdf
4801 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4802 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4803 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4804 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4805 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4806 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4807
4808 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4809 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4810 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4811 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4812 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4813 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
4814
4815 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4816 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4817 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4818 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4819 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4820
4821 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4822 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4823 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4824 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4825 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4826
4827 \else
4828 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4829 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4830 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4831 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4832 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4833
4834 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4835 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4836 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4837 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4838 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4839
4840 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4841 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4842 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4843 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4844
4845 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4846 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4847 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}

```

```

4848 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4849
4850 </ubuntu>
4851 \fi
4852
4853 \ctex@zhmap@endinput
4854 </zhmap>

```

14.15.12 制作 spa 文件

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

```

4855 <*spa>
4856 <*macro>
4857 \input expl3-generic %
4858 \ExplSyntaxOn
4859 \sys_if_engine_xetex:F
4860 {
4861   \msg_new:nn { ctex } { xetex }
4862   { XeTeX~is~required~to~compile~this~document! }
4863   \msg_fatal:nn { ctex } { xetex }
4864 }

```

CJKpunct 定义的标点符号是:

‘ “ 「 『 [([{ < « [【
—...、。、. . : ; ! ? %)] } > »] 】 ’ ” 」 』

注意顺序不能改变。

```

4865 \seq_const_from_clist:Nn \c__ctex_punct_seq
4866 {
4867   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4868   "3008 , "300A , "3016 , "3010 ,
4869   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4870   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4871   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4872 }

```

\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。

```

4873 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4874 {
4875   \iow_open:Nn \g__ctex_spa_iow {#1}
4876   \clist_map_inline:nn {#2}
4877   { \__ctex_write_family:nn ##1 }
4878   \iow_close:N \g__ctex_spa_iow
4879 }
4880 \iow_new:N \g__ctex_spa_iow
4881 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn
4882 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4883 {
4884   \group_begin:
4885   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4886   \l__ctex_punct_font
4887   \clist_clear:N \l__ctex_punct_bounds_clist
4888   \seq_map_inline:Nn \c__ctex_punct_seq
4889   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4890   \iow_now:Nx \g__ctex_spa_iow
4891   {
4892     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4893   { \l__ctex_punct_bounds_clist , , , }
4894 }

```

```

4895   \group_end:
4896 }
4897 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4898 {
4899   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4900   {
4901     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4902     \__ctex_calc_bounds:nn { \c_three } {#1}
4903   }
4904 }
4905 \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。

```

4906 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4907 {
4908   \fp_eval:n
4909   {
4910     round
4911     (
4912       \dim_to_decimal_in_unit:nn
4913       { 100 \xetex_glyphbounds:D #1 ~ #2 }
4914       { 1 em }
4915     )
4916   }
4917 }
4918 \ExplSyntaxOff
4919 </macro>

```

下面是 CTeX 定义的一些字体。

```

4920 <*make>
4921 \input ctexspamacro %
4922
4923 \MAKESPA {ctexpunct.spa}
4924 {
4925   {adobezhsong}      {AdobeSongStd-Light} ,
4926   {adobezhhei}      {AdobeHeitiStd-Regular} ,
4927   {adobezhkai}      {AdobeKaitiStd-Regular} ,
4928   {adobezhfs}       {AdobeFangsongStd-Regular} ,
4929   {fandolzhsong}    {FandolSong} ,
4930   {fandolzhsongb}   {FandolSong-Bold} ,
4931   {fandolzhhei}     {FandolHei} ,
4932   {fandolzhheib}    {FandolHei-Bold} ,
4933   {fandolzhkai}     {FandolKai} ,
4934   {fandolzhfs}      {FandolFang} ,
4935   {founderzhsong}   {FZShuSong-Z01} ,
4936   {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
4937   {founderzhhei}    {FZHei-B01} ,
4938   {founderzhheil}   {FZXiHeiI-Z08} ,
4939   {founderzhkai}    {FZKai-Z03} ,
4940   {founderzhfs}     {FZFangSong-Z02} ,
4941   {founderzhli}     {FZLiShu-S01} ,
4942   {founderzhyou}    {FZXiYuan-M01} ,
4943   {founderzhyoub}   {FZZhunYuan-M02} ,
4944   {ubuntuzhsong}    {AR PL UMinG CN} ,
4945   {ubuntuzhhei}     {WenQuanYi Zen Hei} ,
4946   {ubuntuzhkai}     {AR PL UKai CN} ,
4947   {windowszhsong}   {SimSun} ,
4948   {windowszhhei}    {SimHei} ,
4949   {windowszhkai}    {KaiTi} ,
4950   {windowszhfs}     {FangSong} ,
4951   {windowszhli}     {LiSu} ,
4952   {windowszhyou}    {YouYuan} ,
4953   {windowszhyawei}   {Microsoft YaHei} ,
4954   {windowszhyaeib}  {Microsoft YaHei Bold}
4955 }
4956

```



```

4957 \primitive\end
4958 </make>
4959 </spa>

```

14.16 translator 宏包的中文字典

```

4960 <*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。

```

4961 <*theorem>
4962 \providetranslation{Comments}{评论}
4963 \providetranslation{comments}{评论}
4964 \providetranslation{Comment}{评论}
4965 \providetranslation{comment}{评论}
4966 \providetranslation{Corollaries}{推论}
4967 \providetranslation{corollaries}{推论}
4968 \providetranslation{Corollary}{推论}
4969 \providetranslation{corollary}{推论}
4970 \providetranslation{Definitions}{定义}
4971 \providetranslation{definitions}{定义}
4972 \providetranslation{Definition}{定义}
4973 \providetranslation{definition}{定义}
4974 \providetranslation{Examples}{例}
4975 \providetranslation{examples}{例}
4976 \providetranslation{Example}{例}
4977 \providetranslation{example}{例}
4978 \providetranslation{Exercises}{练习}
4979 \providetranslation{exercises}{练习}
4980 \providetranslation{Exercise}{练习}
4981 \providetranslation{exercise}{练习}
4982 \providetranslation{Facts}{事实}
4983 \providetranslation{facts}{事实}
4984 \providetranslation{Fact}{事实}
4985 \providetranslation{fact}{事实}
4986 \providetranslation{Key Lemmas}{关键引理}
4987 \providetranslation{key lemmas}{关键引理}
4988 \providetranslation{Key Lemma}{关键引理}
4989 \providetranslation{key lemma}{关键引理}
4990 \providetranslation{Key Observations}{关键观察}
4991 \providetranslation{key observations}{关键观察}
4992 \providetranslation{Key Observation}{关键观察}
4993 \providetranslation{key observation}{关键观察}
4994 \providetranslation{Lemmas}{引理}
4995 \providetranslation{lemmas}{引理}
4996 \providetranslation{Lemma}{引理}
4997 \providetranslation{lemma}{引理}
4998 \providetranslation{Main Theorems}{主要定理}
4999 \providetranslation{main theorems}{主要定理}
5000 \providetranslation{Main Theorem}{主要定理}
5001 \providetranslation{main theorem}{主要定理}
5002 \providetranslation{Observations}{观察}
5003 \providetranslation{observations}{观察}
5004 \providetranslation{Observation}{观察}
5005 \providetranslation{observation}{观察}
5006 \providetranslation{Problems}{问题}
5007 \providetranslation{problems}{问题}
5008 \providetranslation{Problem}{问题}
5009 \providetranslation{problem}{问题}
5010 \providetranslation{Proofs}{证明}
5011 \providetranslation{proofs}{证明}
5012 \providetranslation{Proof}{证明}
5013 \providetranslation{proof}{证明}
5014 \providetranslation{Proof Sketch}{证明提要}
5015 \providetranslation{Proof sketch}{证明提要}

```

```

5016 \providetranslation{proof sketch}{证明提要}
5017 \providetranslation{Proof Sketches}{证明提要}
5018 \providetranslation{Proof sketches}{证明提要}
5019 \providetranslation{proof sketches}{证明提要}
5020 \providetranslation{Sketch of Proof}{证明提要}
5021 \providetranslation{Sketch of Proofs}{证明提要}
5022 \providetranslation{Sketch of proof}{证明提要}
5023 \providetranslation{Sketch of proofs}{证明提要}
5024 \providetranslation{sketch of proof}{证明提要}
5025 \providetranslation{sketch of proofs}{证明提要}
5026 \providetranslation{Propositions}{命题}
5027 \providetranslation{propositions}{命题}
5028 \providetranslation{Proposition}{命题}
5029 \providetranslation{proposition}{命题}
5030 \providetranslation{Remarks}{注}
5031 \providetranslation{remarks}{注}
5032 \providetranslation{Remark}{注}
5033 \providetranslation{remark}{注}
5034 \providetranslation{Solutions}{解}
5035 \providetranslation{solutions}{解}
5036 \providetranslation{Solution}{解}
5037 \providetranslation{solution}{解}
5038 \providetranslation{Theorems}{定理}
5039 \providetranslation{theorems}{定理}
5040 \providetranslation{Theorem}{定理}
5041 \providetranslation{theorem}{定理}
5042 </theorem>
5043 </dict>

```

14.17 ctexcap 宏包

```
5044 <*ctexcap>
```

ctexcap 是过时宏包。

```

5045 \clist_new:N \l__ctex_ctexcap_options_clist
5046 \clist_set:Nx \l__ctex_ctexcap_options_clist
5047 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
5048 \msg_new:nnn { ctexcap } { deprecated }
5049 {
5050   Package~`ctexcap'~is~deprecated.\\
5051   Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
5052   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5053 }
5054 \msg_warning:nnx { ctexcap } { deprecated }
5055 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 **ctex**。

```

5056 \PassOptionsToPackage { heading = true } { ctexcap }
5057 \RequirePackageWithOptions { ctex }
5058 </ctexcap>

```

14.18 ctexhook 宏包

```
5059 <*ctexhook>
```

实现 **etoolbox** 宏包的 **\AtEndPreamble** 和 **\AfterEndPreamble**。

```

5060 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
5061 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
5062 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
5063 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
5064 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
5065 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5066 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
5067 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5068 \cs_set_nopar:Npx \document

```

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n

```



```

5069 {
5070   \CTEX@document@left@hook
5071   \exp_not:o { \document }
5072   \CTEX@document@right@hook
5073 }
5074 \tl_new:N \g__ctex_end_preamble_hook_tl
5075 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

`\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。

```

5076 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
5077 {
5078   \@ifpackageloaded {#1}
5079     {#2}
5080     { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
5081 }

```

`\ctex_gadd_hook:Nn` 给钩子附加内容。

```

\ctex_gadd_hook:cn
5082 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
5083 {
5084   \tl_if_exist:NF #1 { \tl_new:N #1 }
5085   \tl_gput_right:Nn #1 {#2}
5086 }
5087 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

`\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
5088 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
5089 {
5090   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
5091     { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
5092 }
5093 \cs_generate_variant:Nn \ctex_package_end_hook:n { o }

```

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

```

5094 \tl_put_left:Nn \@popfilename
5095 {
5096   \cs_if_eq:NNT \@currentx \@pkgextension
5097     { \ctex_package_end_hook:o { \@currname } }
5098 }
5099 </ctexhook>

```

14.19 ctexpatch 宏包

```

5100 <*ctexpatch>
5101 \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。

```

5102 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
5103 {
5104   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5105     { once } {#2} { \use_i:nn } { \use_ii:nn }
5106 }

```

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

```

5107 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5108 {
5109   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5110     { all } {#2} { \use_i:nn } { \use_ii:nn }
5111 }

```

`\ctex_patch_cmd:Nnn` 快捷方式,在补丁的时候关闭 \LaTeX 3 语法和设置 @ 为字母类,补丁失败时给出警告。

```

5112 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5113 {
5114   \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnw #1
5115   { once }
5116   {
5117     \ExplSyntaxOff
5118     \char_set_catcode_letter:n { 64 }
5119   }
5120   { }
5121   { \ctex_patch_failure:N #1 }
5122 }
5123 \cs_new_protected:Npn \ctex_patch_failure:N #1
5124 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5125 \msg_new:nnn { ctex } { patch-failure }
5126 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

`\ctex_preto_cmd:NnnTF` 在宏的原本定义前面增加钩子。

```

5127 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5128 {
5129   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
5130   { left } {#2} { \use_i:nn } { \use_ii:nn }
5131 }

```

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

```

5132 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5133 {
5134   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
5135   { right } {#2} { \use_i:nn } { \use_ii:nn }
5136 }

```

`\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。

```

5137 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
5138 {
5139   \tl_set:Nn \__ctex_patch_true:w {#5}
5140   \tl_set:Nn \__ctex_patch_false:w {#6}
5141   \group_begin:
5142     \char_set_catcode_other:n { 35 }
5143     \ctex_parse_name:NN #1 #2 {#3} {#4}
5144   }
5145   \tl_new:N \__ctex_patch_true:w
5146   \tl_new:N \__ctex_patch_false:w

```

`\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_\_\_\_ % \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。

```

5147 \cs_new_protected:Npn \ctex_parse_name:NN #1#2

```

```

5148 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5149 \group_begin:
5150 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5151 {
5152   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5153   {
5154     \bool_lazy_or:nnTF
5155     { \cs_if_exist_p:c { ##3 ~ } }
5156     { \cs_if_exist_p:c { #1##3 } }
5157     {
5158       \group_begin:
5159       \use:x
5160       {
5161         \group_end:
5162         \__ctex_parse_name:nNNNnN
5163         { \token_get_replacement_spec:N ##2 }
5164         \exp_not:N ##2
5165         \exp_not:c { ##3 ~ }
5166         \exp_not:c { #1##3 }
5167       } {##3} ##1
5168     }
5169     { ##1##2 }
5170   }
5171   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5172   {
5173     \exp_args:Nc ##6
5174     {
5175       \str_case:nnTF {##1}
5176       {
5177         { \protect ##3 } { }
5178         { \x@protect ##2 \protect ##3 } { }
5179       }
5180       {
5181         \str_if_eq_x:nnTF
5182         { \exp_not:n { #1@protected@ ##3 #1##3 } }
5183         {
5184           \exp_last_unbraced:Nf \__ctex_parse_name:w
5185           \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
5186         }
5187         { #1##5 ~ } { ##5 ~ }
5188       }
5189       {
5190         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5191         {
5192           { #1@protected@ ##2 ##4 } { }
5193           { #1@ ##4 } { }
5194         }
5195         { #1##5 } {##5}
5196       }
5197     }
5198   }
5199   \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5200 }
5201 \use:x
5202 {
5203   \__ctex_tmp:w
5204   { \c_backslash_str }
5205   { \c_left_brace_str }
5206   { \tl_to_str:n { testopt } }
5207 }
5208 \group_end:
5209 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

```

分别保存宏的 `\meaning` 中的前缀、参数文本和替换文本。

```

5210 \str_new:N \l__ctex_prefix_str
5211 \str_new:N \l__ctex_parameter_str

