

CT_EX 宏集手册

CTEX.ORG

2019/05/29 v2.4.16*

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. 735dfe2`.

第 1 节 介绍

历史

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

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

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

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

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

最初，Knuth 在设计开发 T_EX 的时候没有考虑到多国语言支持，特别是对多字节的中日韩语言的支持。这使得 T_EX 以至后来的 L^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“CTeX”。

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-ja 宏包

第 2 节 简明教程

2.1 C_TE_X 宏集组成

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

表 1 C_TE_X 宏集的组成

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

2.2 C_TE_X 宏集的安装和更新

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

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

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

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

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

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

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

⁴Windows 系统的命令行是 CMD 命令提示符,你可以使用 Win + R 组合键打开“运行”对话框,然后输入 cmd 确认打开命令提示符窗口。
⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

2.3 使用 CT_EX 文档类

如果用户需要在标准文档类的基础上添加中文支持和中文版式支持, 我们建议用户使用 CT_EX 宏集提供的四个中文文档类。

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

例 1

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

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

以下是使用 `ctexbeamer` 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

New: 2014-03-18

`\ctexset` $\{ \langle \text{键值列表} \rangle \}$

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-ja	原生

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

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

不同的编译方式和中文支持方式会在一定程度上影响 $\text{CT}_{\text{E}}\text{X}$ 宏集的行为, 比如宏包对编码的处理。在用户使用 $\text{Xe}_{\text{L}}\text{La}_{\text{T}}\text{E}_{\text{X}}$ 、 $\text{Lua}_{\text{La}}\text{TeX}$ 及 $\text{up}_{\text{La}}\text{TeX}$ 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 $\text{La}_{\text{T}}\text{E}_{\text{X}}$ 及 $\text{pdf}_{\text{La}}\text{TeX}$ 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 $\text{Xe}_{\text{L}}\text{La}_{\text{T}}\text{E}_{\text{X}}$ 、 $\text{Lua}_{\text{La}}\text{TeX}$ 或 $\text{up}_{\text{La}}\text{TeX}$ 编译。

使用 $\text{Xe}_{\text{L}}\text{La}_{\text{T}}\text{E}_{\text{X}}$ 、 $\text{Lua}_{\text{La}}\text{TeX}$ 或 $\text{up}_{\text{La}}\text{TeX}$ 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 $(\text{pdf})_{\text{La}}\text{TeX}$ 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 $\text{La}_{\text{T}}\text{E}_{\text{X}}$ 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
$\text{Xe}_{\text{L}}\text{La}_{\text{T}}\text{E}_{\text{X}}$	xeCJK 华文字库	xeCJK 中易字库 + 微软雅黑	xeCJK 中易字库	xeCJK Fandol 字库 ³
$\text{Lua}_{\text{La}}\text{TeX}$ 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
$\text{pdf}_{\text{La}}\text{TeX}$	不可用 ⁵	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	不可用 ⁵
$\text{La}_{\text{T}}\text{E}_{\text{X}}$ + 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
$\text{up}_{\text{La}}\text{TeX}$ + 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 $\text{Lua}_{\text{La}}\text{TeX}$ 编译时使用 LuaTeX-ja 宏包。对此, 第 9 节有特别说明。

5 受 $\text{pdf}_{\text{La}}\text{TeX}$ 的限制, 无法嵌入 OpenType 字体。

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

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

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

zhmap	☆	zhmap = <code><true false zhmCJK></code>
Updated: 2014-03-08		
指定字体映射机制。本选项只在使用 pdf \LaTeX / \LaTeX 编译时有意义。		
true		使用 zhmetrics 宏包, 将 CJK 字库通过 <code>\special</code> 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 =
New: 2014-03-08	
<code><none adobe fandol founder mac macnew macold ubuntu windows windowsnew windowsold ...></code>	
指定 CT \LaTeX 宏集加载的字库。	

如果没有指定 fontset 的值, CT \LaTeX 宏集将自动检测用户使用的操作系统, 配置相应的字体 (参见表 3)。

CT \LaTeX 预定义了以下六种中文字库。

adobe	使用 Adobe 公司的四款中文字体, 不支持 pdf \LaTeX 。
fandol	使用 Fandol 中文字体, 不支持 pdf \LaTeX 。
founder	使用方正公司的中文字体。
mac	使用 Mac OS X 系统下的字体, 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 CT \LaTeX 预定义的中文字库, 可以设置 fontset 为下述值之一。

none	不配置中文字体, 需要用户自己配置。
<name>	这里 <name> 为自定义的名字。CT \LaTeX 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 CT \LaTeX 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

\heiti

\fangsong

\kaishu

宋体,CJK 等价命令 \CJKfamily{zhsong}。
黑体,CJK 等价命令 \CJKfamily{zhhei}。
仿宋,CJK 等价命令 \CJKfamily{zhfs}。
楷书,CJK 等价命令 \CJKfamily{zhkai}。

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

\lishu

\youyuan

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

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{zhyahei}。

在 macnew 字库中,还定义了 \pingfang。

\pingfang

苹方黑体,CJK 等价命令 \CJKfamily{zhpfb}。

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
\tiny	七号	5.5	小六	6.5	5	6	6
\scriptsize	小六	6.5	六号	7.5	7	8	8
\footnotesize	六号	7.5	小五	9	8	9	10
\small	小五	9	五号	10.5	9	10	11
\normalsize	五号	10.5	小四	12	10	11	12
\large	小四	12	小三	15	12	12	14
\Large	小三	15	小二	18	14	14	17
\LARGE	小二	18	二号	22	17	17	20
\huge	二号	22	小一	24	20	20	25
\Huge	一号	26	一号	26	25	25	25

10pt ☆
11pt ☆
12pt ☆

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 ☆
New: 2014-03-08

heading = <true|false>

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 ☆
New: 2015-04-15

scheme = <chinese|plain>

选择文章的排版方案,预设有 chinese 和 plain 两种方案。

chinese 对 beamer 以外的文档类,调整默认字号为五号字,并调整行距为 1.3;汉化文档中的标题名字(如“图”、“表”、“目录”和“参考文献”等,见 6.2 节);在 heading = true 的情况下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 indentfirst 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,也可能会影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中,那么该选项将会被自动激活。

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

Updated: 2014-04-11 设置标点处理格式。预定义好的格式有:

`quanjiao` 全角式: 所有标点占一个汉字宽度, 相邻两个标点占 1.5 汉字宽度;
`banjiao` 半角式: 所有标点占半个汉字宽度;
`kaiming` 开明式: 句末点号¹⁴用占一个汉字宽度, 标号和句内点号占半个汉字宽度;
`CCT` CCT 式: 所有标点符号的宽度略小于一个汉字宽度;
`plain` 原样(不调整标点间距)。

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

Updated: 2014-03-08 是否在生成的 PDF 中保留汉字后面的空格。

`true` 总是保留汉字后的空格。此时, 用户需要自行在行尾加上 `%` 处理换行产生的空格¹⁵。
`false` 总是忽略掉汉字后面的空格, 不论汉字后是什么(使用 $(\text{pdf})\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 = <数值>`

New: 2014-04-23

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

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

`scheme = plain` $\text{CT}_{\text{E}}\text{X}$ 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 = <true|false|数值|带单位的数值>`

New: 2014-03-13

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

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

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化, 使之以中文的方式显示今天的日期。如编译本文档的日期就是“2019 年 5 月 29 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“2019 年 5 月 29 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇一九年五月二十九日”。使用全汉字的日期格式。

`old` 效果为“May 29, 2019”。使用文档原来的(英文)日期格式。

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

例 9

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

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

6.2 文档标题汉化

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

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

<hr/> <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/> <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/> <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/> <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/> <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/> <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/> <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/> <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/> <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/> <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/> <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/> <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/> <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 `heading` 选项时,会设置整个文档的页面格式(`page style`)为 `headings`,即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

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

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

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

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

有关 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`, `before skip`, `after skip`, `fix skip`, `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`, `fix skip`, `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 L^AT_EX 辅助文件中的是 `number` 选项的定义。

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

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 C_T_EX 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 = {<代码>}

...

aftername+

*

aftername+= {<代码>}

Updated: 2014-03-08

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

aftername+ 选项用于在原有代码后面附加代码。

表 9 aftername 选项的默认设置

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

表 12 runin 选项的默认设置

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

```
section/runin      * runin = true|false
```

```
subsection/runin  *
```

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

```
paragraph/runin   *
```

```
subparagraph/runin *
```

New: 2015-06-27

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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

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

New: 2015-06-27

afterindent 选项用于设置章节标题后首段的缩进。

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

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

表 13 afterindent 选项的默认设置

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

```
.../beforeskip  * beforeskip = {<弹性间距>}
```

Updated: 2016-05-10

beforeskip 选项用于设置章节标题前的垂直间距。

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

表 14 before skip 选项的默认设置

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

表 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

.../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。

.../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。

如果 indent 的值是以 em、ex 或 \ccwd 为单位, 那么缩进间距的大小是相对于 format 中指定的字号大小。

例如, 设置 \part 标题缩进三个字、\section 标题缩进 20pt:

例 19

```
\ctexset{
  part={
    format+=\raggedright,
```

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

表 16 indent 选项的默认设置

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

表 17 hang 选项的默认设置

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

part/hang	*	hang = true false
chapter/hang	*	
section/hang	*	hang 选项用于设置是否对章节标题实施悬挂缩进(缩进的宽度为名字宽度和 indent 选项设置的宽度之和)。
subsection/hang	*	
subsubsection/hang	*	
paragraph/hang	*	本选项对 beamer/ctexbeamer 文档类无效。对于 \section 级以下标题,若设置了 runin 选项为 true,即标题与随后正文排在同一段,hang 选项没有意义。
subparagraph/hang	*	
Updated: 2019-03-31		

part/pagestyle	*	pagestyle = {\页面格式}
chapter/pagestyle	*	
设置 book/ctexbook 或 report/ctexrep 文档类中, \part 与 \chapter 标题所在页的页面格式(page style)。		
New: 2014-03-21		

表 18 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 19 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 20 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 21 appendix/name 选项的默认设置

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

appendix/number * number = {(数字输出命令)}

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

用法与普通章节的 **number** 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

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

表 22 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 23 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 23 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