```

```
5212 \str_new:N \l__ctex_replacement_str
```

解构待补丁宏的 `\meaning`。若命令不是宏,则走向 `false` 分支。

```
\ctex_get_macro_meaning:NTF
\__ctex_get_macro_meaning:w
5213 \group_begin:
5214 \cs_set_protected:Npn \__ctex_tmp:w #1
5215 {
5216   \prg_new_protected_conditional:Npnn
5217   \ctex_get_macro_meaning:N ##1 { TF }
5218   {
5219     \exp_after:wN \__ctex_get_macro_meaning:w
5220     \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5221   }
5222   \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5223   ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5224   {
5225     \tl_if_empty:nTF { ##4 }
5226     { \prg_return_false: }
5227     {
5228       \str_set:Nn \l__ctex_prefix_str { ##1 }
5229       \str_set:Nn \l__ctex_parameter_str { ##2 }
5230       \str_set:Nn \l__ctex_replacement_str { ##3 }
5231       \prg_return_true:
5232     }
5233   }
5234 }
5235 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5236 \group_end:
```

`\ctex_if_rescanable:NnTF` 检查宏是否可以重建。

```
5237 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5238 {
5239   \ctex_get_macro_meaning:NTF #1
5240   {
5241     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5242     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5243   }
5244   {#4}
5245 }
5246 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:
```

`__ctex_patch_rebuild:Nn` 使用 `\tl_rescan:nn` 来重新记号化 `\meaning` 字符串。

```
5247 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5248 {
5249   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
5250   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
5251   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5252   \use:x
5253   {
5254     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5255     \exp_not:o { \l__ctex_parameter_tl }
5256     { \exp_not:o { \l__ctex_replacement_tl } }
5257   }
5258 }
5259 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5260 {
5261   \str_if_empty:NTF #2
5262   { \tl_clear:N #1 }
5263   { \tl_set_rescan:Nno #1 {#3} {#2} }
5264 }
5265 \tl_new:N \l__ctex_prefix_tl
5266 \tl_new:N \l__ctex_parameter_tl
5267 \tl_new:N \l__ctex_replacement_tl
```

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

```

5268 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnnw #1#2#3#4#5
5269 {
5270   \group_end:
5271   \ctex_if_rescanable:NnTF #1 {#3}
5272   {
5273     \use:x
5274     {
5275       \__ctex_patch_replace:nnnTF {#2}
5276       { \tl_to_str:n {#4} }
5277       { \tl_to_str:n {#5} }
5278     }
5279     {
5280       \__ctex_patch_rebuild:Nn #1 {#3}
5281       \__ctex_patch_true:w
5282     }
5283     { \__ctex_patch_false:w }
5284   }
5285   { \__ctex_patch_false:w }
5286 }

```

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

```

5287 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5288 {
5289   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5290   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5291 }

```

`__ctex_hookto_cmd:Nnnnw` 在宏的前/后附加钩子。

```

5292 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5293 {
5294   \group_end:
5295   \ctex_get_macro_meaning:NnTF #1
5296   {
5297     \str_if_empty:NnTF \l__ctex_parameter_str
5298     { \__ctex_hookto_cmd_parameterless:Nnnnw }
5299     { \__ctex_hookto_cmd_parameter:Nnnnw }
5300     #1 {#2} {#3} {#4}
5301   }
5302   { \__ctex_patch_false:w }
5303 }

```

`__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。

```

5304 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5305 {
5306   \str_if_empty:NnTF \l__ctex_prefix_str
5307   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5308   \tex_edef:D #1
5309   {
5310     \use:c { __ctex_ #2 _hook_aux:nn }
5311     { \exp_not:o {#1} }
5312     { \exp_not:n {#4} }
5313   }
5314   \__ctex_patch_true:w
5315 }
5316 \cs_generate_variant:Nn \tl_rescan:nn { no }
5317 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5318 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

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

```

5319 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5320 {

```

```
5321   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}  
5322   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w  
5323   {  
5324     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}  
5325     \__ctex_patch_rebuild:Nn #1 {#3}  
5326     \__ctex_patch_true:w  
5327   }  
5328   { \__ctex_patch_false:w }  
5329 }  
  
5330 </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。 38		新的标题格式选项 <code>aftertitle</code> 。 84
<code>captiondelimiter</code> 是过时选项。 111		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。 102
<code>fancyhdr</code> 成为过时选项,原选项功能总是打开。 41		<code>\ctex_if_macosx:TF</code> : 改用 <code>/Library/Fonts/Songti.ttc</code> 为特征文件。 72
<code>fntef</code> 成为过时选项,原选项功能总是打开。 41		<code>hyperref</code> : 补充定义 <code>\hypersetup</code> 。 41
<code>hyperref</code> 成为过时选项,原选项功能总是打开。 41		v2.2 (2015/06/21 – 2015/06/30)
<code>indent</code> , <code>noindent</code> 是过时选项。 39		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。 39		<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。 40		不再依赖 <code>etoolbox</code> 宏包。 1
<code>nospace</code> 是过时选项。 41		非 <code>ctexart</code> 类的 <code>part/before skip</code> 和 <code>part/after skip</code> 选项有意义。 87
<code>nozhmap</code> 是过时选项。 40		给 <code>enumitem</code> 宏包注册 <code>\chinese</code> 和 <code>\zhnum</code> 。 82
<code>punct</code> 选项可以设置标点格式。 40		将文档开头和宏包末尾钩子提取到 <code>ctexhook</code> 宏包中。 132
<code>ctex</code> 宏包新增 <code>heading</code> 选项。 41		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。 102
<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。 112
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。 14, 78		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。 111		新增子宏包 <code>ctexpatch</code> 实现给宏打补丁功能。 133
兼容 <code>extsizes</code> 宏包、 <code>beamer</code> 、 <code>memoir</code> 等提供的更多字号选项。 42		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。 38
将中文字号功能提取到可以独立使用的 <code>ctexsize</code> 。 108		代码实现避免使用 <code>\lowercase</code> 技巧 (Joseph Wright)。 37
解决 <code>etoolbox</code> 与 <code>breqn</code> 关于 <code>\end</code> 的冲突。 112		更新 <code>LuaTeX-j</code> 支持 (20150922.0)。 49
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。 50
删除 <code>c19gsn.fd</code> 和 <code>c19gkai.fd</code> 。 1		兼容 <code>titleps</code> 宏包。 97
通过 <code>LuaTeX-j</code> 宏包支持 <code>Lua[®]TeX</code> 。 49		修复 <code>nameformat</code> 作用域问题。 84
新增 <code>autoindent</code> 选项。 39		与 <code>LaTeX3</code> (2015/12/20) 同步。 35
新增 <code>fontset</code> 选项。 39		v2.4 (2015/02/19 – 2016/04/25)
新增 <code>linespread</code> 选项。 39		General: 初步支持 <code>up[®]TeX</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。 41		提供 <code>translator</code> 宏包的中文定理名称翻译。 131
新增 <code>zhmCJK</code> 支持选项。 40		提供 <code>up[®]TeX</code> 的 NFSS 字体定义。 114
新增 <code>zihao</code> 选项。 38		正确更新 <code>CJK</code> 包的 <code>\CJKfamilydefault</code> 。 70
新增统一设置接口 <code>\ctexset</code> 。 78		正确设置 <code>up[®]TeX</code> 下字体命令。 69
应用 <code>LaTeX3</code> 重新整理代码。 1		<code>\ctex_parse_name:NN</code> : 修复宏名解析错误。 134
中文字号不再采用近似值。 108		
自动检测操作系统,载入对应的字体配置。 71		
<code>\CJK@surr</code> : 解决与 <code>\nouppercase</code> 的冲突。 46		
v2.0.1 (2015/05/15)		
General: 修复 10pt、11pt 等选项无效的问题。 42		
v2.0.2 (2015/05/16)		
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。 108		
v2.1 (2015/05/18 – 2015/06/19)		
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>nameformat</code> 可以接受章节名字为参数。 84		
不依赖 <code>ifpdf</code> 宏包。 37		
不再设置 <code>hyperref</code> 宏包的 <code>colorlinks</code> 选项。 72		
给 <code>pdf[®]TeX</code> 下的非 UTF8 编码 CJK 字体族加上 <code>CMap</code> 。 43		

v2.4.2	(2016/05/15)	<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	99
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	50		
<code>\em</code> : 兼容 <code>upL^AT_EX 2016/05/07u00</code> 的定义。	69		
v2.4.3	(2016/06/03 – 2016/08/26)	<code>\CTEX@fontfamily</code> : 解决与 <code>fontspec 2017/01/24 v2.5d</code> 的字体族匹配兼容问题。	61
General: 更新 <code>unicode-math</code> 补丁。	55		
简化 <code>fontspec</code> 补丁。	50		
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		
v2.4.4	(2016/09/09 – 2016/09/19)	General: 常数 <code>\c_minus_one</code> 已过时。	35
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	49	定义 <code>\cht</code> , <code>\cdp</code> 和 <code>\c wd</code> 。	77
改进 <code>hyperref</code> 宏包的标题锚点设置。	94	使用 <code>lazy</code> 函数对 <code>Boolean</code> 表达式进行最小化运算 (<code>L^AT_EX 3 2017/07/19</code>)。	35
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	45		
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	96		
提供 <code>\CTEXifname</code> 。	86		
提供 <code>\partmark</code> 。	86		
新的标题格式选项 <code>break</code> 。	84		
v2.4.5	(2016/10/01 – 2016/10/25)	v2.4.11	(2017/08/17 – 2017/11/21)
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	85	General: 不把 <code>Enclosed Alphanumerics</code> 设置为 <code>J_Achar</code> 。	49
新的标题格式选项 <code>hang</code> 。	84	不把希腊和西里尔字母设置为 <code>J_Achar</code> 。	49
新的标题格式选项 <code>tocline</code> 。	84	因上游 <code>l3keys</code> 变化, 重新定义 <code>format_␣+</code> 等带空格加号的选项。	84
<code>\ps@headings</code> : 修复补丁失败。	99	<code>\ps@headings</code> : 补充页眉空格。	99
v2.4.6	(2016/10/31 – 2016/11/20)	v2.4.12	(2017/12/05 – 2018/01/27)
General: <code>\CTEXifname</code> 初始为假。	86	General: 同步 <code>L^AT_EX 3 2017/12/16</code> 。	1
支持字体属性可选项在后的新语法。	60	修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。	78
重新初始化 <code>\ifthechapter</code> 等。	98	正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	68
v2.4.7	(2016/12/23 – 2016/12/27)	<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	111
General: 依赖 <code>pxeverysel</code> 宏包。	70	v2.4.13	(2018/03/23)
		General: 修正导言区 <code>\selectfont</code> 钩子位置。	76
		v2.4.14	(2018/05/01)
		General: 配置 <code>macnew</code> 的默认字体设置。	121
		区分 <code>macold</code> 及 <code>macnew</code> 。	119
		为 <code>macnew</code> 配置字体命令。	124

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	
_	508, 3240, 3247, 3263, 3266
\#	474, 3141, 3200
.../afterindent	20
.../aftername	18
.../aftername+	18
.../afterskip	21
.../aftertitle	19
.../aftertitle+	19
.../beforeskip	20
.../break	23
.../break+	23
.../fixskip	21
.../format	17
.../format+	17
.../indent	22
.../name	15
.../nameformat	17
.../nameformat+	17
.../number	16
.../numberformat	17
.../numberformat+	17
.../titleformat	19
.../titleformat+	19
.../tocline	23
\ /	1644
\\	15, 33, 34, 102, 145, 147, 149, 206, 216, 672, 1102, 1184, 2326, 3620, 3621, 3734, 3735, 3955, 3985, 5050, 5051, 5052, 5126
\{	1113, 1114, 5052
\}	1113, 1114, 5052
10pt	9, <u>367</u>
11pt	9, <u>367</u>
12pt	9, <u>367</u>
A	
\abovedisplayshortskip .	3825, 3833, 3844, 3864, 3872, 3883
\abovedisplayshortskip	3824, 3827, 3832, 3839, 3843, 3850, 3863, 3866, 3871, 3878, 3882, 3889
\abstractname	2310
abstractname	12
\addCJKfontfeature	1184, 1228
\addCJKfontfeatures	1201
\addcontentsline	2527, 2573, 2577, 2582, 2610, 2614, 2619, 2708, 2713, 2719, 2725, 2848, 2862
\AddEnumerateCounter	2265, 2267, 2268, 2269
\addpenalty	2807, 3460, 3493, 3527, 3541, 3555
\addtocontents	2515, 2521, 2729, 2730
\addvspace	2516, 2522, 2534, 2537, 2729, 2730, 2807, 2811, 3279, 3289, 3298, 3308, 3317, 3327
adobefonts	30
\algorithmname	2317, 2333
algorithmname	12
AlternateFont	27, <u>1340</u>
\appendix	2989
appendix/name	24, <u>2978</u>
appendix/number	24, <u>2978</u>
appendix/numbering	23, <u>2978</u>
\appendixname	2312, 3605
appendixname	12
\arabic	3507
\AtBeginDocument	575, 1633
\AtBeginDvi	433
\AtBeginShipoutFirst	435, 1841, 1856
\AtBeginUTFCommand	719
\AtEndOfClass	135
\AtEndOfPackage	136
\AtEndUTFCommand	725
autoindent	10, <u>185</u> , <u>2120</u>
B	
\baselinestretch	2175
\begin	3285, 3304, 3323
\begingroup	2839, 2875, 2907, 3274, 3293, 3312, 4551, 4584, 4591
\belowdisplayshortskip .	3826, 3834, 3845, 3865, 3873, 3884
\belowdisplayshortskip	3827, 3839, 3850, 3866, 3878, 3889
\bfdefault	966, 972, 1330, 1337
\bfseries	2589, 2592, 2628, 2631, 2644, 2661, 2757, 2762, 2780, 3388, 3390, 3394, 3407, 3408, 3411, 3436, 3438, 3444, 3462, 3466, 3487, 3521, 3538, 3552
\bibname	2315, 2318, 2326, 2334, 2340, 2341
bibname	12
bool commands:	
\bool_gset_false:N	266, 271
\bool_gset_true:N	261
\bool_if:NTF	70, 1142, 2921, 3082, 3092, 3103, 3624
\bool_lazy_or:nnTF	5154
\bool_new:N	131, 1179
\bool_set_false:N	1154
\bool_set_true:N	1172
\c_false_bool	2937, 2939
\c_true_bool	2936, 2938
box commands:	
\box_ht:N	3905
\box_new:N	60
\box_wd:N	1927, 3196
C	
c5size	30
cap	30
captiondelimiter	31, <u>3912</u>
\catcode	4551, 4552, 4553, 4554, 4558, 4559

- CCT 30
- CCTfont 30
- \ccwd 26, 115, 189, 1923, 1970,
1971, 1972, 2126, 2164, 2183, 2184, 2191, 2218, 2219,
2220, 2223, 2225, 2226, 2237, 2248, 2249, 3922, 3923, 3924
- \cdp 2038
- \centering 2625, 2659, 3275,
3294, 3313, 3394, 3406, 3411, 3415, 3444, 3466, 3474, 3509
- \chapter 2685, 3630
- chapter/beforeskip 32
- chapter/fixbeforeskip 32
- chapter/lofskip 22, 2456
- chapter/lotskip 22, 2456
- chapter/numbering 15
- chapter/pagestyle 22, 2456
- \chaptermark 2728
- \chaptername 3433
- char commands:
- \char_set_catcode:nn 73, 512, 513, 548, 549
- \char_set_catcode_active:N 1365
- \char_set_catcode_comment:n 507
- \char_set_catcode_escape:n 508
- \char_set_catcode_letter:n 67, 1636, 3214, 5118
- \char_set_catcode_math_superscript:n 543
- \char_set_catcode_other:N 474, 874, 1348, 3141, 3200
- \char_set_catcode_other:n 542, 5142
- \char_set_catcode_space:n 2044
- \char_value_catcode:n 73, 512, 513, 548, 549
- CharRange 27, 1340
- \Chinese 31, 2261
- \chinese 26, 2259, 2269, 3381, 3443
- \cht 2038
- \CJK 1892
- \CJKbold
4035, 4036, 4046, 4047, 4061, 4062, 4064, 4065, 4067,
4068, 4072, 4073, 4075, 4076, 4078, 4079, 4091, 4092,
4094, 4095, 4097, 4098, 4102, 4103, 4105, 4106, 4108, 4109
- \CJKfamily 522, 526, 530,
534, 1038, 1061, 1276, 1283, 1284, 1285, 1286, 1434,
1672, 4537, 4538, 4539, 4540, 4542, 4543, 4545, 4546, 4547
- \CJKfamilydefault 516,
517, 534, 590, 1263, 1264, 1276, 1286, 1294, 1296, 1300,
1307, 1320, 1655, 1669, 1677, 1731, 1735, 1743, 1756, 1758
- \CJKfilltwosides 1900
- \CJKfontspec 1201
- \CJKglue 1926, 1940, 1954, 1967
- \CJKhook 605
- \CJKnormal 4028, 4029, 4030, 4031, 4032, 4033, 4034,
4039, 4040, 4041, 4042, 4043, 4044, 4045, 4060, 4063,
4066, 4071, 4074, 4077, 4090, 4093, 4096, 4101, 4104, 4107
- \CJKpunctmapfamily
. 638, 639, 640, 641, 642, 643, 647, 648, 649, 650, 651, 652
- \CJKrmdefault 497,
517, 518, 522, 1109, 1120, 1232, 1264, 1265, 1283, 1296,
1299, 1300, 1652, 1655, 1657, 1739, 4158, 4160, 4161,
4166, 4167, 4168, 4171, 4248, 4251, 4252, 4257, 4258,
4259, 4326, 4329, 4330, 4336, 4337, 4338, 4438, 4441,
4442, 4449, 4450, 4451, 4499, 4502, 4503, 4508, 4509, 4510
- \CJKsfdefault 498, 519,
526, 1110, 1121, 1239, 1266, 1284, 1653, 1661, 1740,
4141, 4148, 4149, 4172, 4249, 4327, 4331, 4439, 4443, 4500
- \CJKtilde 591
- \CJKttdefault 499, 520, 530, 1111, 1122, 1245, 1267,
1285, 1654, 1665, 1741, 4159, 4173, 4250, 4328, 4440, 4501
- clearalternatefont 28, 1526
- \cleardoublepage 2548, 2686, 3404, 3431
- \clearpage 2550, 2686, 3404, 3431
- clist commands:
- \clist_clear:N 987, 4887
- \clist_concat:NNN 1007
- \clist_const:Nn 618
- \clist_gput_right:Nn 383, 395, 405, 407
- \clist_gset:Nn 1255
- \clist_if_empty:NTF 1353, 1372, 1612
- \clist_map_break:n 3061, 3615
- \clist_map_function:nN 1529, 1531, 1563, 1577
- \clist_map_inline:Nn 608, 1465
- \clist_map_inline:nn
. 368, 630, 1570, 1884, 1910, 3056, 3612, 3752, 3787, 4876
- \clist_new:N 138, 1004, 1256, 1407, 4905, 5045
- \clist_put_left:Nn 1008, 3681, 3707
- \clist_put_right:Nn 1171, 1453, 4899
- \clist_set:Nn 991, 1390, 1391, 5046
- \clist_use:Nn 5055
- \clubpenalty 2906, 2913
- \contentsname 2305
- contentsname 12
- continuation 12
- cs commands:
- \cs:w 3352
- \cs_end: 3352
- \cs_generate_variant:Nn
..... 93, 450, 462, 951, 1073, 1160, 1180,
1181, 1386, 1423, 1525, 1563, 1571, 5087, 5093, 5209, 5316
- \cs_gset_eq:NN
..... 162, 163, 164, 262, 267, 272, 310, 312, 378,
382, 482, 769, 773, 1147, 1150, 3025, 3026, 3031, 3687, 3714
- \cs_gset_nopar:Npx 770
- \cs_gset_protected_nopar:Npn 2990, 3037
- \cs_gset_protected_nopar:Npx
..... 808, 1034, 1427, 1521, 2966
- \cs_if_eq:NNTF 910, 3023, 5096, 5242, 5322
- \cs_if_exist:NTF 44, 359, 459, 764, 776, 806,
844, 929, 1426, 1518, 2002, 3127, 3339, 3630, 3632, 5101
- \cs_if_exist_p:N 5155, 5156
- \cs_if_exist_use:N 1057, 1058
- \cs_if_exist_use:NTF 4, 747, 1079, 1540, 5090
- \cs_if_free:NTF 444, 454, 610, 625,
1144, 1329, 2139, 2153, 2265, 3075, 3120, 3166, 3171, 3175
- \cs_new:Npn 110, 117, 124,
865, 871, 875, 983, 1060, 2940, 2948, 2975, 3100, 5199, 5317

- `\cs_new_eq:NN` . 6, 7, 8, 9, 20, 23, 95, 106, 107, 108, 133, 134, 360, 451, 595, 656, 785, 810, 886, 917, 1228, 1235, 1752, 1905, 1906, 1907, 1919, 1967, 1978, 1984, 2260, 2261, 2262, 2501, 2504, 2934, 2935, 2936, 2937, 2938, 2939, 2989, 3013, 3014, 3357, 3628, 3633, 4881, 5246, 5318
- `\cs_new_nopar:Npn` 825, 1052, 1105, 1116, 2259, 2438, 2956, 3164, 3169, 4906
- `\cs_new_protected:Npn` 62, 919, 984, 1187, 1351, 1442, 1456, 1607, 1973, 2449, 2526, 3190, 3807, 3813, 4882, 5060, 5062, 5076, 5082, 5102, 5107, 5112, 5123, 5127, 5132, 5137, 5147, 5152, 5171, 5222, 5237, 5247, 5259, 5268, 5287, 5292, 5304, 5319
- `\cs_new_protected_nopar:Npn` 77, 79, 135, 136, 431, 442, 452, 457, 463, 495, 503, 538, 555, 597, 603, 606, 623, 628, 636, 645, 654, 745, 754, 767, 772, 774, 786, 800, 827, 842, 855, 879, 888, 905, 944, 961, 1005, 1013, 1032, 1053, 1063, 1088, 1126, 1138, 1140, 1162, 1164, 1273, 1279, 1290, 1315, 1324, 1370, 1381, 1408, 1417, 1424, 1430, 1450, 1463, 1482, 1496, 1503, 1515, 1535, 1551, 1569, 1572, 1574, 1587, 1599, 1641, 1646, 1648, 1691, 1704, 1718, 1729, 1761, 1779, 1794, 1810, 1821, 1825, 1862, 1869, 1873, 1896, 1902, 1923, 1937, 1969, 1985, 2017, 2062, 2138, 2140, 2165, 2200, 2210, 2215, 2368, 2398, 2485, 2492, 2499, 2505, 2507, 2510, 2919, 2925, 3006, 3016, 3048, 3068, 3073, 3078, 3088, 3098, 3142, 3335, 3344, 3349, 3725, 3740, 3772, 3777, 3927, 3931, 4873, 4897, 5064, 5066, 5088
- `\cs_new_protected_nopar:Npx` ... 467, 1211, 1889, 3129
- `\cs_set:Npn` 1349, 1981, 2425
- `\cs_set_eq:NN` 109, 190, 195, 200, 535, 553, 573, 600, 604, 612, 626, 779, 780, 887, 918, 936, 981, 1038, 1039, 1426, 1434, 1444, 1445, 1446, 1518, 1519, 1520, 1644, 1670, 1734, 2276, 2281, 2284, 2289, 2506, 2508, 2927, 2928, 2929, 2930, 2931, 2932, 3020, 3052, 3062
- `\cs_set_nopar:Npn` 1690, 3181
- `\cs_set_nopar:Npx` 1920, 3090, 5068
- `\cs_set_protected:Npn` 726, 922, 5150, 5214
- `\cs_set_protected_nopar:Npn` 599, 937, 1940, 3122, 3179, 3814
- `\cs_set_protected_nopar:Npx` 1975
- `\cs_to_str:N` 54, 1210, 5148
- `\cs_undefine:N` . 1017, 1018, 1022, 1023, 1447, 1989, 5091
- `cs4size` 30
- `\csname` 2808, 2817, 2818, 2819, 2826, 2844, 2845, 2847, 2850, 2859, 2860, 2861, 2864, 4583, 4585, 4586, 4587, 4592, 4596
- `\CTeX` 26, 3908
- ctex commands:**
 - `\ctex_add_cmap:n` 447, 452
 - `\ctex_add_cmap:Nn` 455, 457, 462
 - `\ctex_add_to_selectfont:n` 1973, 2008, 2036, 2064, 2152
 - `\ctex_after_end_preamble:n` 37, 109, 5060
 - `\ctex_appto_cmd:NnnTF` 478, 1635, 5132
 - `\ctex_assign_heading_name:nn` 2404, 2446, 2982
 - `\ctex_at_end:n` 135, 294, 357, 1967, 3994
 - `\ctex_at_end_package:nn` 133, 434, 481, 734, 927, 1835, 1837, 1855, 2263, 3035, 3108, 3115, 3174, 3176, 3177, 3211, 3251, 3363, 3910, 5076
 - `\ctex_at_end_preamble:n` 362, 580, 587, 1289, 1678, 5060
 - `\ctex_auto_ignorespaces:` 594, 604, 2074
 - `\ctex_CJK_input:n` 538
 - `\ctex_declare_math_sizes:nnnn` 3772, 3806
 - `\ctex_default_pt:n` 113, 117
 - `\ctex_detected_platform:` 1760, 3935, 3944
 - `\ctex_family_cmap:nn` 442, 451
 - `\ctex_file_input:n` 77, 398, 657, 2103, 3639, 3674, 3679, 3684, 3685, 3700, 3705, 3711, 3712, 3816, 3818, 3936, 3940, 3948, 3994, 4119, 4120, 4376, 4377
 - `\ctex_file_wrapper:nnn` 62, 78, 81, 505, 540
 - `\ctex_fix_varioref_label:n` 3347, 3349
 - `\ctex_fontset_error:n` 3927, 4227, 4261, 4295, 4340, 4381, 4384, 4398, 4401, 4453, 4512
 - `\ctex_gadd_hook:Nn` 605, 5080, 5082
 - `\ctex_get_macro_meaning:N` 5217
 - `\ctex_get_macro_meaning:Ntf` 5213, 5239, 5295
 - `\ctex_hypersetup:n` 72, 360, 1818, 1830, 1833, 1848, 1851, 1854
 - `\ctex_if_autoindent_touched:TF` 133, 190, 195, 200, 3666
 - `\ctex_if_ccglue_touched:TF` 1951, 2171
 - `\ctex_if_ccglue_touched_p:` 1951
 - `\ctex_if_macosx:TF` 1801, 1802, 1804, 1810
 - `\ctex_if_preamble:TF` 108, 3963
 - `\ctex_if_rescanable:NnTF` 5237, 5271
 - `\ctex_ignorespaces_case:N` 597, 2071, 2074, 2077
 - `\ctex_load_fontset:` 3931, 3971, 3993
 - `\ctex_load_zhmap:nnnn` 495, 4164, 4255, 4334, 4447, 4506
 - `\ctex_ljt_add_font_features:n` 1162, 1225
 - `\ctex_ljt_add_font_features:nn` 1162, 1163
 - `\ctex_ljt_char_range_key:nn` 1604, 1607
 - `\ctex_ljt_clear_alternate_font:n` 1529, 1535
 - `\ctex_ljt_declare_alternate_shape:nnnnnn` 1436, 1456
 - `\ctex_ljt_declare_char_range:n` 1566, 1569, 1571
 - `\ctex_ljt_declare_char_range:nn` 1573, 1574
 - `\ctex_ljt_def_char_range_key:n` 1579, 1599
 - `\ctex_ljt_ensure_default_family:` 1290, 1748
 - `\ctex_ljt_extract_font:` 779, 786
 - `\ctex_ljt_family_if_exist:nNTF` 1065, 1074, 1294, 1299, 1317, 1320, 1538
 - `\l_ctex_ljt_family_tl` 1067, 1072, 1129, 1130, 1145, 1149, 1152, 1163, 1532, 1533
 - `\ctex_ljt_fontspec:nn` 1126, 1138, 1173, 1220
 - `\ctex_ljt_get_and_define_fonts:nN` 879
 - `\ctex_ljt_get_and_define_fonts_al:nN` 883, 886
 - `\ctex_ljt_get_and_define_fonts_ja:nN` 882, 888
 - `\ctex_ljt_if_alternate_shape_exist:nTF` 789, 802, 820, 829
 - `\ctex_ljt_if_jfont:nTF` 55, 865, 876, 881
 - `\ctex_ljt_if_jfont_math:Ntf` 871, 912
 - `\ctex_ljt_math_group_hook:` 54, 55, 911, 917, 937
 - `\ctex_ljt_pickup_font:` ... 760, 774, 849, 891, 893, 895
 - `\ctex_ljt_reset_alternate_font:n` 1531, 1551
 - `\ctex_ljt_save_alternate_family:Nnnn` 1401, 1421, 1424
 - `\ctex_ljt_save_alternate_seq:Nn` 1376, 1381