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

 $\backslash\text{chinese}$ 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 \LaTeX 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 $\text{CT}\text{\TeX}$ 标志。

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

在 $\text{Lua}\text{\LaTeX}$ 下， $\text{CT}\text{\TeX}$ 宏集依赖 $\text{Lua}\text{\TeX-j}\text{\LaTeX}$ 宏包来完成中文支持。该宏包是日本 \TeX 社区的北川弘典、前田一贵、八登崇之等人开发的，设计目的主要是在 $\text{Lua}\text{\TeX}$ 引擎下实现日本 $\text{p}\text{\TeX}$ 引擎的（大部分）功能。它为了兼容 $\text{p}\text{\LaTeX}$ 的使用习惯，对 $\text{\LaTeX}2_{\epsilon}$ 的 NFSS 作了不少修改和扩充。这对于简体中文用户来说不是必要的，因而 $\text{CT}\text{\TeX}$ 禁用了它在 \LaTeX 格式下的大部分设置，只保留了必要的部分。同时修改了它的字体设置方式，使得相关命令与 xeCJK 宏包大致相同。

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

9.1 *Lua*TeX 下替代字体的设置

AlternateFont	\setCJKfamilyfont {<family>}
New: 2014-04-14	<pre>[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	\setCJKfamilyfont {<family>}
New: 2014-04-14	<pre>[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 *	\ctexset
New: 2014-04-14	<pre>{ 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 $\langle family \rangle$ 的替换字体设置。如果没有给定值,则作用于当前 CJK 字体族。清除与重置操作总是全局的。

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

例如对于 T_EX Live,系统自带的配置文件就在 T_EX Live 安装目录下的 texmf-dist/tex/latex/ctex/config/ 子目录下,可以修改它的副本,保存在本地 TDS 树的 texmf-local/tex/latex/ctex/ 目录下,或者用户 TDS 树的 ~/.texlive2015/texmf-var/tex/latex/ctex/ 目录下,作为本地/用户专有的配置文件。复制配置文件后需要运行 texhash 命令使本地配置文件生效。

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

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

10.1 修改宏包默认选项

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

例 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-jaf 的 jcharwidowpenalty 参数。
\sys_if_engine luatex:T
{
  \ltjsetParameter { jcharwidowpenalty = 10000 }
}
```

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

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

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

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

hyperref

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

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

fntef

旧版本的 `fntef` 选项用于统一 `CCTfntef` 与 `CJKfntef` 的界面, 新版本 `CTEX` 宏集不再支持 `CCT`, 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。

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

在调用 `fntef` 宏包选项的同时, 旧版本 `CTEX` 宏包由于需要支持 `CCT` 系统, 会将以 `\CJK` 开头的 `\CJKunderline` 等宏换名为以 `\CTEX` 开头的 `\CTEXunderline` 等宏。此功能在新版本的 `CTEX` 宏集中已失去意义。

此外, 在 `pdfTEX` 引擎下, 用于设置格式的 `\CJKunderdotbasesep` 等宏也被换名为 `\CTEXunderdotbasesep` 等宏。

在新版本中, 上述由 `fntef` 衍生的相关宏都成为过时命令。

\CTEXsetfont

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

\CTEXindent

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

\CTEXnoindent

设置 `\parindent=0pt`。过时命令。

\CTEXsetup

`\CTEXsetup[⟨选项⟩]{⟨标题⟩}`

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

\CTEXoptions

`\CTEXoptions[⟨选项⟩]`

相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

\Chinese

`\Chinese{⟨counter⟩}`

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

captiondelimter

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

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

例 26

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

11.3 CT_EX 1.02c 以后的 SVN 开发版

`CTEX` 宏包在 1.02c 版本 (2011/03/11) 之后在 Google code 上的 SVN 开发版本, 内部版本号一直升到 1.11 版, 但从未正式发布。SVN 开发版在 1.02c 版本的基础上新增的功能在第 2 版中大多继承了过来, 但新增的命令与选项都不再保持兼容。

`CTEX` 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 CT_EX 2.2 之前的版本

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

在 CT_EX 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 CT_EX 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 CT_EX 2.4.1 和 2.4.2

这两个选项已经被删除, 相应功能由新的选项 fixskip 提供。

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

part/fixbeforeskip
chapter/fixbeforeskip

第 12 节 宏集依赖情况与手工安装方法

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

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

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 宏集。
- indentfirst 宏包, 属于 tools 宏集。
- everyisel 宏包, 属于 ms 宏集。
- ulem 宏包。
- zhnumber 宏包。

➡ 以上是各种编译方式都必需的依赖项。

- CJK 宏集。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包, 它还依赖 oberdiek 宏集。

➡ 以上是使用 pdf_LT_EX 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项, 其中 zhmcjk 是可选的。

- xeCJK 宏集, 它还依赖
 - xtemplate 宏包, 它属于 l3packages 宏集。
 - fontspec 宏包。

- 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 \LaTeX 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 \LaTeX 编译时的依赖项。

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

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

\CTEX 宏集已被 \TeX Live 和 MiK \TeX 收录, 若无特别理由, 我们强烈建议用户使用宏包管理器安装本宏集。

若要手工安装, 请遵循如下步骤:

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

第 13 节 开发人员

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

目前比较活跃的开发维护人员是刘海洋、李清和黄晨成。

参考文献

- [1] DONALD ERVIN KNUTH. *The 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 \@ifpackageloaded { ctexsize }

19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }

20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }

21 \@ifpackageloaded { ctexheading }

22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }

23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }

24 </ctex>

25 </class|ctex>

26 <*class|style>

27 \RequirePackage { xparse , l3keys2e }

28 </class|style>

29 <*class|ctex>

检查 expl3 和 l3keys2e 的版本。

30 \msg_new:nnnn { ctex } { l3-too-old }

31 { Support~package~`#1'~too~old. }

32 {

33 Please~update~an~up~to~date~version~of~the~bundles\\\\

34 `l3kernel'~and~`l3packages'\\\\

35 using~your~TeX~package~manager~or~from~CTAN.

36 }

37 \@ifpackagelater { expl3 } { 2019/03/05 } { }

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,~lua_latex,~pdf_latex,~up_latex,~or~ap_latex. }

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:nxx { ctex } { engine-not-supported } { \c__ctex_engine_str } }

54 </class|ctex>

55 <*class|ctex|ctexheading>

```

56 \RequirePackage { ctexhook , ctexpatch }
<!\ctexheading> 57 \RequirePackage { fix-cm , everyrel }

```

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

```

94 <!\beamer>
95 \int_new:N \g__ctex_section_depth_int
96 \int_gset:Nn \g__ctex_section_depth_int { 2 }
97 <!\beamer>

98 </class|ctex|ctexheading>
99 <*\class|ctex>

```

对旧版本的宏包给出错误信息。

```
100 \msg_new:nnnn { ctex } { package-too-old }
101 { Support~package~`#1'~too~old. }
102 {
103   Please~update~an~up~to~date~version~of~the~package~`#1'\
104   using~your~TeX~package~manager~or~from~CTAN.
105 }
```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```
106 \sys_if_output_pdf:TF
107 { \cs_new_eq:NN \ifctexpdf \if_true: }
108 { \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。

```
109 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
110 \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。

```
111 \cs_new_protected:Npn \ctex_set_default_ccwd:Nn #1#2
112 { \tl_set:Nx #1 { \__ctex_default_ccwd_aux:n {#2} } }
113 \cs_new:Npn \__ctex_default_ccwd_aux:n #1
114 {
115   \exp_not:n {#1}
116   \exp_after:wN \__ctex_default_ccwd_aux:w
117   \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop
118 }
119 \exp_last_unbraced:NNNN
120 \cs_new:Npn \__ctex_default_ccwd_aux:w #1 { \tl_to_str:n { pt } } #2 \q_stop
121 { \tl_if_empty:nT {#2} { \ccwd } }
```

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

```
122 \tl_new:N \l__ctex_encoding_tl
123 \tl_set:Nx \l__ctex_encoding_tl
124 { \sys_if_engine_pdftex:TF { GBK } { UTF8 } }
```

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

```
125 \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。

```
126 \tl_new:N \l__ctex_autoindent_tl
```

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

```
127 \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`。

```
128 \cs_new_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
```

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

```
<class> 129 \cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
<ctex> 130 \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。

```
131 <*class>
132 \clist_new:N \g__ctex_std_options_clist
133 </class>
```

对无效选项给出警告。

```
134 \msg_new:nnn { ctex } { invalid-option }
135 { Option~`\l_keys_key_tl'~is~invalid~in~current~mode. }
136 \msg_new:nnn { ctex } { invalid-value }
137 { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_tl'. }
```

对过时选项或命令给出警告。

```
138 \msg_new:nnn { ctex } { deprecated-option }
139 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
140 \msg_new:nnn { ctex } { deprecated-command }
141 { Command~ #1 is~ deprecated.\\ #2 }
142 \msg_new:nnn { ctex } { deprecated-environment }
143 { Environment~`#1'~is~deprecated.\\ #2 }
144 </class|ctex>
```

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

```
145 <*class|ctex|ctexsize>
146 \int_new:N \g__ctex_font_size_int
147 \int_set:Nn \g__ctex_font_size_int { -1 }
148 </class|ctex|ctexsize>
```

14.2 宏包选项

```
149 <*class|style>
150 \keys_define:nn { ctex / option }
151 {
152 </class|style>

zihao 153 <*class|ctex|ctexsize>
154     zihao .choice: ,
155     zihao .value_required:n = true ,
156     zihao /      5 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 0 } } ,
157     zihao /     -4 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 1 } } ,
158     zihao / false .code:n = { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
<ctexsize> 159 }
160 </class|ctex|ctexsize>
161 <*class|ctex>
162     c5size .code:n =
163     {
164         \msg_warning:nnn { ctex } { deprecated-option }
165         { Option~`zihao=5'~is~ set. }
166         \keys_set:nn { ctex / option } { zihao = 5 }
167     } ,
168     cs4size .code:n =
169     {
170         \msg_warning:nnn { ctex } { deprecated-option }
171         { Option~`zihao=-4'~is~ set. }
172         \keys_set:nn { ctex / option } { zihao = -4 }
173     } ,
174     c5size .value_forbidden:n = true ,
175     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。

```
176     linespread .fp_set:N = \l__ctex_line_spread_fp ,
```

```

177 linespread .initial:n = { \c_nan_fp } ,
178 linespread .value_required:n = true ,