- `\ctex_lsj_save_alterdate_seq:Nnnwn` 1381
`\ctex_lsj_save_alterdate_seq:Nnnwnw` 1384, 1387
`\ctex_lsj_save_char_range:n` 1577, 1584
`\ctex_lsj_save_char_range:nn` 1586, 1587
`\ctex_lsj_select_alterdate_font:` 827, 2011
`\ctex_lsj_select_font:` 745, 2010
`\ctex_lsj_set_alterdate_family:nn` 1379, 1408
`\ctex_lsj_set_alterdate_family:nnn` . 1420, 1428, 1430
`\ctex_lsj_set_alterdate_family:Nnnnn` 1412, 1417, 1423
`\ctex_lsj_set_alterdate_family:nnTF` 992, 1370
`\ctex_lsj_set_alterdate_seq:n` 1342, 1347
`\ctex_lsj_set_alterdate_shape:n` ... 1484, 1496, 1507
`\ctex_lsj_set_alterdate_shape:nnN` 1469, 1503
`\ctex_lsj_set_alterdate_shape:nnn` 1472, 1479
`\ctex_lsj_set_alterdate_shape:nnnn` 1481, 1482
`\ctex_lsj_set_alterdate_shape:Nnnnnnn` .. 1459, 1463
`\ctex_lsj_set_family:nnn`
..... 984, 1157, 1204, 1214, 1232, 1239, 1245, 1251
`\ctex_lsj_set_kanjiskip:N` 1646, 1945
`\ctex_lsj_set_math_letter:NN` 919, 931, 939
`\ctex_lsj_set_xkanjiskip:N` 1646, 2024
`\ctex_lsj_subst_font:` 780, 800
`\ctex_lsj_swap_cs:NN` 1435, 1442
`\ctex_lsj_switch_family:n`
..... 1062, 1063, 1073, 1130, 1158, 1212
`\ctex_lsj_update_mathfont:` 1312, 1315
`\ctex_lsj_update_mathfont:n` 1318, 1321, 1324
`\ctex_lsj_use_math_group:Nn` 905, 936
`\ctex_make_spa:nn` 4873
`\ctex_mono_jfm:n` 944, 952, 2094
`\ctex_package_end_hook:n` 5088, 5097
`\ctex_parse_name:NN` 1992, 5143, 5147
`\ctex_parse_name:NNn` 5148, 5152, 5209
`\ctex_patch_boot:NnnTF`
..... 5104, 5109, 5114, 5129, 5134, 5137
`\ctex_patch_cmd:Nnn`
..... 1686, 3144, 3150, 3156, 3223, 3226, 3230,
3236, 3246, 3254, 3257, 3262, 3265, 3922, 3923, 3924, 5112
`\ctex_patch_cmd_all:NnnTF` 5107
`\ctex_patch_cmd_once:NnnTF` 1680, 1994, 5102
`\ctex_patch_failure:N` 484, 524, 528, 532, 536,
1639, 1659, 1663, 1667, 1671, 1999, 3207, 3217, 5121, 5123
`\ctex_plane_to_utfxvibe:Nn` 554
`\ctex_preto_cmd:NnnTF` 521,
525, 529, 533, 1656, 1660, 1664, 1668, 3204, 3213, 5127
`\ctex_punct_map_bfseries:nn`
..... 628, 4149, 4161, 4167, 4252,
4258, 4330, 4331, 4337, 4442, 4443, 4444, 4450, 4503, 4509
`\ctex_punct_map_family:nn` 623, 4141, 4148,
4158, 4159, 4166, 4248, 4249, 4250, 4257, 4326, 4327,
4328, 4336, 4438, 4439, 4440, 4449, 4499, 4500, 4501, 4508
`\ctex_punct_map_itshape:nn` 645,
4160, 4168, 4251, 4259, 4329, 4338, 4441, 4451, 4502, 4510
`\ctex_punct_map_series:nnn` 632, 633, 636
`\ctex_punct_set:n` 606,
4136, 4165, 4247, 4256, 4325, 4335, 4437, 4448, 4498, 4507
`\ctex_punct_space:nn` 654
`\ctex_scheme_input:n` 79, 3365, 3641
`\ctex_select_size:` .. 2127, 2134, 2137, 2138, 2159, 2213
`\ctex_set_default_ccwd:Nn` 110, 199, 2133, 2158
`\ctex_set_font_size:Nnn` 3807, 3822,
3830, 3841, 3852, 3853, 3854, 3855, 3856, 3857, 3858,
3861, 3869, 3880, 3891, 3892, 3893, 3894, 3895, 3896, 3897
`\ctex_set_ignorespaces:` 597, 605
`\ctex_set_upfamily:nnn`
..... 1691, 4184, 4185, 4186, 4187, 4188,
4189, 4191, 4271, 4272, 4273, 4274, 4350, 4351, 4352,
4353, 4462, 4463, 4464, 4465, 4466, 4468, 4521, 4522, 4523
`\ctex_set_upfonts:nnnnnn`
..... 1718, 4179, 4267, 4346, 4458, 4517
`\ctex_set_upmap:nnn`
... 1704, 1720, 1721, 1722, 1723, 4190, 4192, 4467, 4469
`\ctex_set_zhmap:n` 430, 500, 1706
`\ctex_titleps_hook:` 3140, 3175, 3176
`\ctex_titlesec_hook:` 3048, 3112
`\ctex_update_ccglue:` 1937, 2178, 2197, 2241
`\ctex_update_ccwd:` 1923, 2172, 2249
`\ctex_update_default_family:` 47, 580, 1289, 1678, 1728
`\ctex_update_em_unit:` 1969, 2167, 2217
`\ctex_update_kanjisize:` 2062, 2064
`\ctex_update_parindent:` 2145, 2200, 2242
`\ctex_update_size:` 2139, 2140
`\ctex_update_stretch:` 2144, 2165
`\ctex_update_xkanjiskip:` 2016, 2036
`\ctex_update_ziju:` 2147, 2215
`\ctex_varioref_hook:` 3344, 3355, 3363
`\ctex_zhmap_case:nnn`
..... 134, 262, 267, 272, 4134, 4229, 4297, 4425, 4489
`\ctex_zhmap_input:n` 500, 503
`\ctex_zihao:n` 3724, 3725
`\ctex_ziju:n` 2209, 2210
ctex internal commands:
`\g_ctex_after_end_preamble_hook_tl` 5063, 5067, 5075
`_ctex_assign_heading_name:nnn` 2446
`\l_ctex_autoindent_tl`
..... 132, 189, 194, 199, 2126, 2130, 2133, 2202, 2205
`_ctex_calc_bounds:nn` 4901, 4902, 4906
`_ctex_ccglue:` 1954, 1967
`\l_ctex_ccglue_skip` 1941,
1945, 1948, 1950, 1958, 1962, 2176, 2187, 2196, 2232, 2240
`\c_ctex_class_tl`
... 84, 86, 410, 415, 420, 425, 3628, 3634, 3636, 3637, 3639
`_ctex_clear_fntef_color:n` ... 1862, 1869, 1873, 1888
`\c_ctex_cmap_encoding_seq` 440, 446
`\l_ctex_ctexcap_options_clist` 5045, 5046, 5055
`_ctex_def_heading_keys:n` 2398, 2476
`_ctex_default_pt:w` 119, 124
`\l_ctex_encoding_tl` 128, 222,
223, 488, 491, 582, 590, 1831, 2100, 2257, 3669, 3696, 3710
`\g_ctex_end_preamble_hook_tl` 5061, 5065, 5074
`\c_ctex_engine_file_str` 43, 2103
`\c_ctex_engine_str` 43

`\l_ctex_family_default_init_tl` 1731, 1751
`_ctex_family_default_wrap:n` 1734, 1752, 1755
`_ctex_fix_varioref_label:n` 3349
`\g_ctex_font_size_flag` 151,
162, 163, 164, 378, 382, 404, 3686, 3687, 3713, 3714, 3815
`\c_ctex_font_size_prop` 3727, 3738, 3782, 3809
`\c_ctex_font_size_seq` 3736, 3739, 3750
`\l_ctex_font_size_tl` 3727, 3728, 3774, 3775, 3809, 3810
`\g_ctex_fontset_tl` 226, 1763, 1781, 1806, 1813, 1814,
3933, 3936, 3939, 3940, 3946, 3948, 3968, 3970, 3975, 3992
`_ctex_get_font_sizes:Nn` 3774, 3777
`_ctex_get_macro_meaning:w` 5213
`\l_ctex_heading_bool` 305, 3624
`\l_ctex_heading_skip` 2489,
2496, 2498, 2500, 2501, 2512, 2513, 2516, 2518, 2519, 2522
`\c_ctex_headings_cs_seq` 3008, 3021, 3030
`\c_ctex_headings_seq` ... 2358, 2473, 3125, 3188, 3346
`_ctex_hookto_cmd:Nnnnw` 5129, 5134, 5292
`_ctex_hookto_cmd_parameter:Nnnnw` 5299, 5319
`_ctex_hookto_cmd_parameterless:Nnnnw` . 5298, 5304
`_ctex_initial_heading:n` 2368, 2475
`\g_ctex_kanjisize_func` 2042, 2052, 2063
`_ctex_left_hook_aux:nn` 5317
`\l_ctex_line_spread_fp`
..... 111, 182, 3688, 3689, 3715, 3716, 3901, 3903
`\l_ctex_line_stretch_tl` 2158, 2163, 2164, 2168
`\l_ctex_ltj_add_alternate_bool` 1142, 1154, 1172, 1179
`_ctex_ltj_alternate_cs:n`
..... 1018, 1021, 1022, 1023, 1057, 1058, 1060, 1145,
1148, 1149, 1151, 1152, 1376, 1413, 1475, 1500, 1540, 1557
`\l_ctex_ltj_alternate_family_tl` 1438, 1441
`\l_ctex_ltj_alternate_options_clist`
..... 1391, 1392, 1403, 1407
`\l_ctex_ltj_alternate_seq`
..... 988, 1360, 1361, 1369, 1375, 1383, 1631
`\l_ctex_ltj_base_CJKfamily_tl`
..... 989, 1003, 1055, 1475, 1500
`\l_ctex_ltj_base_family_tl` 65,
1019, 1050, 1056, 1419, 1460, 1519, 1538, 1543, 1544, 1556
`_ctex_ltj_change_encoding:` 953, 1037, 1433
`\l_ctex_ltj_char_range_clist` 987,
1344, 1353, 1372, 1390, 1402, 1414, 1437, 1459, 1610, 1612
`_ctex_ltj_char_range_parse_feature:w` . 1615, 1619
`\g_ctex_ltj_char_range_prop` 1467, 1574
`\l_ctex_ltj_char_range_tl`
..... 1467, 1470, 1576, 1578, 1580, 1582, 1589
`_ctex_ltj_check_family:n` 1013, 1374
`\l_ctex_ltj_current_font_tl`
..... 51, 747, 751, 752, 764, 806, 811
`\l_ctex_ltj_current_shape_tl` 829, 834, 839, 840
`_ctex_ltj_declare_char_range:nn` 1570, 1572
`\g_ctex_ltj_default_features_clist` 1007, 1255, 1256
`_ctex_ltj_family_csname:n` ... 1017, 1034, 1052, 1079
`\g_ctex_ltj_family_font_name_prop`
..... 994, 1010, 1015, 1090, 1166, 1292, 1304
`\g_ctex_ltj_family_font_options_prop` 995, 1010, 1169
`\g_ctex_ltj_family_int` . 1132, 1134, 1452, 1453, 1455
`\g_ctex_ltj_family_name_prop`
..... 1010, 1019, 1042, 1076, 1419
`_ctex_ltj_family_unknown_warning:n` 1070, 1088, 1548
`\l_ctex_ltj_font_options_clist` 991, 996,
997, 998, 1004, 1041, 1170, 1171, 1174, 1410, 1411, 1415
`_ctex_ltj_fontspec:nnn` 1133, 1140, 1160
`\g_ctex_ltj_fontspec_family_tl`
..... 1040, 1043, 1044, 1045, 1051, 1056, 1081
`\g_ctex_ltj_fontspec_prop` 1128, 1156, 1161
`_ctex_ltj_gset_family_cs:nn` 999, 1032
`_ctex_ltj_if_jfont_math:w` 872, 875
`\l_ctex_ltj_jfm_tl` 944, 965, 967, 1008
`\c_ctex_ltj_math_fam_int` 973, 974
`\c_ctex_ltj_math_family_tl`
..... 1326, 1328, 1330, 1333, 1337
`\c_ctex_ltj_math_tl`
..... 968, 969, 971, 973, 1251, 1317, 1327, 1332, 1336
`_ctex_ltj_msg_def_family_map:n` 1103, 1105
`_ctex_ltj_msg_family_map:n` 1031, 1102, 1116
`_ctex_ltj_noembed_wrap:n` 981, 983
`_ctex_ltj_pass_args:nnnn`
..... 1187, 1203, 1213, 1219, 1231, 1238, 1244, 1250
`_ctex_ltj_patch_external_font:w` 792, 825
`_ctex_ltj_pop_fontname:` 763, 772, 851
`_ctex_ltj_post_arg:w` 1190, 1196
`_ctex_ltj_push_fontname:n` 759, 767, 848
`\g_ctex_ltj_reset_alternate_prop`
..... 1024, 1542, 1554, 1562
`_ctex_ltj_save_alternate_shape:Nnn` 1474, 1499, 1515
`_ctex_ltj_save_fontname:w` 769, 773
`_ctex_ltj_select_font_aux:` 748, 754
`_ctex_ltj_set_alternate_family:n` 1046, 1053
`_ctex_ltj_tmp:w` 1349, 1366, 1444, 1446, 1447
`\l_ctex_ltj_tmp_seq` 1357, 1358, 1361, 1368
`\l_ctex_ltj_tmp_tl`
..... 990, 991, 1015, 1026, 1029, 1167, 1175,
1209, 1212, 1214, 1294, 1299, 1317, 1318, 1320, 1321,
1355, 1356, 1357, 1394, 1396, 1397, 1404, 1614, 1622, 1631
`\g_ctex_ltj_unknown_family_seq` ... 1092, 1094, 1099
`_ctex_ltj_update_family_aux:` 1277, 1279
`_ctex_ltj_update_family_uid:N` 997, 1410, 1449
`_ctex_ltj_use_global_options:N` 998, 1005, 1392, 1411
`\c_ctex_macosx_file_str` 1812, 1816
`\l_ctex_msyh_suffix_tl`
..... 4123, 4124, 4128, 4145, 4147, 4181
`\l_ctex_parameter_str` 5210, 5229, 5250, 5297
`\l_ctex_parameter_tl` 5250, 5255, 5266
`_ctex_parse_name:nNNNnN` 5162, 5171
`_ctex_parse_name:w` 5184, 5190, 5199
`_ctex_patch_cmd:Nnnnnw` 5104, 5109, 5114, 5268
`_ctex_patch_false:w` 5140, 5146, 5283, 5285, 5302, 5328
`_ctex_patch_rebuild:Nn` 5241, 5247, 5280, 5321, 5325
`_ctex_patch_replace:nnnTF` 5275, 5287
`_ctex_patch_rescan:NNn` 5249, 5250, 5251, 5259
`_ctex_patch_true:w` 5139, 5145, 5281, 5314, 5326