```

autoindent 自动调整段落的首行缩进功能。

```

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

```

indent 仅为兼容性保留,已过时。

```

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

```

```

GBK 216 GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
UTF8 217 UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
218 GBK .value_forbidden:n = true ,
219 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。

```

220 fontset .tl_gset:N = \g__ctex_fontset_tl ,
221 nofonts .code:n =
222 {
223   \msg_warning:nnn { ctex } { deprecated-option }
224   {
225     Option~ `fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
226     fontset~ option.~
227   }
228   \keys_set:nn { ctex / option } { fontset = none }
229 } ,

```

```

230 adobefonts .code:n =
231 {
232   \msg_warning:nnn { ctex } { deprecated-option }
233   {
234     Option~`fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
235     fontset~ option.
236   }
237   \keys_set:nn { ctex / option } { fontset = adobe }
238 },
239 winfonts .code:n =
240 {
241   \msg_warning:nnn { ctex } { deprecated-option }
242   {
243     Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
244     fontset~ option.
245   }
246   \keys_set:nn { ctex / option } { fontset = windows }
247 },
248 nofonts .value_forbidden:n = true ,
249 winfonts .value_forbidden:n = true ,
250 adobefonts .value_forbidden:n = true ,

```

zhmap

```

251 zhmap .choice: ,
252 zhmap .default:n = { true } ,
253 zhmap / zhmCJK .code:n =
254 {
255   \bool_gset_true:N \g__ctex_zhmCJK_bool
256   \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
257 },
258 zhmap / true .code:n =
259 {
260   \bool_gset_false:N \g__ctex_zhmCJK_bool
261   \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
262 },
263 zhmap / false .code:n =
264 {
265   \bool_gset_false:N \g__ctex_zhmCJK_bool
266   \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
267 },
268 nozhmap .code:n =
269 {
270   \msg_warning:nnn { ctex } { deprecated-option }
271   { Option~`zhmap=false'~ is~ set. }
272   \keys_set:nn { ctex / option } { zhmap = false }
273 },
274 nozhmap .value_forbidden:n = true ,

```

punct 设置标点符号输出格式。

```

275 punct .tl_set:N = \l__ctex_punct_tl ,
276 punct .default:n = { quanjiao } ,
277 punct .initial:n = { quanjiao } ,
278 nopunct .code:n =
279 {
280   \msg_warning:nnn { ctex } { deprecated-option }
281   { Option~`punct=plain'~ is~ set. }
282   \keys_set:nn { ctex / option } { punct = plain }
283 },
284 nopunct .value_forbidden:n = true ,

```

space

```

285 space .choices:nn =
286 { true , auto , false }
287 {
288   \exp_args:Nx \ctex_at_end:n

```



```

289     { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
290   } ,
291   space .default:n = { true } ,
292   nospace .code:n =
293   {
294     \msg_warning:nnn { ctex } { deprecated-option }
295     { Option~ `space=false'~ is~ set. }
296     \keys_set:nn { ctex / option } { space = false }
297   } ,
298   nospace .value_forbidden:n = true ,

heading 299   heading .bool_set:N = \l__ctex_heading_bool ,

300 </class|ctex>
301 <*class|ctex|ctexheading>

sub3section 302 <!*beamer>
sub4section 303   sub3section .code:n =
304     { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
305   sub4section .code:n =
306     { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
307   sub3section .value_forbidden:n = true ,
308   sub4section .value_forbidden:n = true ,
309 <!/beamer>

scheme 310   scheme .tl_set:N = \l__ctex_scheme_tl ,
311 <*ctexheading>
312   scheme .default:n = { plain } ,
313   scheme .initial:n = { plain }
314 }
315 </ctexheading>
316 <!*ctexheading>
317   scheme .default:n = { chinese } ,
318   scheme .initial:n = { chinese } ,
319 </!ctexheading>

320 </class|ctex|ctexheading>
321 <*class|ctex>

322   cap .code:n =
323   {
324     \msg_warning:nnn { ctex } { deprecated-option }
325     { Option~ `scheme = chinese'~ is~ set. }
326     \keys_set:nn { ctex / option } { scheme = chinese }
327   } ,
328   nocap .code:n =
329   {
330     \msg_warning:nnn { ctex } { deprecated-option }
331     { Option~ `scheme = plain'~ is~ set. }
332     \keys_set:nn { ctex / option } { scheme = plain }
333   } ,
334   cap .value_forbidden:n = true ,
335   nocap .value_forbidden:n = true ,

fntef 这些过时的宏包兼容选项，原选项功能总是打开的。
fancyhdr
hyperref 336   fntef .code:n =
337   {
338     \msg_warning:nnn { ctex } { deprecated-option }
339     { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
340   } ,
341   fancyhdr .code:n =
342   {
343     \msg_warning:nnn { ctex } { deprecated-option }
344     { `fancyhdr'~ package~ is~ loaded. }

```

```

345     \RequirePackage { fancyhdr }
346   } ,
347   hyperref .code:n =
348   {
349     \msg_warning:nnn { ctex } { deprecated-option }
350     { `hyperref'~ package~ will~ be~ loaded. }
351     \ctex_at_end:n
352     {
353       \cs_if_exist:NF \hypersetup
354       { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
355     }
356     \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
357   } ,
358 }

359 </class>ctex>
360 <*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

```

361 \tl_clear_new:N \l__ctex_tmp_tl
362 \clist_map_inline:nn
363 {
364   10pt , 11pt , 12pt ,
365   8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
366 }
367 {
368   \tl_put_right:Nn \l__ctex_tmp_tl
369   {
370     #1 .code:n =
371     <*class>
372     { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
373     </lclass>
374     <*class>
375     {
376       \int_gset:Nn \g__ctex_font_size_int { 2 }
377       \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
378     } ,
379     </class>
380     #1 .value_forbidden:n = true ,
381   }
382 }
383 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
384 \tl_clear:N \l__ctex_tmp_tl

```

将未知选项传给标准文档类。

```

385 <*class>
386 \keys_define:nn { ctex / option }
387 {
388   unknown .code:n =
389   { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
390 }
391 </class>

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

394 <*class>style>
395 \ProcessKeysOptions { ctex / option }
396 </class>style>

397 <*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。

```

398 \if_case:w \g__ctex_font_size_int

```

```

399 \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
400 \or:
401 \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
402 \fi:

```

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

```

403 <*article>
404 \tl_const:Nn \c__ctex_class_tl { article }
405 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
406 \LoadClass { article }
407 </article>
408 <*book>
409 \tl_const:Nn \c__ctex_class_tl { book }
410 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
411 \LoadClass { book }
412 </book>
413 <*report>
414 \tl_const:Nn \c__ctex_class_tl { report }
415 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
416 \LoadClass { report }
417 </report>
418 <*beamer>
419 \tl_const:Nn \c__ctex_class_tl { beamer }
420 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
421 \LoadClass { beamer }
422 </beamer>
423 </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。

```

424 <*pdfTeX|uptex|aptex>
425 \cs_new_protected_nopar:Npn \ctex_set_zhmap:n #1
426 {
427   \AtBeginDvi {#1}
428   \ctex_at_end_package:nn { atbegshi }
429   { \AtBeginShipoutFirst {#1} }
430 }
431 \@onlypreamble \ctex_set_zhmap:n
432 </pdfTeX|uptex|aptex>
433 <*pdfTeX>

```

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

```

434 \seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
435 { C19 , C10 , C00 , C09 , C40 , C60 }

```

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

```

436 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2
437 {
438   \cs_if_free:cF { #1 + #2 }
439   {
440     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
441     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
442   }
443 }
444 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
445 \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。

```

446 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
447 {
448   \cs_if_free:NF \CJK@plane
449   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
450 }
451 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
452 {
453   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
454   #1
455 }
456 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
457 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
458 {
459   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
460   \tex_immediate:D \tex_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
461   \cs_new_protected_nopar:Npx #1
462   {
463     \exp_not:N \tex_pdffontattr:D \exp_not:N \tex_font:D
464     { /ToUnicode ~ \int_use:N \tex_pdflastobj:D \c_space_tl 0 ~ R }
465   }
466 }

```

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

```

467 \group_begin:
468 \char_set_catcode_other:N \#
469 \sys_if_output_pdf:TF
470 {
471   \group_end:
472   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
473   { \CTEX@Family@CMap {#1} {#2} }
474   {
475     \ctex_at_end_package:nn { cmap }
476     { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
477   }
478   { \ctex_patch_failure:N \DeclareFontFamily }
479 }
480 { \group_end: }

```

首先检查选项,决定是否载入 `zhmCJK` 宏包。

```

481 \if_bool:N \g__ctex_zhmCJK_bool
482 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
483 \RequirePackage { zhmCJK }

```

不载入 `zhmCJK` 宏包时直接调用 `CJK` 及相关宏包。

```

484 \else:
485   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
486   { \RequirePackage { CJK } }
487   { \RequirePackage { CJKutf8 } }
488   \RequirePackage { CJKpunct , CJKspace }

```

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

```

489 \cs_new_protected_nopar:Npn \ctex_load_zhmap:n #1#2#3#4
490 {
491   \tl_set:Nn \CJKrmdefault {#1}
492   \tl_set:Nn \CJKsfdefault {#2}
493   \tl_set:Nn \CJKttdefault {#3}
494   \ctex_set_zhmap:n { \ctex_zhmap_input:n {#4} }
495 }
496 \@onlypreamble \ctex_load_zhmap:n

```

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

```

497 \cs_new_protected_nopar:Npn \ctex_zhmap_input:n #1
498 {
499   \ctex_file_wrapper:nnn
500   {
501     \char_set_catcode_comment:n { 37 } % %
502     \char_set_catcode_escape:n { 92 } % \
503   }
504   { \file_input:n {#1} }
505   {
506     \char_set_catcode:nn { 37 } { \char_value_catcode:n { 37 } }
507     \char_set_catcode:nn { 92 } { \char_value_catcode:n { 92 } }
508   }
509 }