_ctex_plus_key_aux:nn	2428, 2430, 2431, 2432, 2433, 2434, 2435, 2438
\l_ctex_prefix_str	5210, 5228, 5249, 5306, 5307
\l_ctex_prefix_tl	5249, 5254, 5265
\l_ctex_punct_bounds_clist	4887, 4893, 4899, 4905
\c_ctex_punct_family_clist	608, 618
\l_ctex_punct_font	4885, 4886
\c_ctex_punct_seq	4865, 4888
\l_ctex_punct_tl	281, 577, 578, 664, 952, 2091, 2092, 2093, 2094
_ctex_rebuild_cmd:w	5241, 5242, 5246, 5321, 5322
\l_ctex_replacement_str	5210, 5230, 5251, 5289, 5290, 5324
\l_ctex_replacement_tl	5251, 5256, 5267
_ctex_restore_selectfont:	1985, 1989, 1998
_ctex_right_hook_aux:nn	5318
_ctex_save_bounds:n	4889, 4897
_ctex_save_cmap:Nn	459, 463
_ctex_save_font_size:nn	3738
\l_ctex_scheme_tl	316, 3365, 3641
\g_ctex_section_depth_flag	94, 310, 312, 3563, 3579, 3595
\c_ctex_section_headings_seq	2354, 2360, 2963, 3066
_ctex_set_font_size:nnNn	3810, 3813
\g_ctex_spa_iow	4875, 4878, 4880, 4890
\c_ctex_std_class_tl	3612, 3627, 3628, 3694, 3717
\g_ctex_std_options_clist	137, 383, 395, 405, 407, 411, 416, 421, 426
_ctex_titlesec_format:n	3062, 3070, 3073
_ctex_titlesec_format:Nn	3052, 3076, 3078
_ctex_titlesec_format_explicit:Nn	3053, 3088
_ctex_titlesec_hook:n	3062, 3066, 3068
_ctex_titlesec_spacing:Nn	3071, 3098
_ctex_titlesec_spacing:nnnnn	3099, 3100
_ctex_tmp:N	1981, 1992
_ctex_tmp:w	5150, 5203, 5214, 5235
\l_ctex_tmp_box	58, 1926, 1927, 3192, 3196
\l_ctex_tmp_dim	58, 2168, 2169, 2174, 2177, 2183, 2222, 2224, 2226, 2235, 2237
\l_ctex_tmp_int	58, 558, 560, 561, 563, 567, 569, 2182, 2184, 2185, 2191, 2192, 2224, 2225, 2226, 2229, 2230, 2235, 2237, 2962, 2965, 2969
\l_ctex_tmp_tl	58, 367, 374, 389, 390, 465, 466, 557, 559, 1983, 1994, 2400, 2458, 2471, 2481, 3693, 3694, 3782, 3783
\c_ctex_upper_case_file_str	1800, 1808
\c_ctex_version_tl	3, 6, 7, 8, 9, 20, 23
_ctex_write_family:nn	4877, 4882
\l_ctex_xkanjiskip_skip	2016
\l_ctex_xkanjiskip_tl	2023, 2028, 2029
\c_ctex_zero_tl	2142, 2149, 2151
\g_ctex_zhmCJK_bool	131, 261, 266, 271, 487, 581
\l_ctex_ziju_dim	2218, 2219, 2223, 2234, 2236, 2240, 2244
\l_ctex_ziju_tl	2142, 2150, 2151, 2212, 2218
\ctexbibname	2327, 2347
\CTEXcounter	2262
\CTEXdigits	26, 2272
\CTEXfilltwosides	31, 1896, 1906
\CTEXifname	16, 2504, 2760, 2942, 2950, 2958, 3183, 3225, 3228, 3232, 3244, 3248, 3256, 3259, 3264, 3267
\CTEXindent	31, 2245
\CTEXnoindent	31, 2245
\CTEXnumber	26, 2272
\CTEXnumberline	2956, 2976, 3440
\CTEXoptions	31, 2107
\ctexset	5, 2104, 2111, 2117
\CTEXsetfont	31, 2137
\CTEXsetup	31, 2107
\CTEXsout	31
\ctexspadef	654, 4892
\CTEXthechapter	16, 2707, 2951, 3118, 3244, 3264
\CTEXtheparagraph	16
\CTEXthepart	16, 2943, 3123
\CTEXthesection	16, 3225, 3232, 3248, 3256, 3267
\CTEXthesubparagraph	16
\CTEXthesubsection	16, 3228, 3259
\CTEXthesubsubsection	16
\CTEXunderdblline	31
\CTEXunderdot	31
\CTEXunderline	31
\CTEXunderwave	31
\CTEXxout	31
\CurrentOption	395
\cwd	2038
D	
declarecharrange	27, 1564
\DeclareFontEncoding	954
\DeclareFontFamily	473, 963, 4012, 4013, 4014, 4015, 4016, 4017
\DeclareFontShape	964, 966, 1695, 1696, 1699, 1700, 4028, 4029, 4030, 4031, 4032, 4033, 4034, 4035, 4036, 4039, 4040, 4041, 4042, 4043, 4044, 4045, 4046, 4047, 4051, 4052, 4053, 4054, 4055, 4056, 4060, 4061, 4062, 4063, 4064, 4065, 4066, 4067, 4068, 4071, 4072, 4073, 4074, 4075, 4076, 4077, 4078, 4079, 4083, 4084, 4085, 4086, 4090, 4091, 4092, 4093, 4094, 4095, 4096, 4097, 4098, 4101, 4102, 4103, 4104, 4105, 4106, 4107, 4108, 4109, 4113, 4114
\DeclareFontSubstitution	957
\DeclareKanjifamily	1693, 1694, 4019, 4020, 4021, 4022, 4023, 4024
\DeclareMathSizes	3775
\DeclareSymbolFont	969, 1327
\def	2568, 2605, 2637, 2655, 2667, 2698, 2738, 2747, 2771, 2790, 2816, 2820, 2855, 2870, 2884, 2885, 2889, 3835, 3846, 3874, 3885, 4557, 4593
\defaultCJKfontfeatures	1229
\defbeamertemplate	3273, 3292, 3311
\defbeamertemplatealias	3330, 3331, 3332
dim commands:	
\dim_abs:n	2236

`\dim_add:Nn` 2219
`\dim_compare:nNnTF` ... 112, 2169, 2204, 2220, 2227, 2488
`\dim_eval:n` 2189, 2235
`\dim_max:nn` 3195
`\dim_min:nn` 2236
`\dim_new:N` 61, 1936, 2039, 2040, 2041, 2244
`\dim_set:Nn` 1927, 1970, 1971,
 1972, 2168, 2174, 2205, 2218, 2222, 2249, 2495, 3193, 3905
`\dim_sub:Nn` 2226
`\dim_to_decimal:n` 37, 3746, 3747, 3784
`\dim_to_decimal_in_unit:nn` 4912
`\dim_use:N` 120, 3905
`\dim_zero:N` 2255
`\c_max_dim` 2169, 2488
`\c_zero_dim` 2177, 2189, 2204, 2220, 2227, 2413, 3596
`\dimexpr` 2587, 2642, 2753, 2777, 3280, 3299, 3318
`\directlua` 705, 706
`\document` 5068, 5071

E

`\edef` 4576, 4577
`\else` . 1682, 1683, 1687, 1688, 2540, 2549, 2558, 2574, 2579,
 2611, 2616, 2686, 2694, 2710, 2716, 2722, 2734, 2743,
 2801, 2806, 2825, 2830, 2849, 2854, 2863, 2883, 2899,
 2912, 3404, 3431, 4588, 4589, 4635, 4674, 4705, 4764, 4827

else commands:

`\else:` 490, 730, 823, 869, 1955, 3029, 3229
`\em` 1680
`\eminnershape` 1682, 1683, 1688, 1690
`\end` 3288, 3307, 3326, 4957
`\endCJKfilltwosides` 1902
`\endcsname` 2808,
 2817, 2818, 2819, 2826, 2844, 2845, 2847, 2850, 2859,
 2860, 2861, 2864, 4583, 4585, 4586, 4587, 4590, 4592, 4596
`\endCTEXfilltwosides` 1902, 1907
`\endgroup` 2846, 2882, 2907, 3290, 3309, 3328, 4576, 4584, 4599
`\endinput` 4581
`\endlinechar` 4555, 4556

etex commands:

`\etex_dimexpr:D` 120, 2183
`\etex_glueexpr:D` 2175
`\etex_iffontchar:D` 728
`\everypar` 2805, 2902, 2914
`\EverySelectfont` 2002, 2005

exp commands:

`\exp_after:wN`
 119, 776, 792, 794, 869, 872, 3099, 3728, 3775, 3810, 5219
`\exp_args:cc` 3340
`\exp_args:Nc` 3071, 3076, 5173
`\exp_args:Nco` 3181
`\exp_args:Nnc` 101
`\exp_args:NNf` 1622
`\exp_args:NNNo` 3904
`\exp_args:Nno` 3352
`\exp_args:No` 559, 1809, 3095, 4889, 5235
`\exp_args:Nx` 294, 587, 661, 2209, 3724, 3903

`\exp_last_unbraced:Nn` 5184
`\exp_last_unbraced:NNn` 1993
`\exp_not:N` 124, 125,
 469, 589, 590, 591, 738, 739, 741, 811, 847, 957, 1042,
 1043, 1044, 1045, 1401, 1522, 1739, 1740, 1741, 1755,
 1891, 1892, 1893, 1920, 2374, 2375, 2376, 2381, 2383,
 2384, 2385, 2386, 2389, 2390, 2393, 2405, 2406, 2407,
 2408, 2409, 2418, 2419, 2420, 2422, 2425, 2440, 2442,
 2444, 2968, 2970, 2971, 2972, 2973, 3083, 3084, 3093,
 3094, 3131, 3133, 3134, 3744, 3946, 5164, 5165, 5166, 5254
`\exp_not:n` 69,
 389, 740, 742, 1038, 1039, 1040, 1041, 1174, 1175, 1402,
 1403, 1404, 1428, 1522, 1734, 1756, 1976, 2426, 2481,
 3102, 3341, 5047, 5071, 5182, 5254, 5255, 5256, 5311, 5312
`\exp_stop_f:` 728, 729
`\expandafter`
 4558, 4559, 4583, 4584, 4585, 4586, 4587, 4592, 4596, 4597
`\ExplSyntaxOff` 66, 72, 478, 521, 525, 529, 533, 684,
 1656, 1660, 1664, 1668, 1681, 1995, 3204, 3272, 4918, 5117
`\ExplSyntaxOn` 71, 716, 3333, 4858

F

`\familydefault` 1275, 1286, 1737
`fancyhdr` 30, 342
`\fangsong` 4539
`\fi` .. 2531, 2536, 2542, 2551, 2560, 2563, 2578, 2583, 2591,
 2600, 2615, 2620, 2630, 2650, 2670, 2678, 2679, 2682,
 2686, 2696, 2714, 2720, 2726, 2737, 2746, 2750, 2758,
 2759, 2768, 2774, 2786, 2791, 2797, 2803, 2810, 2812,
 2834, 2835, 2851, 2865, 2868, 2887, 2896, 2915, 2916,
 3224, 3227, 3231, 3241, 3242, 3247, 3255, 3258, 3263,
 3266, 3284, 3303, 3322, 3404, 3431, 4588, 4589, 4598, 4851

fi commands:

`\fi:` 408, 537,
 574, 593, 732, 823, 869, 1955, 2000, 3032, 3233, 3249, 3819
`\figurename` 2308
`figurename` 12

file commands:

`\g_file_curr_name_str` 676, 1809
`\file_if_exist:nTF`
 . 48, 1796, 1798, 1800, 1812, 3939, 4118, 4125, 4127, 4375
`\file_if_exist_input:nTF` 86
`\file_input:n` 78, 87, 89, 510, 546
`fntef` 31, 342
`\font` 4012, 4013, 4014, 4015, 4016, 4017
`\fontfamily` 1268
`fontset` 7, 226, 3959
`\fontsize` 3728

fontspec commands:

`\g_fontspec_encoding_tl` 962
`\l_fontspec_family_tl` 1461
`\fontspec_set_family:Nnn` 1040, 1438
`\fontspec_visible_space:` 726
`\fontspec_visible_space_fallback:` 731

fontspec internal commands:

`_fontspec_fontname_wrap:n` 56, 981

\footnotesep 3904
 \footnotesize 3841, 3880, 3904
fp commands:
 \fp_compare:nNnTF 3688, 3715, 3901
 \fp_eval:n 2212, 4908
 \fp_set:Nn 3689, 3716
 \fp_use:N 2149, 3903
 \c_nan_fp 183
 \c_zero_fp 2149, 3688, 3715, 3901

G

GBK 6, 222
 \gdef 2994, 2995, 2996, 2997, 3000, 3001, 3002, 3003
 \global 2690, 2901, 2904
 \globaljfont 794
 \glueexpr 2841, 2856,
 2877, 2884, 2885, 2911, 3279, 3289, 3298, 3308, 3317, 3327
group commands:
 \group_begin: 473,
 721, 756, 778, 804, 846, 873, 921, 986, 1036, 1347, 1432,
 1517, 1537, 1553, 1733, 1980, 2043, 2380, 2388, 2456,
 3018, 3140, 3199, 3904, 4884, 5065, 5141, 5149, 5158, 5213
 \group_end: 477, 486, 725, 761,
 782, 817, 850, 877, 925, 1001, 1047, 1367, 1439, 1523,
 1549, 1560, 1745, 1993, 2061, 2391, 2395, 2480, 3033,
 3163, 3203, 3904, 4895, 5065, 5161, 5208, 5236, 5270, 5294
 \group_insert_after:N 814, 815
 \gtfamily 1682, 1687

H

hbox commands:
 \hbox_set:Nn 1926, 3192
 heading 9, 305
 \heiti 4538
 \hskip ... 2841, 2856, 2877, 2884, 2885, 2910, 2911, 3255, 3258
 \hspace 2573, 2610, 2943, 2951, 3123
 \Huge 2631, 2661, 2762, 2780, 3408, 3438, 3858, 3897
 \huge 2592,
 2628, 2644, 2757, 3390, 3407, 3411, 3436, 3444, 3857, 3896
 hyperref 31, 342
 \hypersetup 359, 360, 1822
 \hyphenchar 4012, 4013, 4014, 4015, 4016, 4017

I

if commands:
 \if_bool:N 487, 581
 \if_case:w 404, 3815
 \if_cs_exist:N 554, 1979
 \if_false: 107
 \if_meaning:w 1954
 \if_true: 106
 \ifctexpdf 105
 \ifdim 2795, 2837, 2873, 2891
 \IfNoValueF 2112, 2118
 \IfNoValueTF 1189, 1210, 1486, 1591, 1624, 1626, 2452

\ifnum 2569,
 2588, 2606, 2627, 2699, 2755, 2821, 2849, 2863, 3224,
 3227, 3231, 3238, 3239, 3247, 3255, 3258, 3263, 3266, 4588
 \ifodd 2536, 2538, 2563, 2570, 2588, 2600, 2607,
 2627, 2650, 2670, 2692, 2703, 2750, 2768, 2774, 2786,
 2799, 2810, 2826, 2838, 2874, 2892, 2896, 3282, 3301, 3320
 \ifx 4586, 4587, 4592
 \ifzhmappdf 4604, 4671, 4702, 4733, 4800
 \ignorespaces 2917
 \immediate 4595
 indent 30, 202
 \indexname 2311
 indexname 12
 \input 4857, 4921
 \insertcontinuationtext 2320, 2329, 2336
 \insertpart 3286, 3287
 \insertromanpartnumber 3277, 3377
 \insertsection 3305, 3306
 \insertsectionnumber 3296, 3475
 \insertsubsection 3324, 3325
 \insertsubsectionnumber 3315, 3504
int commands:
 \c_five 3747
 \c_four 312, 567, 569
 \int_compare:nNnTF
 560, 2185, 2229, 3563, 3579, 3595, 3686, 3713
 \int_const:Nn 973
 \int_div_truncate:nn 80, 567
 \int_eval:n 1487, 1489, 1490, 1592, 1594, 1595
 \int_from_hex:n 559
 \int_gincr:N 1132, 1452
 \int_incr:N 2965
 \int_mod:nn 569
 \int_new:N 59, 152, 1455
 \int_set:Nn 153, 544, 550, 558, 2182, 2224
 \int_sub:Nn 563, 2230
 \int_to_Hex:n 561, 566, 568
 \int_use:N ... 470, 550, 861, 1134, 1453, 2052, 2969, 4889
 \int_zero:N 2962
 \c_one 163, 1480, 1585, 2237, 2447, 4901
 \c_six 3747
 \c_three 310, 2229, 3579, 4902
 \c_two 95, 164, 378, 382, 2230, 3196, 3563, 3595
 \c_zero 162, 2185, 3687, 3714
 \interlinepenalty
 2585, 2624, 2640, 2658, 2761, 2779, 2842, 2843, 2878, 2879
iow commands:
 \iow_close:N 4878
 \iow_indent:n 5052
 \iow_new:N 4880
 \iow_now:Nn 4890
 \iow_open:Nn 4875
 \itemsep 3838, 3849, 3877, 3888
 \itshape 1682, 1683, 1687, 1688