510 \tl_if_exist:NF \CJKfamilydefault
511 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
512 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
513 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
514 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
515 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
516 { \CJKfamily { \CJKrmdefault } }
517 { }
518 { \ctex_patch_failure:N \rmfamily }
519 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
520 { \CJKfamily { \CJKsfdefault } }
521 { }
522 { \ctex_patch_failure:N \sffamily }
523 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
524 { \CJKfamily { \CJKttdefault } }
525 { }
526 { \ctex_patch_failure:N \ttfamily }
527 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
528 { \CJKfamily { \CJKfamilydefault } }
529 { \cs_set_eq:NN \reset@font \normalfont }
530 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

531 `\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。

```

532 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
533 {
534   \ctex_file_wrapper:nnn
535   {
536     \char_set_catcode_other:n { 60 } % <
537     \char_set_catcode_math_superscript:n { 94 } % ^
538     \int_set:Nn \tex_endlinechar:D { -1 }
539   }
540   { \file_input:n {#1} }
541   {
542     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
543     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
544     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
545   }
546 }
547 \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

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

\uppercase 的实现。

```

548 \if_cs_exist:N \CJK@surr
549 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
550 {
551   \tl_set:Nx \l__ctex_tmp_tl {#2}
552   \int_set:Nn \l__ctex_tmp_int
553     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
554   \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
555     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
556     {
557       \int_sub:Nn \l__ctex_tmp_int { 256 }
558       \tl_gset:Nx #1
559         {
560           \int_to_Hex:n
561             { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
562           \int_to_Hex:n
563             { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
564         }
565     }
566   }
567 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
568 \fi:

```

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

```

569 \AtBeginDocument
570 {
571   \str_if_eq:eeF { \l__ctex_punct_tl } { quanjiao }
572   { \punctstyle { \l__ctex_punct_tl } }
573 }

```

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

```

574 \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。

```

575 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
576 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
577 {
578   \CJK@makeActive
579   \CJK@loadBinding { standard }
580 }

```

在导言区结束时调用 \CJK@envStart 启用完整的中文功能。

\CJK@envStart 的定义是

```

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

```

```

\CJKfontenc{#2}{#1}
\CJKfamily{#3}
\def\CJK@series{\f@series}
\def\CJK@shape{\f@shape}%
\csname CJKhook\endcsname}

```

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

```

581 \exp_args:Nx \ctex_at_end_preamble:n
582 {
583   \exp_not:N \CJK@envStart
584   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
585   \exp_not:N \CJKtilde
586 }

```

zhmCJK 判断结束。

```
587 \fi:
```

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

```

588 <@@=>
589 \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。

```
590 <@@=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 \CJKhook 备用。

```

591 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
592 {
593   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
594     { \cs_set_eq:NN \CJK@ignorespaces #1 }
595   \ctex_set_ignorespaces:
596 }
597 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
598 { \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。

```
599 \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。

```

600 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
601 {
602   \clist_map_inline:Nn \c__ctex_punct_family_clist
603   {
604     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
605     {
606       \cs_set_eq:cc
607       { CJKpunct@ ##1 @spaces }
608       { c__ctex_ #1 ##1 _punct_spaces_tl }
609     }
610   }
611 }
612 \clist_const:Nn \c__ctex_punct_family_clist
613 {
614   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
615   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
616 }

```

```

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

617 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
618 {
619   \cs_if_free:cF { CJKpunct@ #2 @spaces }
620   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
621 }

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

622 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
623 {
624   \clist_map_inline:nn {#1}
625   {
626     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
627     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
628   }
629 }
630 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
631 {
632   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
633   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
634   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
635   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
636   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
637   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
638 }

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

639 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
640 {
641   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
642   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
643   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
644   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
645   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
646   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
647 }

\ctex_punct_space:nn
\ctexspadef 定义标点的边界信息。

648 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
649 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
650 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

651 \ctex_file_input:n { ctexspa.def }
652 </pdfTeX>

```

14.3.2 ctex-engine-xetex.def

```

653 <*xetex>
654 \RequirePackage { xeCJK }
655 \exp_args:Nx \xeCJKsetup
656 {
657   LoadFandol = false ,
658   PunctStyle = \l__ctex_punct_tl
659 }

```

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**。

```

660 \@ifpackagelater { fontspec } { 2014/05/25 } { }

```



```

661 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
662 </xetex>

```

14.3.3 ctex-engine-luatex.def

```

663 <*luatex>

```

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

```

664 \msg_new:nnn { ctex } { luatexja-loaded }
665 {
666   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
667   Loading~file~`#1'~will~abort!
668 }
669 \ifpackageloaded { luatexja }
670 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
671 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
672 \RequirePackage { luatexja }
673 \ifpackagelater { luatexja } { 2015/09/21 } { }
674 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
675 \RequirePackage { fontspec }
676 \ifpackagelater { fontspec } { 2014/05/25 } { }
677 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

14.3.3.1 LuaTeX-ja 的默认设置

```

678 \ExplSyntaxOff

```

以下设置抄录自 ltjdefs.sty, 略有改动。

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

```

679 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
680 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
681 \ltjdefcharrange{3}{%
682   "2000-"206F, "2070-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF,
683   "2B00-"2BFF}
684 \ltjdefcharrange{4}{%
685   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
686   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
687   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
688 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
689 \ltjdefcharrange{6}{%
690   "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
691   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
692 \ltjdefcharrange{7}{%
693   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF,
694   "A830-"A83F, "AC00-"D7FF}
695 \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。

```

696 \ltjdefcharrange{9}{%
697   "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}

```

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

```

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

```

以下设置抄录自 ltj-latex.sty。

```

700 \directlua{
701   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
702   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
703 }
704 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
705   xkanjiskip=.25\zw plus 1pt minus 1pt,

```

```

706 autospacing, autoxspacing, jacharrange={-1},
707 yalbaselineshift=\z@, yjabaselineshift=\z@,
708 jcharwidowpenalty=500, differentjfm=paverage
709 }
710 \ExplSyntaxOn

```

14.3.3.2 LuaTeX-jā 的补丁

```
711 <@@=ctex_ltx>
```

在 L^AT_EX 下, LuaTeX-jā 对 fontspec、xunicode、unicode-math 和 listings 打了补丁。其中前三个是把 \char 换成 \ltjalchar, 确保字符是 ALchar 类。我们这里用 xunicode-addon 来处理 xunicode。

```

712 \RequirePackage { xunicode-addon }
713 \AtBeginUTFCommand
714 {
715   \group_begin:
716   \lua_now:e { tex.globaldefs = 0 }
717   \ltj@allalchar
718 }
719 \AtEndUTFCommand { \group_end: }

```

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

```

720 \cs_set_protected:Npn \fontspec_visible_space:
721 {
722   \tex_iffontchar:D \tex_font:D "2423 \exp_stop_f:
723   \ltjalchar "2423 \exp_stop_f:
724   \else:
725     \fontspec_visible_space_fallback:
726   \fi:
727 }

```

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

```

728 \ctex_at_end_package:nn { listings }
729 {
730   \use:x
731   {
732     \exp_not:N \RequirePackage { lltp-listings }
733     \tl_set:Nn \exp_not:N \lstlistingname
734     { \exp_not:o { \lstlistingname } }
735     \tl_set:Nn \exp_not:N \lstlistlistingname
736     { \exp_not:o { \lstlistlistingname } }
737   }
738 }

```

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。

```

739 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
740 {
741   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
742   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
743 }
744 \tl_new:N \CJK@family
745 \tl_new:N \l__ctex_ltj_current_font_tl
746 \tl_set:Nn \l__ctex_ltj_current_font_tl
747 { \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。

```

748 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
749 {
750   \group_begin:
751   \tl_set_eq:NN \f@encoding \CJK@encoding
752   \tl_set_eq:NN \f@family \CJK@family
753   \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
754   \ctex_ltj_pickup_font:
755   \group_end:
756   \font@name
757   \__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。

```

758   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
759   { \__ctex_ltj_select_font_aux: }
760 }
761 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
762 {
763   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
764   \cs_gset_nopar:Npx \font@name {#1}
765 }
766 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
767 { \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。

```

768 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
769 {
770   \exp_after:wN \cs_if_exist:NF \font@name
771   {
772     \group_begin:
773     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
774     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
775     \define@newfont
776     \group_end:
777   }
778 }
779 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

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

```

780 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
781 {
782   \get@external@font
783   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }

```

```

784     {
785         \tl_set:Nx \external@font
786         { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
787     }
788     \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。

```

789     \font@name
790     \lua_now:e { font.current(tex.getattribute('ltj@curjfont')) }
791     \use:c { \f@encoding + \f@family }
792     \use:c { \curr@fontshape }
793 }

```

`\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。

```

794 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
795 {
796   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
797   {
798     \group_begin:
799     \tl_set_eq:NN \CJK@family \f@family
800     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
801     {
802       \cs_gset_protected_nopar:Npx \subst@correction
803       {
804         \cs_new_eq:NN
805         \exp_not:c { \l__ctex_ltj_current_font_tl }
806         \font@name
807       }
808       \group_insert_after:N \group_insert_after:N
809       \group_insert_after:N \subst@correction
810     }
811     \group_end:
812   }
813 }

```

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

```

814 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
815 {
816   \lua_now:e { luatexja.jfont.does_alt_set ('lua_escape:e {#1}') }
817   \prg_return_true: \else: \prg_return_false: \fi:
818 }

```

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

```
function print_aftl_address()
```

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

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

```
819 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
820 { #1 \lua_now:e { luatexja.jfont.print_aftl_address() } ~ at }
```

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

```
821 \cs_new_protected_nopar:Npn \ctex_ltj_select_alterfont:
822 {
823   \ctex_ltj_if_alterfont_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
824   {
825     \lua_now:e
826     {
827       luatexja.jfont.output_alt_font_cmd
828       ('y', '\lua_escape:e { \l__ctex_ltj_current_shape_tl }')
829     }
830     \lua_now:e { luatexja.jfont.pickup_alt_font_a ('f@size') }
831   }
832 }
833 \tl_new:N \l__ctex_ltj_current_shape_tl
834 \tl_set:Nn \l__ctex_ltj_current_shape_tl
835 { \CJK@encoding / \CJK@family / \f@series / \f@shape }
```

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

```
836 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
837 {
838   \cs_if_exist:cF { #1/\f@size }
839   {
840     \group_begin:
841     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
842     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
843     \ctex_ltj_pickup_font:
844     \group_end:
845     \__ctex_ltj_pop_fontname:
846   }
847 }
848 <@@=>
```

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

```
849 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
850 {
851   \ltj@@getjfontnumber #1
852   \lua_now:e
853   {
854     luatexja.jfont.pickup_alt_font_b
855     ( \int_use:N \ltj@tempcntc, '\lua_escape:e {#2}' )
856   }
857 }
```

14.3.3.4 数学字体族

```
858 <@@=ctex_ltx>
```

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

\ctex_ltj_if_jfont:nTF 参数 #1 是一个 L^AT_EX 2_ε 编码名称或者字体命令。 L^AT_EX 2_ε 字体命令的一般形式是:

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

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

```

859 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
860 {
861   \lua_now:e
862   { luatexja.jfont.is_kenc( string.match('\lua_escape:e {#1}', '[^/]+') ) }
863   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
864 }

```

$\backslash ctex_ltj_if_jfont_math:N$ TF #1 是一个形式为 $\backslash M@ \langle encoding \rangle$ 的命令, 它由 $\backslash DeclareFontEncoding$ 的第三个参数来定义。

```

865 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
866 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
867 \group_begin:
868 \char_set_catcode_other:N M
869 \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
870 { \ctex_ltj_if_jfont:nTF {#3} }
871 \group_end:
872 <@@=>

```

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

```

873 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
874 {
875   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
876   { \ctex_ltj_get_and_define_fonts_ja:nN }
877   { \ctex_ltj_get_and_define_fonts_al:nN }
878   {#1} #2
879 }
880 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
881 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
882 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
883 {
884   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
885   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
886   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
887   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
888   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
889   \ctex_ltj_pickup_font:
890   \tl_put_right:Nx \math@fonts
891   {
892     \ltj@setpar@global
893     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
894     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
895     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
896   }
897 }
898 <@@=ctex_ltj>

```

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

```

899 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
900 {
901   \mode_if_math:T
902   {
903     \math@bgroup
904     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
905     \ctex_ltj_math_group_hook:
906     \ctex_ltj_if_jfont_math:NTF #1
907     { \jfam } { \mathgroup } #2 \scan_stop:
908     \math@egroup
909   }
910 }

```

$\backslash getanddefine@fonts$
 $\backslash ctex_ltj_get_and_define_fonts:nN$

$\backslash use@mathgroup$
 $\backslash ctex_ltj_use_math_group:Nn$

```

911 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
912 \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。

```

913 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
914 {
915   \group_begin:
916   \cs_set_protected:Npn #1 ##1##2##3
917     { \ltjsetmathletter { ##1 } }
918   #2
919   \group_end:
920 }
921 \ctex_at_end_package:n { unicode-math }
922 {
923   \cs_if_exist:NTF \um_input_math_symbol_table:
924   {
925     \ctex_ltj_set_math_letter:NN
926     \um_sym:nnn
927     \um_input_math_symbol_table:
928   }
929   {
930     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
931     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
932       { \__um_switchto_literal: }
933     \ctex_ltj_set_math_letter:NN
934     \__um_sym:nnn
935     \__um_input_math_symbol_table:
936   }
937 }

```

14.3.3.5 字体族的定义与使用

```

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

```

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

```

938 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
939 {
940   \str_if_eq:nnTF {#1} { plain }
941     { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
942     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
943 }
944 \tl_new:N \l__ctex_ltj_jfm_tl
945 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
946 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

```

\CJK@encoding
\__ctex_ltj_change_encoding:

```

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

```

947 \tl_const:Nn \CJK@encoding { LTJY3 }
948 \DeclareFontEncoding { \CJK@encoding } { } { }
949 \use:x
950 {
951   \exp_not:N \DeclareFontSubstitution
952     { \CJK@encoding } { song } { \mddefault } { \updefault }
953 }
954 \lua_now:e { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
955 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
956   { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
957 \DeclareFontFamily { \CJK@encoding } { song } { }
958 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }

```

```

959 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
960 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
961 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
962 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
963 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
964 { \CJK@encoding } { song } { \mddefault } { \updefault }
965 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
966 { \CJK@encoding } { song } { \bfdefault } { \updefault }
967 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
968 \jfam \c__ctex_ltj_math_fam_int

```

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

969 \newfontfeature { CID } { cid = #1 }
970 \newfontfeature { JFM } { jfm = #1 }
971 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

972 \keys_define:nn { fontspec-preparse-external }
973 {
974   NoEmbed .code:n =
975     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
976 }
977 \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。

```

978 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
979 {
980   \group_begin:
981   \clist_clear:N \l__ctex_ltj_char_range_clist
982   \seq_clear:N \l__ctex_ltj_alterate_seq
983   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
984   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
985   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
986   \ctex_ltj_set_alterate_family:nnF {#1} {#3}
987   {
988     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
989     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
990       {#1} { \l__ctex_ltj_font_options_clist }
991     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
992     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
993     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
994   }
995   \group_end:
996 }
997 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
998 \clist_new:N \l__ctex_ltj_font_options_clist
999 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1000 {
1001   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1002   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1003 }

```

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

```

\g__ctex_ltj_family_font_name_prop
\g__ctex_ltj_family_font_options_prop
1004 \prop_new:N \g__ctex_ltj_family_name_prop
1005 \prop_new:N \g__ctex_ltj_family_font_name_prop
1006 \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。

```

1007 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1008 {
1009   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1010   {

```



```

1011 \cs_undefine:c { \l__ctex_ltj_family_csname:n {#1} }
1012 \cs_undefine:c { \l__ctex_ltj_alternate_cs:n {#1} }
1013 \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1014 {
1015     \use:c { \l__ctex_ltj_alternate_cs:n { clear / #1 } }
1016     \cs_undefine:c { \l__ctex_ltj_alternate_cs:n { clear / #1 } }
1017     \cs_undefine:c { \l__ctex_ltj_alternate_cs:n { reset / #1 } }
1018     \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1019 }
1020 \msg_warning:nxxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1021 }
1022 }
1023 \tl_new:N \l__ctex_ltj_tmp_tl
1024 \msg_new:nnn { ctex } { redefine-family }
1025 { Redefining~CJKfamily~\l__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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

```

1026 \cs_new_protected_nopar:Npn \l__ctex_ltj_gset_family_cs:nn #1#2
1027 {
1028     \cs_gset_protected_nopar:cpx { \l__ctex_ltj_family_csname:n {#1} }
1029     {
1030         \group_begin:
1031         \l__ctex_ltj_change_encoding:
1032         \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1033         \exp_not:n { \cs_set_eq:NN \CTEX@fontfamily \use_none:n }
1034         \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1035         { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1036         \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1037         { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1038         \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1039         \exp_not:N \g__ctex_ltj_fontspec_family_tl
1040         \l__ctex_ltj_set_alternate_family:n {#1}
1041         \group_end:
1042     }
1043 }
1044 \tl_new:N \l__ctex_ltj_base_family_tl
1045 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1046 \cs_new_nopar:Npn \l__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1047 \cs_new_protected_nopar:Npn \l__ctex_ltj_set_alternate_family:n #1
1048 {
1049     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1050     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1051     \cs_if_exist_use:c { \l__ctex_ltj_alternate_cs:n { reset / #1 } }
1052     \cs_if_exist_use:c { \l__ctex_ltj_alternate_cs:n {#1} }
1053 }
1054 \cs_new:Npn \l__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

\CJKfamily 切换字体。

```

1055 \NewDocumentCommand \CJKfamily { m }
1056 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1057 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1058 {
1059     \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1060     {
1061         \tl_set:Nn \l__ctex_ltj_family_tl {#1}
1062         \selectfont
1063     }
1064     { \l__ctex_ltj_family_unknown_warning:n {#1} }
1065 }
1066 \tl_new:N \l__ctex_ltj_family_tl
1067 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

```

\ctex_ltj_family_if_exist:nNTF 判断 CJK 字体族 #1 是否存在, 若存在则把实际族名保存到 #2 中。
1068 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
1069 {
1070   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1071   { \prg_return_true: }
1072   {
1073     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1074     {
1075       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1076       \prg_return_true:
1077     }
1078     { \prg_return_false: }
1079   }
1080 }
1081 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { x } { T , F , TF }

\__ctex_ltj_family_unknown_warning:n 1082 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1083 {
1084   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1085   {
1086     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1087     {
1088       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1089       \msg_warning:nnn { ctex } { family-unknown } {#1}
1090     }
1091   }
1092 }
1093 \seq_new:N \g__ctex_ltj_unknown_family_seq
1094 \msg_new:nnn { ctex } { family-unknown }
1095 {
1096   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1097   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1098 }
1099 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1100 {
1101   \str_case:e:nnF {#1}
1102   {
1103     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1104     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1105     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1106   }
1107   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1108   [...] \{...\}
1109 }
1110 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1111 {
1112   \str_case:e:nnF {#1}
1113   {
1114     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1115     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1116     \CJKttdefault { \token_to_str:N \CJKttdefault }
1117   }
1118   {#1}
1119 }

\ctex_ltj_fontspec:nn 1120 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1121 {
1122   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1123   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1124   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1125   {
1126     \int_gincr:N \g__ctex_ltj_family_int
1127     \__ctex_ltj_fontspec:xnn
1128     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1129     {#1} {#2}
1130   }

```

```

1131 }
1132 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1133 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1134 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1135 {
1136   \bool_if:NT \l__ctex_ltj_add_altername_bool
1137   {
1138     \cs_if_free:cF
1139     { \__ctex_ltj_altername_cs:n { reset / \l_ctex_ltj_family_tl } }
1140     {
1141       \cs_gset_eq:cc
1142       { \__ctex_ltj_altername_cs:n { reset / #1 } }
1143       { \__ctex_ltj_altername_cs:n { reset / \l_ctex_ltj_family_tl } }
1144       \cs_gset_eq:cc
1145       { \__ctex_ltj_altername_cs:n { clear / #1 } }
1146       { \__ctex_ltj_altername_cs:n { clear / \l_ctex_ltj_family_tl } }
1147     }
1148     \bool_set_false:N \l__ctex_ltj_add_altername_bool
1149   }
1150   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1151   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1152   \ctex_ltj_switch_family:n {#1}
1153 }
1154 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1155 \prop_new:N \g__ctex_ltj_fontspec_prop

```