J

\jfam 913, 974

K

\kaishu 4540

\kanjifamily 1657, 1661, 1665, 1669, 1674

\kanjifamilydefault 1677

kernel internal commands:

\l_kernel_expl_bool 70

keys commands:

\l_keys_choice_tl 295

\keys_define:nn 156, 389, 392, 978,
1340, 1449, 1526, 1564, 1603, 2066, 2087, 2120, 2154,
2277, 2303, 2331, 2341, 2344, 2347, 2481, 2978, 3912, 3959

\keys_if_exist:nnTF 1601

\l_keys_key_tl 141, 143, 145

\keys_set:nn 29, 172,
178, 234, 243, 252, 278, 288, 295, 302, 332, 338, 2105,
2112, 2118, 3368, 3423, 3451, 3483, 3518, 3535, 3549,
3565, 3573, 3581, 3589, 3596, 3597, 3599, 3665, 3667, 4001

\keys_set_known:nn 3644

\keys_set_known:nnN 990

L

\labelformat 3352

\languagealias 3673, 3678, 3699, 3704

\LARGE 3856, 3895

\Large 2589, 3388, 3394, 3462, 3466, 3855, 3894

\large 3487, 3854, 3893

\lastbox 2905

\leavevmode 2531, 2791

left commands:

\c_left_brace_str 5205

\leftmargin 3835, 3846, 3874, 3885

\leftmargini 3835, 3846, 3874, 3885

\let 1988, 2824, 2833, 3828, 3867, 4579, 4585

\linespread 3903

linespread 10, 182

linestretch 11, 2154

\linewidth 2183, 2191, 2223

\lishu 4542

\listfigurename 2306

listfigurename 12

\listtablename 2307

listtablename 12

\LoadClass 412, 417, 422, 427

\long 4593

\lstlistingname 739, 740

\lstlistlistingname 741, 742

\ltjalchar 729

\ltjdefcharrange 685, 686, 687, 690, 694, 695, 698, 701, 702

LTJFONTUID 1449

\ltjgetparameter 1931, 1958, 2020, 2034

ltjitaliccorr internal commands:

_ctex_ltjitaliccorr 1643

\ltjsetkanjiskip 1647

\ltjsetmathletter 923

\ltjsetparameter 704, 710, 1642

\ltjsetxkanjiskip 1649

lua commands:

\lua_escape_x:n
.. 822, 834, 861, 868, 1477, 1492, 1493, 1510, 1511, 1520

\lua_now_x:n 722, 796,
822, 826, 831, 836, 858, 867, 960, 1498, 1522, 1765, 2045

luatex commands:

\luatex_luafunction:D 2063

M

\MAKESPA 4881, 4923

\markboth 2503, 2594, 2622

\mathgroup 913

\mcfamily 1687

\mddefault 958, 964, 970, 1328, 1333

mode commands:

\mode_if_math:TF 907

msg commands:

\msg_critical:nnn 53, 676, 3928

\msg_error:nn 3979

\msg_error:nnn 19, 22, 38, 41, 667, 680, 683, 2293, 3729, 3811

\msg_error:nnnn 3945, 3974

\msg_fatal:nn 4863

\msg_new:nn 4861

\msg_new:nnn 140, 142, 144, 146, 148, 670,
1030, 1100, 1182, 1878, 2324, 3618, 3929, 3988, 5048, 5125

\msg_new:nnnn 12, 30, 45, 99, 2295, 3731, 3953, 3982

\msg_warning:nn 1177, 1882, 2084, 2095, 2346

\msg_warning:nnn
.... 170, 176, 204, 214, 229, 238, 247, 276, 286, 300,
330, 336, 344, 349, 355, 1095, 3637, 3916, 3966, 5054, 5124

\msg_warning:nnnn 1026, 1891, 1898, 2110, 2116, 2247, 2253

N

\newCJKfontfamily 1201

\NewDocumentCommand
1061, 1196, 1201, 1207, 1217, 1223, 1229, 1236, 1242,

1248, 1254, 1387, 1479, 1584, 1619, 1672, 2105, 2108,

2114, 2137, 2208, 2245, 2251, 2272, 2274, 2446, 3723,

3908, 4537, 4538, 4539, 4540, 4542, 4543, 4545, 4546, 4547

\newfontfeature 975, 976, 977

\newluafunction 2042

\newpage 2672, 2677

\ngostype 44, 1783

\nobreak 2589,

2597, 2647, 2757, 2762, 2765, 2780, 2783, 2893, 3389, 3437

\nobreakspace 2589, 2628

nocap 30

\noexpand 4577, 4579, 4580, 4581

nofonts 30

\noindent 2923

noindent 30

nopunct 30

\normalem 1861

\normalfont 533, 535, 536, 1233, 1240,
1246, 1311, 1668, 1670, 1671, 1679, 2586, 2587, 2625,
2626, 2641, 2642, 2659, 2660, 2752, 2753, 2776, 2777, 2973
\normalsize 3521, 3538, 3552, 3822, 3861, 3899, 3907
nospace 30
nozhmap 30
\null 2561, 2675
\numberline 2709, 2850, 2864, 2951, 2959, 3134, 3190

O

\onecolumn 2556
or commands:
 \or: 406, 3817

P

\pagestyle 3664
\par 2494, 2532, 2589, 2628, 2631, 2644,
2661, 2757, 2762, 2780, 2792, 2893, 3278, 3279, 3286,
3289, 3297, 3298, 3305, 3308, 3316, 3317, 3324, 3327,
3370, 3389, 3400, 3418, 3426, 3437, 3478, 3480, 3512, 3514
paragraph/afterskip 32
paragraph/beforeskip 32
paragraph/hang 22
paragraph/numbering 15
paragraph/runin 20
\parindent
2204, 2205, 2227, 2248, 2249, 2254, 2255, 2586, 2587,
2641, 2642, 2752, 2753, 2776, 2777, 3280, 3299, 3318, 3597
\parsep 3837, 3838, 3848, 3849, 3876, 3877, 3887, 3888
\part 2529
part/beforeskip 32
part/fixbeforeskip 32
part/numbering 15
part/pagestyle 22, 2456
\partmark 2502, 2595, 2623
\partname 2589, 2628, 3277, 3372
\PassOptionsToClass 411, 416, 421, 426
\PassOptionsToPackage 488, 1826, 2257, 5056
\pdfmapline 4605,
4606, 4607, 4608, 4609, 4610, 4611, 4613, 4614, 4615,
4616, 4617, 4618, 4619, 4621, 4622, 4623, 4624, 4625,
4626, 4628, 4629, 4630, 4631, 4632, 4633, 4734, 4735,
4736, 4737, 4738, 4739, 4740, 4742, 4743, 4744, 4745,
4746, 4747, 4748, 4750, 4751, 4752, 4753, 4754, 4755,
4757, 4758, 4759, 4760, 4761, 4762, 4801, 4802, 4803,
4804, 4805, 4806, 4808, 4809, 4810, 4811, 4812, 4813,
4815, 4816, 4817, 4818, 4819, 4821, 4822, 4823, 4824, 4825
\pdfoutput 4588
\pdfstringdefDisableCommands 3911
pdftex commands:
 \pdftex_pdffontattr:D 469
 \pdftex_pdflastobj:D 470
 \pdftex_pdfobj:D 466
\pingfang 4547
prg commands:
 \prg_do_nothing:
 917, 1426, 1518, 1907, 1978, 2071, 2935, 5246

\prg_generate_conditional_variant:Nnn 1087
\prg_new_conditional:Npnn 820, 1951
\prg_new_protected_conditional:Npnn 1074, 5216
\prg_return_false: ... 823, 1084, 1955, 1959, 1963, 5226
\prg_return_true: 823, 1077, 1082, 1955, 1959, 1963, 5231
\primitive 4957
\ProcessKeysOptions 401
\proofname 2298, 2313
proofname 12
prop commands:
 \prop_get:NnN 1169
 \prop_get:NnNTF
 1076, 1128, 1166, 1419, 1467, 3727, 3782, 3809
 \prop_gpop:NnNTF 1015, 1019, 1554
 \prop_gput:Nnn ... 994, 995, 1042, 1156, 1542, 1578, 3744
 \prop_gremove:Nn 1024
 \prop_if_empty:NTF 1090, 1292
 \prop_map_break:n 1306
 \prop_map_inline:Nn 1304
 \prop_new:N 1010, 1011, 1012, 1161, 1562, 1583, 3738
\protect 2516, 2522, 2709,
2729, 2730, 2850, 2864, 2951, 2959, 3133, 3145, 3147,
3151, 3153, 3157, 3159, 3167, 3172, 3184, 3185, 5177, 5178
\ProvideDocumentCommand 2502
\ProvidesFile 4593, 4601, 4668, 4699, 4730, 4797
\providetranslation
 4962, 4963, 4964, 4965, 4966, 4967, 4968,
4969, 4970, 4971, 4972, 4973, 4974, 4975, 4976, 4977,
4978, 4979, 4980, 4981, 4982, 4983, 4984, 4985, 4986,
4987, 4988, 4989, 4990, 4991, 4992, 4993, 4994, 4995,
4996, 4997, 4998, 4999, 5000, 5001, 5002, 5003, 5004,
5005, 5006, 5007, 5008, 5009, 5010, 5011, 5012, 5013,
5014, 5015, 5016, 5017, 5018, 5019, 5020, 5021, 5022,
5023, 5024, 5025, 5026, 5027, 5028, 5029, 5030, 5031,
5032, 5033, 5034, 5035, 5036, 5037, 5038, 5039, 5040, 5041

ptex commands:

\ptex_kanjiskip:D 1934, 1948, 1962
\ptex_xkanjiskip:D 2021, 2025, 2035
punct 10, 30, 281, 2087
\punctstyle 578, 2092

Q

\quad 2817,
3224, 3225, 3227, 3228, 3231, 3232, 3244, 3248, 3256,
3259, 3264, 3267, 3395, 3445, 3455, 3488, 3522, 3539, 3553

quark commands:

\q_mark 5220, 5223
\q_stop 120, 125, 872,
875, 1384, 1388, 1615, 1620, 5185, 5190, 5199, 5220, 5223
\quotation 3921

R

\raggedright 2586, 2641, 2752, 2776, 3387, 3435
\refname 2314, 2319, 2326, 2335, 2343, 2344
refname 12
\refstepcounter ... 2572, 2609, 2705, 2828, 3335, 3357, 3358

`\relax` 2587, 2606, 2627, 2642, 2753, 2777, 2793,
 2829, 2836, 2841, 2856, 2872, 2877, 2884, 2885, 2890,
 2909, 2911, 3255, 3258, 3279, 3280, 3289, 3298, 3299,
 3308, 3317, 3318, 3327, 4551, 4555, 4558, 4586, 4587, 4592
`\renewcommand` 2530, 2546, 2685
`\RenewDocumentCommand` 1268, 3358
`\RequirePackage` 27,
 56, 57, 351, 362, 489, 492, 493, 494, 660, 678, 681,
 718, 738, 1726, 1835, 1844, 1860, 1866, 2258, 3718, 5101
`\RequirePackageWithOptions` 5057
`resetalternatfont` 28, 1526
reverse commands:
`\reverse_if:N` 581
`\rmdefault` 1283, 1739
`\rmfamily` 521, 524, 1656, 1659

S

scan commands:
`\scan_stop:`
 ... 65, 120, 794, 913, 1519, 1520, 2175, 2183, 4885, 5067
`scheme` 9, 316
`\scriptsize` 3852, 3891
`\secdef` 2543, 2565, 2697
`section/afterskip` 32
`section/beforeskip` 32
`section/hang` 22
`section/numbering` 15
`section/runin` 20
`\sectionname` 3296, 3472
`\selectfont` 1068, 1545, 1558, 1675, 1992, 1999, 2013, 2139, 3728
seq commands:
`\seq_clear:N` 988
`\seq_concat:NNN` 1360
`\seq_const_from_clist:Nn` . 440, 2355, 2365, 3008, 4865
`\seq_gput_left:Nn` 2361, 2362
`\seq_gput_right:Nn` 1094, 3750
`\seq_gset_eq:NN` 2360
`\seq_if_empty:NTF` 1375
`\seq_if_in:NnTF` 446, 1092
`\seq_map_function:NN` 3066, 3188
`\seq_map_inline:Nn`
 ... 1383, 2473, 2963, 3011, 3021, 3030, 3125, 3346, 4888
`\seq_new:N` 1099, 1368, 1369, 2359, 3739
`\seq_put_right:Nn` 1631
`\seq_set_filter:NNn` 1358
`\seq_set_split:Nnn` 1357
`\seq_use:Nnnn` 3736
`\setbox` 2905
`\setCJKfamilyfont` .. 1113, 1201, 4146, 4152, 4153, 4154,
 4155, 4156, 4157, 4200, 4201, 4209, 4210, 4212, 4213,
 4214, 4215, 4217, 4239, 4241, 4243, 4245, 4284, 4285,
 4286, 4287, 4311, 4316, 4321, 4323, 4365, 4367, 4369,
 4370, 4389, 4390, 4391, 4392, 4411, 4415, 4416, 4417,
 4418, 4431, 4432, 4433, 4434, 4435, 4436, 4477, 4478,
 4479, 4480, 4481, 4482, 4495, 4496, 4497, 4530, 4531, 4532