```

\ctex_ltj_add_font_features:n 1156 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1157 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1158 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1159 {
1160   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1161   {#1} \l__ctex_ltj_tmp_tl
1162   {
1163     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1164     {#1} \l__ctex_ltj_font_options_clist
1165     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1166     \bool_set_true:N \l__ctex_ltj_add_altername_bool
1167     \ctex_ltj_fontspec:xx
1168     { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1169     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1170   }
1171   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1172 }
1173 \bool_new:N \l__ctex_ltj_add_altername_bool
1174 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1175 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1176 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1177 {
1178   \token_to_str:N \addCJKfontfeature (s)~ignored.\
1179   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1180 }

```

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

```

1181 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1182 {
1183   \IfNoValueTF {#2}
1184   { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1185   {
1186     \use:x { #1 {#2} {#3} }
1187     #4
1188   }
1189 }
1190 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1191 {
1192   \use:x { #1 {#4} {#2} }
1193   #3

```

1194 }

```

\setCJKfamilyfont 1195 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 1196 {
  \CJKfontspec 1197   \__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 1198   { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1199   { }
1200 }
1201 \NewDocumentCommand \newCJKfontfamily { o m o m }
1202 {
1203   \tl_set:Nx \l__ctex_ltj_tmp_tl
1204   { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1205   \cs_new_protected_nopar:Npx #2
1206   { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1207   \__ctex_ltj_pass_args:nnnn
1208   { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1209   { }
1210 }
1211 \NewDocumentCommand \CJKfontspec { o m }
1212 {
1213   \__ctex_ltj_pass_args:nnnn
1214   { \ctex_ltj_fontspec:nn } {#1} {#2}
1215   { \tex_ignorespaces:D }
1216 }
1217 \NewDocumentCommand \addCJKfontfeatures { m }
1218 {
1219   \ctex_ltj_add_font_features:x {#1}
1220   \tex_ignorespaces:D
1221 }
1222 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\setCJKmainfont 1223 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1224 {
\setCJKmonofont 1225   \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1226   { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1227   { \normalfont }
1228 }
1229 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1230 \NewDocumentCommand \setCJKsansfont { o m }
1231 {
1232   \__ctex_ltj_pass_args:nnnn
1233   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1234   { \normalfont }
1235 }
1236 \NewDocumentCommand \setCJKmonofont { o m }
1237 {
1238   \__ctex_ltj_pass_args:nnnn
1239   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1240   { \normalfont }
1241 }
1242 \NewDocumentCommand \setCJKmathfont { o m }
1243 {
1244   \__ctex_ltj_pass_args:nnnn
1245   { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1246   { }
1247 }
1248 \NewDocumentCommand \defaultCJKfontfeatures { m }
1249 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1250 \clist_new:N \g__ctex_ltj_default_features_clist
1251 \@onlypreamble \setCJKmainfont
1252 \@onlypreamble \setCJKsansfont
1253 \@onlypreamble \setCJKmonofont
1254 \@onlypreamble \setCJKmathfont
1255 \@onlypreamble \setCJKromanfont
1256 \@onlypreamble \defaultCJKfontfeatures

1257 \tl_if_exist:NF \CJKfamilydefault

```

```

1258 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1259 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1260 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1261 \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。

```

1262 \RenewDocumentCommand \fontfamily { m }
1263 {
1264   \tl_set:Nx \f@family {#1}
1265   \CTEX@fontfamily {#1}
1266 }
1267 \cs_new_protected_nopar:Npn \CTEX@fontfamily #1
1268 {
1269   \str_if_eq:nnTF {#1} { \familydefault }
1270   { \CJKfamily { \CJKfamilydefault } }
1271   { \__ctex_ltj_update_family_aux: }
1272 }
1273 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_aux:
1274 {
1275   \str_case_e:nn { \f@family }
1276   {
1277     { \rmdefault } { \CJKfamily { \CJKrmdefault } }
1278     { \sfdefault } { \CJKfamily { \CJKsfdefault } }
1279     { \ttdefault } { \CJKfamily { \CJKttdefault } }
1280     { \familydefault } { \CJKfamily { \CJKfamilydefault } }
1281   }
1282 }

```

在导言区末尾更新 \CJKfamilydefault。

```

1283 \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。

```

1284 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1285 {
1286   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1287   {
1288     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1289     {
1290       \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
1291       { \use:n }
1292       {
1293         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1294         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1295         { \use:n }
1296       }
1297     }
1298     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1299     {
1300       \prop_map_break:n
1301       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1302     }
1303   }
1304 }
1305 \normalfont
1306 \ctex_ltj_update_mathfont:
1307 }
1308 }

```

\ctex_ltj_update_mathfont:

更新数学字体为实际的字体。

```

1309 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1310 {
1311   \ctex_ltj_family_if_exist:xNTF { \__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1312   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }

```

```

1313     {
1314         \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1315         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1316     }
1317 }
1318 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1319 {
1320     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1321     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1322     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1323     \cs_if_free:cTF
1324     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1325     {
1326         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1327         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1328     }
1329     {
1330         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1331         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1332     }
1333 }

```

14.3.3.6 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1334 \keys_define:nn { ctex_ltj / fontspec }
1335 {
1336     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1337     AlternateFont .value_required:n = true ,
1338     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1339     CharRange .value_required:n = true
1340 }

```

\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。

```

1341 \group_begin:
1342     \char_set_catcode_other:N \l
1343     \cs_set:Npn \__ctex_ltj_tmp:w #1
1344     {
1345         \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1346         {
1347             \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1348             {
1349                 \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1350                 \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1351                 \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1352                 \seq_set_filter:Nn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1353                 { ! \tl_if_blank_p:n { ###1 } }
1354                 \seq_concat:NNN \l__ctex_ltj_alternate_seq
1355                 \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1356             }
1357         }
1358     }
1359     \char_set_catcode_active:N \l
1360     \__ctex_ltj_tmp:w { || }
1361 \group_end:
1362 \seq_new:N \l__ctex_ltj_tmp_seq
1363 \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。

```

1364 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1365 {

```

```

1366 \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1367 {
1368   \__ctex_ltj_check_family:n {#1}
1369   \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1370   { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1371   #3
1372 }
1373 { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1374 }

```

保存由 `AlternateFont` 设置的替代字体序列。

```

1375 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1376 {
1377   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1378   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1379 }
1380 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1381 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1382 { m m m +0{ } } m u{ \q_stop } }
1383 {
1384   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1385   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1386   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1387   \tl_if_blank:nTF {#5}
1388   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1389   {
1390     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1391     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1392   }
1393   \use:x
1394   {
1395     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1396     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1397     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1398     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1399   }
1400 }
1401 \clist_new:N \l__ctex_ltj_alternate_options_clist

```

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。

```

1402 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1403 {
1404   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1405   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1406   \ctex_ltj_set_alternate_family:coonn
1407   { \__ctex_ltj_alternate_cs:n {#1} }
1408   { \l__ctex_ltj_char_range_clist }
1409   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1410 }
1411 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1412 {
1413   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1414   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1415   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1416 }
1417 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

保存替代字体序列的定义,以备定义主字体时使用。

```

1418 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1419 {
1420   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1421   \cs_gset_protected_nopar:Npx #1

```

```

1422     { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1423   }

```

实际定义替代字体族。

```

1424 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1425 {
1426   \group_begin:
1427   \__ctex_ltj_change_encoding:
1428   \cs_set_eq:NN \CJKfamily \use_none:n
1429   \ctex_ltj_swap_cs:NN
1430   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1431   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1432   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1433   \group_end:
1434 }
1435 \tl_new:N \l__ctex_ltj_alternate_family_tl

```

交换两个控制序列的意义。

```

1436 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1437 {
1438   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1439   \cs_set_eq:NN #1 #2
1440   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1441   \cs_undefine:N \__ctex_ltj_tmp:w
1442 }

```

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。

```

1443 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1444 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1445 {
1446   \int_gincr:N \g__ctex_ltj_family_int
1447   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1448 }
1449 \int_new:N \g__ctex_ltj_family_int

```

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>*。

```

1450 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1451 {
1452   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1453   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1454   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1455   { \l_fontspec_family_tl } {#3} {#4}
1456 }

```

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。

```

1457 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1458 {
1459   \clist_map_inline:Nn #1
1460   {
1461     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1462     {
1463       \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1464       \l__ctex_ltj_char_range_tl
1465     }
1466     { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1467   }

```



```

1468 \__ctex_ltj_save_alternate_shape:cnn
1469 { \__ctex_ltj_alternate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1470 { luatexja.jfont.clear_alt_font_latex }
1471 { '\lua_escape:e { \CJK@encoding/#2/#3/#4 }' }
1472 }

```

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。

```

1473 \NewDocumentCommand \ctex_ltj_set_alternate_shape:nnn
1474 { m m > { \SplitArgument { 1 } { -> } } m }
1475 { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
1476 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
1477 {
1478   \ctex_ltj_set_alternate_shape:n
1479   {
1480     \IfNoValueTF {#4}
1481     { \int_eval:n {#3} , \int_eval:n {#3} , }
1482     {
1483       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1484       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1485     }
1486     '\lua_escape:e { \CJK@encoding/#2 }' ,
1487     '\lua_escape:e { \CJK@encoding/#1 }'
1488   }
1489 }
1490 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1491 {
1492   \lua_now:e { luatexja.jfont.set_alt_font_latex ( #1 ) }
1493   \__ctex_ltj_save_alternate_shape:cnn
1494   { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1495   { luatexja.jfont.set_alt_font_latex } {#1}
1496 }

```

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

```

1497 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1498 {
1499   \tl_map_inline:Nn #3
1500   {
1501     \ctex_ltj_set_alternate_shape:n
1502     {
1503       ##1 ,
1504       '\lua_escape:e { \CJK@encoding/#2 }' ,
1505       '\lua_escape:e { \CJK@encoding/#1 }'
1506     }
1507   }
1508 }

```

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。

```

1509 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1510 {
1511   \group_begin:
1512   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1513   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1514   \cs_set_eq:NN \lua_escape:e \scan_stop:
1515   \cs_gset_protected_nopar:Npx #1
1516   { \exp_not:o {#1} \exp_not:N \lua_now:e { #2 ( #3 ) } }
1517   \group_end:
1518 }
1519 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nnn { c }

```

clearalternatefont
resetalternatefont

清除和重置操作总是全局的。

```

1520 \keys_define:nn { ctex }
1521 {
1522   clearalternatefont .code:n =
1523   { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1524   resetalternatefont .code:n =
1525   { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1526   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1527   resetalternatefont .default:n = \l_ctex_ltj_family_tl
1528 }
1529 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1530 {
1531   \group_begin:
1532   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1533   {
1534     \cs_if_exist_use:cT { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1535     {
1536       \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1537       {#1} { \l__ctex_ltj_base_family_tl }
1538       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1539       \selectfont
1540     }
1541   }
1542   { \__ctex_ltj_family_unknown_warning:n {#1} }
1543   \group_end:
1544 }
1545 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1546 {
1547   \group_begin:
1548   \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1549   {
1550     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1551     \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1552     \selectfont
1553   }
1554   \group_end:
1555 }
1556 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1557 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

declarecharrange

预先声明字符范围。

```

1558 \keys_define:nn { ctex }
1559 {
1560   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1561   declarecharrange .value_required:n = true
1562 }
1563 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1564 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1565 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1566 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1567 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

\ctex_ltj_declare_char_range:nn
\g__ctex_ltj_char_range_prop

#1 是名字, #2 是范围。

```

1568 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1569 {
1570   \tl_clear:N \l__ctex_ltj_char_range_tl
1571   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1572   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1573   \ctex_ltj_def_char_range_key:n {#1}
1574   \tl_clear:N \l__ctex_ltj_char_range_tl
1575 }
1576 \tl_new:N \l__ctex_ltj_char_range_tl
1577 \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。

```

1578 \NewDocumentCommand \ctex_ltj_save_char_range:n
1579 { > { \SplitArgument { 1 } { -> } } m }
1580 { \ctex_ltj_save_char_range:nn #1 }
1581 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1582 {
1583   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1584   { {
1585     \IfNoValueTF {#2}
1586     { \int_eval:n {#1} , \int_eval:n {#1} }
1587     {
1588       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1589       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1590     }
1591   } }
1592 }
```

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

```

1593 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1594 {
1595   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1596   {
1597     \keys_define:nn { ctex_ltj / fontspec }
1598     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1599   }
1600 }
```

`\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。

```

1601 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1602 {
1603   \tl_if_blank:nTF {#2}
1604   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1605   {
1606     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1607     {
1608       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1609       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1610     }
1611   }
1612 }
```

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

```

1613 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1614 { +o o u { \q_stop } }
1615 {
1616   \exp_args:NNf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1617   {
1618     \IfNoValueTF {#1} { {#3} }
1619     {
1620       \IfNoValueTF {#2}
1621       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ {#1} ] {#3} } }
1622       { [ {#1} ] { [ #2 ] } }
1623     }
1624   }
1625   \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1626 }
```

14.3.3.7 其它设置

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

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

1627 \AtBeginDocument
1628 {
1629   \ctex_appto_cmd:NnnTF \verbatim@font
1630   { \char_set_catcode_letter:n { 64 } }
1631   { \CTEX@verbatim@font@hook }
1632   { }
1633   { \ctex_patch_failure:N \verbatim@font }
1634 }
1635 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1636 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
```

`_ctex_ltjitaliccorr` L_AT_EX 的倾斜校正也要重新定义。

```
1637 <@@= >
1638 \cs_set_eq:NN \@@italiccorr \/
1639 <@@=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`。

```
\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。
\ctex_ltj_set_xkanjiskip:N \ltjsetxkanjiskip

1640 \cs_new_protected_nopar:Npn \ctex_ltj_set_kanjiskip:N
1641 { \ltj@setpar@global \ltjsetkanjiskip }
1642 \cs_new_protected_nopar:Npn \ctex_ltj_set_xkanjiskip:N
1643 { \ltj@setpar@global \ltjsetxkanjiskip }

1644 </luatex>
```

14.3.4 ctex-engine-uptex.def

```
1645 <*uptex|aptex>
```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```
1646 \tl_set:Nn \CJKrmdefault { zhrm }
1647 \tl_set:Nn \CJKsfdefault { zhsf }
1648 \tl_set:Nn \CJKttdefault { zhht }
1649 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1650 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }
1651 \RenewDocumentCommand \rmfamily { }
1652 {
1653   \not@math@alphabet \rmfamily \mathrm
1654   \romanfamily \rmdefault
1655   \kanjifamily \CJKrmdefault
1656   \selectfont
1657 }
1658 \RenewDocumentCommand \sffamily { }
1659 {
```

```

1660 \not@math@alphabet \sffamily \mathsf
1661 \romanfamily \sfdefault
1662 \kanjifamily \CJKsfdefault
1663 \selectfont
1664 }
1665 \RenewDocumentCommand \ttfamily { }
1666 {
1667 \not@math@alphabet \ttfamily \mathtt
1668 \romanfamily \ttdefault
1669 \kanjifamily \CJKttdefault
1670 \selectfont
1671 }
1672 \NewDocumentCommand \CJKfamily { m }
1673 {
1674 \kanjifamily {#1}
1675 \selectfont
1676 }

```

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

```

1677 \DeclareErrorKanjifont{JY2}{zhrm}{m}{n}{10}
1678 \DeclareKanjisubstitution{JY2}{zhrm}{m}{n}
1679 \DeclareKanjisubstitution{JT2}{zhrm}{m}{n}
1680 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
1681 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
1682 \jfam \symmincho

```

在导言区末尾更新 `\CJKfamilydefault`。

```

1683 \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。

```

1684 \normalfont

```

$\backslash\text{em}$ 取消 $\text{up}\text{\LaTeX}$ 对 $\backslash\text{em}$ 使用 `\mcfamily`、`\gtfamily` 命令的重定义, 恢复 $\text{\LaTeX}2_{\epsilon}$ 对 $\backslash\text{em}$ 的原始定义。如果用户已经重定义了 $\backslash\text{em}$, 则新定义保持不变。 $\text{up}\text{\LaTeX} 2016/05/07\text{u00}$ 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。

```

1685 \ctex_patch_cmd_once:NnnTF \em
1686 { \ExplSyntaxOff }
1687 { \eminnershape \else \gtfamily \itshape }
1688 { \eminnershape \else \itshape }
1689 { }
1690 {
1691 \ctex_patch_cmd:Nnn \em
1692 { \mcfamily \upshape \else \gtfamily \itshape }
1693 { \eminnershape \else \itshape }
1694 }
1695 \cs_set_nopar:Npn \eminnershape { \upshape }

```

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

```

1696 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1697 {
1698 \DeclareKanjifamily{JY2}{#1}{}
1699 \DeclareKanjifamily{JT2}{#1}{}
1700 \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1701 \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1702 \tl_if_empty:nF { #3 }
1703 {
1704 \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1705 \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1706 }
1707 }

```

```

1708 \@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。

1709 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1710 {
1711   \ctex_set_zhmap:n
1712   {
1713     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1714     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1715     \tl_if_empty:nF { #3 }
1716     {
1717       \special{ pdf:mapline~ #1b-h~  UniGB-UTF16-H~ #3 }
1718       \special{ pdf:mapline~ #1b-v~  UniGB-UTF16-V~ #3 }
1719     }
1720   }
1721 }
1722 \@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。

```

1723 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1724 {
1725   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1726   \ctex_set_upmap:nnn { upserifit } { #3 } {}
1727   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1728   \ctex_set_upmap:nnn { upmono } { #6 } {}
1729 }
1730 \@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。

```

1731 \RequirePackage { pxeveryysel }
1732 </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。

```

1733 <*/pdf|luatex|uptex|aptex>
1734 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1735 {
1736   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1737   {
1738     \group_begin:
1739     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1740     \tl_gset:Nx \CJKfamilydefault
1741     {
1742       \str_case:onF { \familydefault }
1743       {
1744         { \rmdefault } { \exp_not:N \CJKrmdefault }
1745         { \sfdefault } { \exp_not:N \CJKsfdefault }
1746         { \ttdefault } { \exp_not:N \CJKttdefault }
1747       }
1748       { \CJKfamilydefault }
1749     }
1750     \group_end:
1751   }

```

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

```

1752 <*luatex>
1753     \ctex_ltj_ensure_default_family:
1754 </luatex>
1755 }
```

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

```

1756 \tl_new:N \l__ctex_family_default_init_tl
1757 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1758 \tl_set:Nx \l__ctex_family_default_init_tl
1759 {
1760     \exp_not:N \__ctex_family_default_wrap:n
1761     { \exp_not:o { \CJKfamilydefault } }
1762 }
1763 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1764 </pdfTeX|uatTeX|uptTeX|aptTeX>
```

14.3.6 操作系统的判断

在 Lua_{TeX} 下直接用调用 `os.name` 来判断。

```

1765 <*luatex>
1766 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1767 {
1768     \tl_gset:Nx \g__ctex_fontset_tl
1769     {
1770         \lua_now:e
1771         {
1772             if ~ os.name == 'windows' then ~
1773                 tex.sprint ( 'windows' )
1774             elseif ~ os.name == 'macosx' then ~
1775                 tex.sprint ( 'mac' )
1776             else ~
1777                 tex.sprint ( 'fandol' )
1778             end
1779         }
1780     }
1781 }
1782 </luatex>
```

Ap_{TeX} 可以使用 `\ngostype` 来判断。

```

1783 <*aptex>
1784 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1785 {
1786     \tl_gset:Nx \g__ctex_fontset_tl
1787     {
1788         \str_case:onF { \ngostype }
1789         {
1790             { Win32 } { windows }
1791             { Win64 } { windows }
1792             { Darwin } { mac }
1793         }
1794         { fandol }
1795     }
1796 }
1797 </aptex>
```

pdf_{TeX} 和 X_Y_{TeX} 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!

```

1798 <*xetTeX|pdfTeX|uptTeX>
```

```

1799 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1800 {
1801   \file_if_exist:nTF { /dev/null }
1802   {
1803     \file_if_exist:nTF { nul: }
1804     {
1805       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1806       { \ctex_if_macosx:TF { mac } { windows } }
1807       { \ctex_if_macosx:TF { mac } { fandol } }
1808     }
1809     { \ctex_if_macosx:TF { mac } { fandol } }
1810   }
1811   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1812 }
1813 \str_const:Nx \c__ctex_upper_case_file_str
1814 { \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。