`\setCJKmainfont` 1109, 1229, 4137, 4196, 4204,
 4231, 4277, 4299, 4356, 4386, 4403, 4427, 4472, 4491, 4526
`\setCJKmathfont` 1229
`\setCJKmonofont` 1111, 1229, 4151, 4199, 4208,
 4238, 4283, 4310, 4364, 4388, 4410, 4430, 4476, 4494, 4529
`\setCJKromanfont` 1235, 1261
`\setCJKsansfont` 1110, 1229, 4140, 4144, 4198, 4206,
 4237, 4282, 4305, 4362, 4387, 4409, 4429, 4474, 4493, 4528
`\SetSymbolFont` 971, 1332, 1336
`\sfdefault` 1284, 1740
`\sffamily` 525, 528, 1660, 1663
skip commands:
`\skip_horizontal:N` 1941
`\skip_if_eq:nnTF` 1958, 1962, 2019, 2513, 2519
`\skip_new:N` 1950, 2032, 2498
`\skip_set:Nn` 1931,
 1934, 2023, 2033, 2176, 2187, 2232, 2240, 2500, 2512, 2518
`\skip_set_eq:NN` 1948, 2025
`\skip_sub:Nn` 2489, 2496
`\skip_use:N` 2516, 2522
`\skip_zero:N` 2196
`\c_zero_skip` 2411, 2412, 2464, 2465, 2513, 2519
`\small` 3830, 3869
`\songti` 4537
`\space` 2706, 2757, 3372, 3433, 3472, 3503, 3605
`space` 10, 30, 291, 2066
`\special` 1708, 1709, 1712,
 1713, 1842, 1856, 4636, 4637, 4638, 4639, 4640, 4641,
 4642, 4644, 4645, 4646, 4647, 4648, 4649, 4650, 4652,
 4653, 4654, 4655, 4656, 4657, 4659, 4660, 4661, 4662,
 4663, 4664, 4675, 4676, 4677, 4678, 4679, 4681, 4682,
 4683, 4684, 4685, 4687, 4688, 4689, 4690, 4692, 4693,
 4694, 4695, 4706, 4707, 4708, 4709, 4710, 4712, 4713,
 4714, 4715, 4716, 4718, 4719, 4720, 4721, 4723, 4724,
 4725, 4726, 4765, 4766, 4767, 4768, 4769, 4770, 4771,
 4773, 4774, 4775, 4776, 4777, 4778, 4779, 4781, 4782,
 4783, 4784, 4785, 4786, 4788, 4789, 4790, 4791, 4792,
 4793, 4828, 4829, 4830, 4831, 4832, 4834, 4835, 4836,
 4837, 4838, 4840, 4841, 4842, 4843, 4845, 4846, 4847, 4848
`\SplitArgument` 1480, 1585, 2447
str commands:
`\c_backslash_str` 5204
`\c_colon_str` 899, 900, 901
`\str_case:nnTF` 1737, 1783, 5175, 5190
`\str_case_x:nn` 1281
`\str_case_x:nnTF` 1107, 1118
`\str_const:Nn` 43, 50, 1808, 1816
`\str_if_empty:NTF` 5261, 5297, 5306
`\str_if_eq:nnTF` 491, 582,
 946, 1275, 1831, 1839, 3669, 3696, 3710, 3968, 3985, 3992
`\str_if_eq_x:nnTF` 577, 1296, 3965, 5181
`\str_lower_case:n` 465
`\str_new:N` 5101, 5210, 5211, 5212
`\str_set:Nn` 5228, 5229, 5230
`\str_upper_case:n` 1809
`\strutbox` 3905

sub3section 9, 308
 sub4section 9, 308
 subparagraph/afterskip 32
 subparagraph/beforeskip 32
 subparagraph/hang 22
 subparagraph/numbering 15
 subparagraph/runin 20
 subsection/afterskip 32
 subsection/beforeskip 32
 subsection/hang 22
 subsection/numbering 15
 subsection/runin 20
 \subsectionname 3315, 3503
 subsubsection/afterskip 32
 subsubsection/beforeskip 32
 subsubsection/hang 22
 subsubsection/numbering 15
 subsubsection/runin 20
 sys commands:
 \c_sys_engine_str 44
 \sys_if_engine_pdftex:TF
 130, 4132, 4224, 4292, 4380, 4397, 4423, 4487
 \sys_if_engine_uptex:TF
 4177, 4265, 4344, 4383, 4400, 4456, 4515
 \sys_if_engine_xetex:TF 4475, 4859
 \sys_if_output_pdf:TF 105, 475, 1834, 4226, 4294

T

\tablename 2309
 tablename 12
 \TeX 3909

T_EX and L^AT_EX 2_ε commands:

@@italiccorr 1644
 @@par 2842, 2878, 3456, 3489, 3523, 3567, 3583
 @afterheading 2602, 2652, 2736, 2745, 2898
 @afterindentfalse 2541, 2691, 2695, 2796, 2802
 @afterindenttrue 2539, 2693, 2794, 2800
 @Alph 3601, 3606
 @chapapp 2706, 2757, 3240, 3263
 @chapter 2697, 2698
 @chinese 2260, 2269
 @clubpenalty 2913
 @currentHref 3042
 @currentlabel 3337, 3341
 @currentxt 4, 5047, 5096
 @currname 4, 5047, 5097
 @dblarg 2815
 @defaultunits 37
 @empty 101, 2824, 2833
 @endpart 2634, 2664, 2666
 @EverySelectfont@Init 1979, 1987
 @firstoftwo 3184
 @hangfrom 2922
 @ifclassloaded 3614
 @ifpackagelater 37, 40, 666, 679, 682, 1867
 @ifpackageloaded 18, 21, 675, 1819, 3105, 3208, 3354, 5078

\@ifpackagewith 3050, 3059, 3110
 \@ifstar 2813
 \@listI 3828, 3867
 \@listi 3828, 3835, 3846, 3867, 3874, 3885
 \@M 2585,
 2624, 2640, 2658, 2761, 2779, 2842, 2843, 2878, 2879, 2906
 \@makechapterhead 2733, 2735, 2747
 \@makeschapterhead 2742, 2744, 2771
 \@minus 3457, 3490, 3524, 3540, 3554, 3824,
 3826, 3832, 3834, 3836, 3837, 3843, 3845, 3847, 3848,
 3863, 3865, 3871, 3873, 3875, 3876, 3882, 3884, 3886, 3887
 \@namedef 3145, 3147, 3151, 3153, 3157, 3159, 3167, 3172
 \@nameuse 3145, 3147, 3157, 3159, 3167
 @ne 3103, 3227, 3258
 @nil 847
 @nobreakfalse 2900
 @noskipsecfalse 2904
 @noskipsectrue 2901
 @onlypreamble 37, 437, 502,
 1257, 1258, 1259, 1260, 1261, 1262, 1703, 1717, 1725, 3952
 @part 2543, 2565, 2567
 @pkgextension 6, 7, 8, 9, 20, 23, 677, 5096
 @plus 3401,
 3402, 3457, 3458, 3490, 3491, 3524, 3525, 3540, 3554,
 3568, 3584, 3824, 3825, 3826, 3832, 3833, 3834, 3836,
 3837, 3843, 3844, 3845, 3847, 3848, 3863, 3864, 3865,
 3871, 3872, 3873, 3875, 3876, 3882, 3883, 3884, 3886, 3887
 @popfilename 133, 5094
 @schapter 2697, 2738
 @seccntformat 2816, 2829
 @secondoftwo 3185
 @secpenalty 2807, 3460, 3493, 3527, 3541, 3555
 @sect 2815, 2820
 @setfontsize 3814
 @spart 2543, 2565, 2636
 @ssect 2814, 2870
 @startsection 93, 95, 102, 2790, 2968
 @svsec 2824, 2829, 2833, 2841, 2857, 2858
 @svsechd 2855, 2884, 2885, 2907
 @tempdima 3193, 3195
 @tempskipa 2793, 2795, 2796,
 2807, 2836, 2837, 2872, 2873, 2890, 2891, 2894, 2909, 2910
 @tempswafalse 2559
 @tempswattrue 2557
 @topnewpage 2733, 2742
 @topnum 2690
 @xsect 2869, 2888, 2889
 @zhdig 2268
 @zhnum 2260, 2267
 \abstractname 12
 \addCJKfontfeature 65
 \AfterEndPreamble 132
 \algorithmname 12
 \appendixname 12, 24
 \AtBeginDocument 37, 46
 \AtBeginDvi 43

<code>\AtBeginShipoutFirst</code>	43	<code>\CTEX@chapter@afterskip</code>	2767, 2785
<code>\AtEndOfClass</code>	38	<code>\CTEX@chapter@aftertitle</code>	2764, 2782
<code>\AtEndOfPackage</code>	38	<code>\CTEX@chapter@beforeskip</code>	2749, 2773
<code>\AtEndOfPackageFile*</code>	133	<code>\CTEX@chapter@break</code>	2688
<code>\AtEndPreamble</code>	132	<code>\CTEX@chapter@fixskip</code>	2750, 2768, 2774, 2786
<code>\baselineskip</code>	10, 11, 76, 80	<code>\CTEX@chapter@format</code>	2754, 2778
<code>\bfseries</code>	48	<code>\CTEX@chapter@indent</code>	2753, 2777
<code>\bibname</code>	12	<code>\CTEX@chapter@lofskip</code>	2462
<code>\c@chapter</code>	3606	<code>\CTEX@chapter@lotskip</code>	2463
<code>\c@secnumdepth</code>	2569, 2588, 2606, 2627, 2699, 2755, 2821, 2849, 2863, 3224, 3227, 3231, 3238, 3239, 3247, 3255, 3258, 3263, 3266	<code>\CTEX@chapter@numbering</code>	2703, 3003
<code>\c@section</code>	3601	<code>\CTEX@chapter@pagestyle</code>	2461, 2689
<code>\caption</code>	31	<code>\CTEX@chapter@titleformat</code>	2763, 2781
<code>\catcode</code>	36, 45, 45, 124, 133	<code>\CTEX@chapter@tocline</code>	2940
<code>\ccwd</code> ...	11, 11, 25, 26, 31, 31, 37, 74, 75, 76, 79, 79, 80, 80	<code>\CTEX@chaptername</code>	2760
<code>\chapter</code>	9, 15, 16, 22, 24, 24, 86, 106	<code>\CTEX@document@left@hook</code>	5064, 5070
<code>\chaptermark</code>	13	<code>\CTEX@document@right@hook</code>	5066, 5072
<code>\chaptername</code>	15	<code>\CTEX@Family@CMap</code>	451, 479, 482
<code>\char</code>	50	<code>\CTEX@fixheadingskip</code>	2487, 2492, 2536, 2600, 2650, 2670, 2768, 2786, 2810, 2896
<code>\CheckCommand</code>	76	<code>\CTEX@fixskip</code>	2810, 2896, 2930, 2937
<code>\Chinese</code>	31	<code>\CTEX@fixtopskip</code>	2484, 2563, 2750, 2774
<code>\chinese</code>	26, 31, 82	<code>\CTEX@fontfamily</code>	1039, 1268
<code>\CJK@@ignorespaces</code>	47, 595	<code>\CTEX@hang</code>	2921, 2931, 2938
<code>\CJK@encoding</code>	55, 57, 64, 753, 757, 841, 953, 1327, 1330, 1332, 1336, 1477, 1492, 1493, 1510, 1511	<code>\CTEX@hangfrom</code>	2841, 2877, 2919
<code>\CJK@envStart</code>	46, 46, 47, 589	<code>\CTEX@headingskip</code>	2501, 2537, 2564, 2601, 2651, 2671, 2751, 2769, 2775, 2787, 2811, 2897
<code>\CJK@family</code>	50, 745, 758, 805, 841, 1065, 1544, 1554, 1556, 2013, 2014	<code>\CTEX@hyperheadinghook</code>	3016, 3046
<code>\CJK@ignorespaces</code>	47, 47, 600, 604	<code>\CTEX@ifnamefalse</code>	2504, 2575, 2580, 2612, 2617, 2638, 2656, 2711, 2717, 2723, 2739, 2822, 2831, 2871
<code>\CJK@input</code>	46, 538	<code>\CTEX@ifnametrue</code>	2504, 2571, 2608, 2704, 2827
<code>\CJK@loadBinding</code>	45, 46, 585	<code>\CTEX@makeanchor</code>	2576, 2581, 2613, 2618, 2712, 2832, 3006, 3014, 3026, 3031, 3037
<code>\CJK@loadEncoding</code>	45	<code>\CTEX@makeanchor@chapter</code>	2718, 2724
<code>\CJK@makeActive</code>	46, 584	<code>\CTEX@makeanchor@schapter</code>	2740
<code>\CJK@plane</code>	46, 76, 454, 455, 465, 2002	<code>\CTEX@makeanchor@sect</code>	2823
<code>\CJK@surr</code>	46, 554	<code>\CTEX@makeanchor@spart</code>	2639, 2657
<code>\CJK@upperReset</code>	47	<code>\CTEX@makeanchor@ssect</code>	2814
<code>\CJKfamily</code>	57	<code>\CTEX@part@afterindent</code>	2538
<code>\CJKfamilydefault</code>	46, 47, 61, 61, 69, 70, 70, 70, 70	<code>\CTEX@part@afterskip</code>	2590, 2629, 3283
<code>\CJKglue</code>	80, 111	<code>\CTEX@part@afterskip</code>	2599, 2649, 2669, 3289
<code>\CJKhook</code>	47, 47	<code>\CTEX@part@aftertitle</code> ...	2596, 2633, 2646, 2663, 3287
<code>\CJKrmdefault</code>	44	<code>\CTEX@part@beforeskip</code>	2535, 2562, 3279
<code>\CJKsymbol</code>	76	<code>\CTEX@part@break</code>	2533, 2552
<code>\CJKunderdotbasesep</code>	31	<code>\CTEX@part@fixskip</code>	2536, 2563, 2600, 2650, 2670
<code>\CJKunderline</code>	31	<code>\CTEX@part@format</code>	2587, 2626, 2643, 2660, 3281
<code>\clearpage</code>	47	<code>\CTEX@part@indent</code>	2587, 2642, 3280
<code>\contentsname</code>	11, 12	<code>\CTEX@part@numbering</code> ...	2570, 2588, 2607, 2627, 3282
<code>\CS</code>	51, 52	<code>\CTEX@part@pagestyle</code>	2460, 2554
<code>\CTEX@addloflotskip</code>	2509, 2731	<code>\CTEX@part@titleformat</code> ..	2593, 2632, 2645, 2662, 3287
<code>\CTEX@addtocline</code> ...	2526, 2584, 2621, 2727, 2853, 2867	<code>\CTEX@part@tocline</code>	2940
<code>\CTEX@afterindent</code>	2799, 2929, 2936	<code>\CTEX@partname</code>	2590, 2629, 3283
<code>\CTEX@aftertitle</code>	2881, 2886, 2928, 2935	<code>\CTEX@postappendix</code>	2988, 2996, 3002
<code>\CTEX@appendix@number</code>	2983, 2995, 3001	<code>\CTEX@postchapter</code>	3002
<code>\CTEX@appendix@numbering</code>	2984, 2997, 3003	<code>\CTEX@postsection</code>	2996
<code>\CTEX@chapter@afterindent</code>	2692	<code>\CTEX@preappendix</code>	2987, 2994, 3000
<code>\CTEX@chapter@aftername</code>	2760	<code>\CTEX@prechapter</code>	3000

\CTEX@presection	2994	\do@subst@correction	51, 52, 780
\CTEX@runin	2838, 2874, 2892, 2932, 2939	\em	69
\CTEX@save@appendix	2989, 2992	\escapechar	54
\CTEX@save@refstepcounter	3357, 3360	\EverySelectfont	76
\CTEX@section@aftername	3302	\external@font	791, 792, 794
\CTEX@section@afterskip	3308	\extract@font	51, 779
\CTEX@section@aftertitle	3306	\f@baselineskip	2175
\CTEX@section@beforeskip	3298	\f@encoding	757, 797, 910
\CTEX@section@format	3300	\f@family	51, 758, 797, 805, 1270, 1281
\CTEX@section@indent	3299	\f@series	753, 841
\CTEX@section@numbering	2997, 3301	\f@shape	753, 841
\CTEX@section@titleformat	3306	\f@size	753, 759, 836, 844, 847, 848, 1927, 1970, 3196
\CTEX@sectionname	3302	\fangsong	8
\CTEX@selectfont@hook	1973, 2002, 2005	\figurename	12
\CTEX@selectfont@save	1984, 1988	\font	52
\CTEX@setcurrentlabel@n	3335, 3361	\font@name	51, 52, 762, 769, 770, 773, 776, 794, 795, 812, 890, 891, 892, 893, 894, 901
\CTEX@setheadingskip	2499, 2535, 2562, 2599, 2649, 2669, 2749, 2767, 2773, 2785, 2809, 2895	\fontcharwd	75
\CTEX@subsection@aftername	3321	\fontfamily	61
\CTEX@subsection@afterskip	3327	\footnotesep	111, 111
\CTEX@subsection@aftertitle	3325	\footnotesize	111
\CTEX@subsection@beforeskip	3317	\get@external@font	788
\CTEX@subsection@format	3319	\getanddefine@fonts	879
\CTEX@subsection@indent	3318	\globaljfont	51
\CTEX@subsection@numbering	3320	\H@old@chapter	3020
\CTEX@subsection@titleformat	3325	\hbox	86
\CTEX@subsectionname	3321	\heiti	8
\CTEX@thechapter	3001	\hrule	86
\CTEX@thesection	2995	\Hy@chapapp	2712, 2718, 2724, 2740
\CTEX@titleformat@n	93, 2880, 2886, 2927, 2934	\Hy@driver	1839
\CTEX@titlepslabel@clear	3154, 3164	\Hy@MakeCurrentHrefAuto	3039
\CTEX@titlepslabel@set	3148, 3160, 3164	\Hy@org@chapter	3020
\CTEX@toc@width@n	3190, 3205, 3215	\Hy@raisedlink	3040
\CTEX@todayold	2276, 2281	\Hy@unicodetrue	73
\CTEX@update@sectionformat@n	2798, 2925	\hyper@anchorend	3043
\CTEX@verbatim@font@hook	1637, 1641	\hyper@anchorstart	3042
\ctex@zhmap@endinput	4853	\HyPsd@ConvertToUnicode	73
\CTEXdigits	26	\HyPsd@LoadUnicode	73
\CTEXifname	16	\HyPsd@pdfencoding	73
\CTEXnumber	26, 26	\HyPsd@ToBigChars	73
\CTEXnumberline	23	\if@mainmatter	2701, 2756, 3238, 3632, 3633
\CTEXoptions	31, 31, 78	\if@nobreak	2804
\ctexset	5, 5, 7, 11, 14, 27, 28, 31, 31, 112	\if@noskipsec	2531, 2791, 2903
\CTEXsetup	31	\if@openright	2547, 2674, 2686, 3404, 3431
\CTEXthechapter	13, 16, 96	\if@tempwa	2680
\CTEXthesection	13, 16, 99	\if@twocolumn	2555, 2732, 2741
\CTEXunderdotbasesep	31	\if@twoside	2673, 3222, 3245
\CTEXunderline	31	\iffalse	54
\curr@fontshape	759, 789, 798, 802, 848	\ifHy@implicit	3019
\DeclareAlternateKanjiFont	64	\ifin@	54, 869
\DeclareFontEncoding	54	\iftrue	54
\DeclareFontFamily	43, 52	\indexname	12
\DeclareFontShape	52	\insertcontinuationtext	12
\DeclareFontShape@	64, 1436	\itdefault	52
\DeclareRobustCommand	134	\itshape	48
\define@newfont	51, 781	\kaishu	8

\kanjiskip	76	\pickup@jfont	785
\labelformat	101	\pingfang	8
\leftmark	13	\ProcessKeysOptions	29
\linewidth	80	\proofname	12, 82
\lishu	8	\protected	97
\listfigurename	12	\protected@edef	2273, 2275, 2829, 3337
\listtablename	12	\ProvidesFile	124
\ltj@@does@alt@set	52	\ps@fancy	3251
\ltj@@getjfontnumber	53, 857	\ps@headings	3221
\ltj@@set@stackfont	899, 900, 901	\refname	12, 12
\ltj@allalchar	723	\refstepcounter	101
\ltj@curjfont	52, 76	\relax	46, 79
\ltj@pickup@altfont@auxy	842	\reset@font	535, 1670
\ltj@pickup@altfont@copy	855	\rightmark	13
\ltj@setpar@global	68, 898, 1647, 1649	\rmfamily	49
\ltj@tempcntc	53, 861	\roman	26
\ltjalchar	50	\scantokens	134
\ltjsetkanjiskip	68	\scriptfont@name	893, 900
\ltjsetparameter	68	\section . 9, 14, 15, 20, 20, 21, 22, 22, 23, 24, 24, 32, 83, 99	
\ltjsetxkanjiskip	68	\section*	15
\m@ne	2569, 2588, 2699, 2755, 3231, 3238, 3239, 3263, 4012, 4013, 4014, 4015, 4016, 4017	\sectionmark	13
\markboth	13	\sectionname	15
\markright	13	\sectionpage	14, 100
\math@bgroup	909	\selectfont	53, 70, 76, 76, 79
\math@egroup	914	\setCJKfamilyfont	27, 27
\math@fonts	896	\setCJKmainfont	27
\mathbb	111	\sf@size	892
\maxdimen	11, 79	\sffamily	49
\meaning	134, 135, 136, 136	\size@update	79, 80, 1996, 1997, 2139, 2153
\newcommand	134	\sldefault	52
\newrobustcmd	134	\songti	8
\newtitlemark	98	\special	7, 72
\nfss@catcodes	55	\split@name	847
\ngostype	71	\SplitArgument	85
\normalsize	8, 107	\ssf@size	894
\nouppercase	46	\strutbox	111
\p@	1927, 1970, 2628, 2729, 2730, 2748, 2757, 2766, 2772, 2784, 3196, 3400, 3427, 3428, 3429, 3430, 3437, 3824, 3825, 3826, 3832, 3833, 3834, 3836, 3837, 3843, 3844, 3845, 3847, 3848, 3863, 3864, 3865, 3871, 3872, 3873, 3875, 3876, 3882, 3883, 3884, 3886, 3887	\subparagraph	9, 20, 36
\pagenumbering	26	\subsectionname	15
\pagestyle	10	\subsectionpage	14, 100
\paragraph	9, 20, 36	\subst@correction	808, 815
\parindent	10, 31, 79, 81, 111	\tablename	12
\parskip	86	\textfont@name	891, 899
\part	14, 20, 22, 86	\tf@size	890
\partmark	86	\thechapter	13, 16, 24
\partname	15	\thesection	13, 16, 24, 99
\partpage	14, 100	\titleformat	95
\PassOptionsToClass	43	\titlespacing	95
\PassOptionsToPackage	72	\today	11, 11
\pdfmapline	117, 120	\trans@languagepath	3681, 3707
\pdfstringdef	73	\ttfamily	49
\pickup@font	51	\ttl@a	3123, 3131
		\ttl@chapterout	3118
		\ttl@extract	95
		\ttl@setifthe	3177
		\ttl@setsubmark	97, 3150, 3156
		\ttl@settopmark	97, 3144, 3175

- \ttl@tocpart 3120, 3122
- \ttlh@hang 3084, 3094
- \ttlh@runin 3083, 3093
- \uppercase 46
- \use@mathgroup 905, 936
- \usepackage 30, 31
- \verbatim@font 1635, 1639
- \wrong@fontshape 51
- \x@protect 5178
- \xdef 97
- \XeTeXglyphbounds 129
- \xkanjiskip 76
- \yahei 8
- \youyuan 8
- \z@ 710, 713, 2586, 2641,
2690, 2752, 2776, 2795, 2837, 2873, 2891, 2905, 3103,
3224, 3247, 3255, 3266, 3825, 3833, 3844, 3864, 3872, 3883
- \zhdig 82
- \zhdigits 26
- \zhnum 82
- \zhnumber 26, 26
- \zihao 25
- \ziju 25
- \zw 75
- tex commands:
 - \tex_def:D 5254
 - \tex_divide:D 80, 2184, 2225
 - \tex_edef:D 5308
 - \tex_endlinechar:D 544, 550
 - \tex_font:D 469, 728, 4885
 - \tex_iftrue:D 3633
 - \tex_ignorespaces:D
1062, 1221, 1226, 2077, 2209, 3724, 5067
 - \tex_immediate:D 466
 - \tex_pagegoal:D 2488
 - \tex_parskip:D 2496
 - \tex_prevdepth:D 2495
 - \tex_topskip:D 2489
- \the 4555, 4558, 4559, 4578, 4582
- \thechapter 2706, 2709, 2757, 3240, 3263, 3434
- \theparagraph 3537
- \thepart 2573, 2589, 2610, 2628, 3374
- \thesection 3224, 3231, 3247, 3255, 3266, 3454
- \thesubparagraph 3551
- \thesubsection 3227, 3258, 3486
- \thesubsubsection 3520
- \thispagestyle 2553, 2554, 2676, 2687, 2689
- \tiny 3853, 3892
- tl commands:
 - \c_space_tl 470
 - \tl_clear:N 194, 390,
1576, 1580, 1863, 1874, 2014, 2130, 2453, 2471, 3779, 5262
 - \tl_clear_new:N 58, 367
 - \tl_const:Nn
3, 410, 415, 420, 425, 517, 655, 677, 953, 968, 1264,
1265, 1266, 1267, 1326, 2149, 2372, 2378, 3615, 3634, 3636
 - \tl_gput_right:Nn 447, 5061, 5063, 5085
 - \tl_gset:Nn 561, 564, 890,
892, 894, 1300, 1735, 1763, 1781, 1806, 1813, 1814, 3970
 - \tl_gset_eq:NN 1044, 1758
 - \tl_gset_rescan:Nnn 1307
 - \tl_head:N 3783
 - \tl_if_blank:nTF 1393, 1489, 1490, 1594, 1595, 1609, 1627
 - \tl_if_blank_p:n 1359
 - \tl_if_empty:NTF 748, 2202, 2384, 3351, 3933
 - \tl_if_empty:nTF 1697, 1710, 5225
 - \tl_if_eq:NNTF 1731, 2142, 3694
 - \tl_if_exist:NTF 84, 516, 518, 519, 520, 1263,
1265, 1266, 1267, 2298, 2329, 2340, 2343, 3627, 3717, 5084
 - \tl_if_in:NnTF 5289
 - \tl_map_inline:Nn 1505
 - \tl_map_inline:nn 3780
 - \tl_new:N 128, 132, 518, 519, 520,
750, 751, 839, 950, 1003, 1029, 1050, 1051, 1072, 1441,
1582, 1751, 2028, 2150, 2163, 2300, 2370, 2371, 2987,
2988, 4123, 5074, 5075, 5084, 5145, 5146, 5265, 5266, 5267
 - \tl_put_left:Nn 1987, 5094
 - \tl_put_right:Nn
374, 896, 1589, 1622, 2400, 2442, 2444, 3783, 3784
 - \tl_replace_all:Nnn 1356, 1397
 - \tl_rescan:nn 136, 5307, 5316
 - \tl_set:Nn 114, 115, 129,
189, 222, 223, 465, 497, 498, 499, 557, 739, 741, 752,
791, 840, 947, 948, 989, 1055, 1067, 1209, 1270, 1355,
1394, 1396, 1437, 1610, 1614, 1652, 1653, 1654, 1655,
1677, 1753, 1983, 2013, 2029, 2091, 2100, 2126, 2164,
2212, 2301, 2451, 2454, 2458, 3080, 3099, 3118, 3123,
3131, 3693, 3911, 4124, 4128, 4171, 4172, 4173, 5139, 5140
 - \tl_set_eq:NN
757, 758, 805, 891, 893, 962, 1056, 1081, 1544, 1556, 2151
 - \tl_set_rescan:Nnn 5263
 - \tl_tail:N 3085
 - \tl_tail:n 3095
 - \tl_to_str:n 125, 126, 5206, 5235, 5276, 5277
 - \tl_trim_spaces:n 1573
- \today 2276, 2281, 2284, 2289
- today 11, 2276
- token commands:
 - \token_get_replacement_spec:N 5163, 5185
 - \token_to_meaning:N 5220
 - \token_to_str:N
54, 872, 881, 890, 892, 894, 1109, 1110, 1111, 1113,
1120, 1121, 1122, 1184, 2326, 2327, 3732, 4892, 5052, 5124
- \toks 4555, 4556, 4558, 4559, 4578, 4582
- \topsep 3836, 3847, 3875, 3886
- \TrimSpaces 2108
- \ttdefault 1285, 1741
- \ttfamily 529, 532, 1664, 1667
- two commands:
 - \c_two_hundred_fifty_six 560, 563
- \twocolumn 2681
- \typeout 2706, 2707, 3118

U

um commands:

\um_input_math_symbol_table: 929, 933
 \um_sym:nnn 932

um internal commands:

_um_input_math_symbol_table: 941
 _um_switchto_literal: 938
 _um_sym:nnn 940
 \undefined 4580
 \unless 2838, 2874, 2892
 \unskip 2908
 \updefault ... 958, 964, 966, 970, 972, 1328, 1330, 1333, 1337
 \upshape 1687, 1690

use commands:

\use:N 759, 797, 798, 848, 890, 892, 894,
 973, 1021, 1557, 2512, 2518, 2527, 2959, 5290, 5310, 5324
 \use:n 64, 122, 133, 389, 736, 847, 955, 1139,
 1192, 1198, 1297, 1301, 1399, 1573, 1752, 1905, 2478,
 2923, 2934, 3201, 3625, 3742, 3942, 5159, 5201, 5252, 5273
 \use:nn 5318
 \use_i:nn 108, 869, 2506, 5105, 5110, 5130, 5135
 \use_i:nnn 262
 \use_ii:nn ... 109, 869, 2504, 2508, 5105, 5110, 5130, 5135
 \use_ii:nnn 134, 267
 \use_iii:nnn 272
 \use_none:n 190, 195, 200, 1038, 1039, 1300, 1434, 1906, 2262
 \use_none:nn 482
 \usebeamercolor 3277, 3296, 3315, 3417, 3477, 3511
 \usebeamerfont 3276, 3286,
 3295, 3305, 3314, 3324, 3416, 3419, 3476, 3479, 3510, 3513
 \uselanguage 3672, 3677, 3698, 3703
 \usepackage 5052
 UTF8 6, 222

V

\verse 3921
 \vfil 2561, 2668

\vskip 2598, 2601,
 2628, 2648, 2651, 2671, 2757, 2766, 2769, 2784, 2787,
 2894, 2897, 3278, 3297, 3316, 3400, 3418, 3437, 3478, 3512
 \vspace 2564, 2748, 2751, 2772, 2775

W

winfonts 30
 \write 4595

X

\x 4557, 4560,
 4561, 4562, 4563, 4564, 4565, 4566, 4567, 4568, 4569,
 4570, 4571, 4572, 4573, 4574, 4575, 4576, 4583, 4593, 4597
 \xdef 4596
 \xeCJKsetup 661, 1870, 2072, 2075, 2078, 2093
 xetex commands:
 \xetex_charglyph:D 4889
 \xetex_glyphbounds:D 4913

Y

\yahei 4545, 4546
 \youyuan 4543

Z

\zhdig 2268
 \zhdigits 26, 2275
 zhmap 7, 30, 257
 \zhnum 2267
 zhnum commands:
 \zhnum_counter:n 2259
 \zhnumber 26, 2273
 \zhnumsetup 2285, 2290
 \zhtoday 2284, 2289
 \zihao 25, 3723, 3732
 zihao 8, 159
 \ziju 25, 2208
 \zw 711, 1931, 1971, 2030