```

1815 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1816 {
1817   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1818   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1819   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1820 }
1821 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1822 </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。

```

1823 <*\class|ctex>
1824 \@ifpackageloaded { hyperref }
1825 {
1826   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1827   { \hypersetup {#1} }
1828 }
1829 {
1830   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1831   { \PassOptionsToPackage {#1} { hyperref } }
1832 }
1833 </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。

```

1834 <*\pdfTeX>
1835 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1836 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1837 {
1838   \ctex_hypersetup:n { CJKbookmarks = true }
1839   \sys_if_output_pdf:TF
1840   { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1841   {
1842     \ctex_at_end_package:nn { hyperref }

```



```

1843     {
1844         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1845         {
1846             \AtBeginShipoutFirst
1847             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1848         }
1849         { \RequirePackage { xCJK2uni } }
1850     }
1851 }
1852 }
1853 { \ctex_hypersetup:n { pdfencoding = unicode } }
1854 </pdfTeX>

```

在 $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ 下, `hyperref` 在处理带有非 ASCII 字符和 `\%` 的书签时有问题¹⁷。事实上, `hyperref` 在驱动文件 `hxeTeX.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...
```

$\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ 的维护者 Khaled Hosny 已经注意到了这个问题¹⁹。需要注意的是, `hxeTeX.def` 重载了宏包选项 `unicode`, 目的是不能设置它为 `false`, 但也导致它不会改变 `\HyPsd@pdfencoding`。如果 `hyperref` 先于 $\text{C}_{\text{E}}\text{T}_{\text{E}}\text{X}$ 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 $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ 下更保险的 `pdfencoding` 选项来设置。为了与 $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ 下的行为一致(使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 $\text{L}_{\text{u}}\text{a}\text{T}_{\text{E}}\text{X}$ 下也启用这个选项。

```

1855 <*xetex|luatex>
1856 \ctex_hypersetup:n { pdfencoding = unicode }
1857 </xetex|luatex>

```

我们假定 $\text{u}_{\text{p}}\text{T}_{\text{E}}\text{X}$ 使用 `DVIPDFMX` 驱动输出, 于是使用与 `pdfTEX` 类似的设置。注意 $\text{u}_{\text{p}}\text{T}_{\text{E}}\text{X}$ 需要使用 UTF8-UTF16 的编码转换。

```

1858 <*uptex|aptex>
1859 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1860 \ctex_at_end_package:nn { hyperref }
1861 { \AtBeginShipoutFirst { \special { pdf:tounicode~ UTF8-UTF16 } } }
1862 </uptex|aptex>

1863 <*pdfTeX|xetex|luatex|uptex|aptex>

```

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

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

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

14.3.8 CJKfntef、xeCJKfntef 相关设置

CT_EX 宏集对 pdf_TE_X 与 X_EL_AT_EX 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。

```

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

```

```

1922 }
1923 {
1924   \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1925   \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1926 }
1927 </pdfTeX>

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1928 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1929 <*pdfTeX|xetex>
1930 {
1931   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1932   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1933 }
1934 </pdfTeX|xetex>
1935 <*luatex>
1936 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1937 </luatex>
1938 <*uptex|aptex>
1939 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
1940 </uptex|aptex>
1941 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1942 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1943 <*pdfTeX|xetex>
1944 {
1945   \cs_set_protected_nopar:Npn \CJKglue
1946     { \skip_horizontal:N \l__ctex_ccglue_skip }
1947 }
1948 </pdfTeX|xetex>
1949 <*luatex>
1950 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
1951 </luatex>
1952 <*uptex|aptex>
1953 { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
1954 </uptex|aptex>
1955 \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

```

1956 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1957 {
1958 <*pdfTeX|xetex>
1959   \if_meaning:w \CJKglue \__ctex_ccglue:
1960     \prg_return_false: \else: \prg_return_true: \fi:
1961 </pdfTeX|xetex>
1962 <*luatex>
1963   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1964     { \prg_return_false: } { \prg_return_true: }
1965 </luatex>
1966 <*uptex|aptex>
1967   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
1968     { \prg_return_false: } { \prg_return_true: }
1969 </uptex|aptex>
1970 }

```

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 @@。

```

1971 <*pdfTeX|xetex>
1972 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1973 </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-jan 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。

```

1974 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
<pdf|xetex> 1975 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1976 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1977 { \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-jan 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。

```

1978 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1979 {
1980   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1981   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1982 }
1983 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1984 \if_cs_exist:N \@EverySelectfont@Init
1985   \group_begin:
1986   \cs_set:Npn \__ctex_tmp:N #1
1987   {
1988     \tl_set:Nn \l__ctex_tmp_tl {#1}
1989     \cs_new_eq:NN \CTEX@selectfont@save #1
1990     \cs_new_protected_nopar:Npn \__ctex_restore_selectfont:
1991     {
1992       \tl_put_left:Nn \@EverySelectfont@Init
1993       { \let #1 \CTEX@selectfont@save }
1994       \cs_undefine:N \__ctex_restore_selectfont:
1995     }
1996   }
1997   \ctex_parse_name:NN \__ctex_tmp:N \selectfont
1998   \exp_last_unbraced:NNo \group_end:
1999   \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
2000   { \ExplSyntaxOff }
2001   { \size@update }
2002   { \CTEX@selectfont@hook \size@update }
2003   { \__ctex_restore_selectfont: }
2004   { \ctex_patch_failure:N \selectfont }
2005 \fi:

```

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

```

2006 <*pdf|xetex>
2007 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
2008 </pdf|xetex>
2009 <*xetex|luatex|uptex|aptex>
2010 \EverySelectfont { \CTEX@selectfont@hook }
2011 </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。

```

2012 <*luatex>
2013 \ctex_add_to_selectfont:n

```

```

2014 {
2015   \ctex_ltj_select_font:
2016   \ctex_ltj_select_alterate_font:
2017 }
2018 \tl_set:Nn \CJK@family { song } \selectfont
2019 \tl_clear:N \CJK@family
2020 </luatex>

```

\ctex_update_xkanjiskip:
\l__ctex_xkanjiskip_skip

upTeX 和 LuaTeX-jā 对 \xkanjiskip 都是即时赋值。单位 zw 与字体相关，因此需要每次 \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。

```

2021 <*luatex|uptex|aptex>
2022 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
2023 {
2024   \skip_if_eq:nnT
2025   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
2026   { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2027   {
2028     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
2029     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
2030     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2031   }
2032 }
2033 \tl_new:N \l__ctex_xkanjiskip_tl
2034 \tl_set:Nn \l__ctex_xkanjiskip_tl
2035 { .25\zw plus 1pt minus 1pt }
2036 { .25zw plus 1pt minus 1pt }
2037 \skip_new:N \l__ctex_xkanjiskip_skip
2038 \skip_set:Nn \l__ctex_xkanjiskip_skip
2039 { \ltjgetparameter { xkanjiskip } }
2040 { \tex_xkanjiskip:D }

2041 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2042 </luatex|uptex|aptex>

```

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。

\cht
\cdp
\cwd

```

2043 <*luatex>
2044 \dim_new:N \cht
2045 \dim_new:N \cdp
2046 \dim_new:N \cwd
2047 \newluafunction \g__ctex_kanjisize_func
2048 \group_begin:
2049 \char_set_catcode_space:n { 32 }
2050 \lua_now:e
2051 {
2052   local nulltable = { }
2053   local t = lua.get_functions_table()
2054   local fmt = luatexja.jfont.font_metric_table
2055   local getattribute = tex.getattribute
2056   local setdimen = tex.setdimen
2057   t[\int_use:N \g__ctex_kanjisize_func] = function ()
2058     local ft = fmt[getattribute('ltj@curjfont')] or nulltable
2059     local ft = ft and ft.char_type or nulltable
2060     local fk = ft and ft[0] or nulltable
2061     setdimen('cht', fk.height or 0)
2062     setdimen('cdp', fk.depth or 0)
2063     setdimen('cwd', fk.width or ft.zw or 0)
2064   end
2065 }
2066 \group_end:
2067 \cs_new_protected_nopar:Npn \ctex_update_kanjisize:
2068 { \tex_luafunction:D \g__ctex_kanjisize_func }

```

```

2069 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2070 </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。

```

2071 \keys_define:nn { ctex }
2072 {
2073   <*pdf|xtex>
2074     space .choice: ,
2075     space / true .code:n =
<pdf|xtex> 2076       { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<xtex> 2077       { \xeCJKsetup { CJKspace = true } } ,
2078     space / auto .code:n =
<pdf|xtex> 2079       { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
<xtex> 2080       { \xeCJKsetup { CJKspace = false } } ,
2081     space / false .code:n =
<pdf|xtex> 2082       { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
<xtex> 2083       { \xeCJKsetup { CJKspace = false } } ,
2084     space .default:n = { true } ,
2085     space .initial:n = { auto }
2086   </pdf|xtex>
2087   <*lua|u|up|ap|tex>
2088     space .code:n =
2089     { \msg_warning:nn { ctex } { invalid-option } }
2090   </lua|u|up|ap|tex>
2091 }

```

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

```

2092 \keys_define:nn { ctex }
2093 {
2094   punct .code:n =
2095   {
2096     \tl_set:Nx \l__ctex_punct_tl { #1 }
<pdf|xtex> 2097     \punctstyle { \l__ctex_punct_tl }
<xtex> 2098     \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
<luatex> 2099     \ctex_mono_jfm:o { \l__ctex_punct_tl }
<uptex|ap|tex> 2100     \msg_warning:nn { ctex } { invalid-option }
2101   } ,
2102   punct .default:n = { quanjiao } ,
2103 }

```

XeLaTeX、LuaLaTeX 和 upLaTeX 总是使用 UTF8 编码。

```

2104 <*xtex|lua|u|up|ap|tex>
2105 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2106 </xtex|lua|u|up|ap|tex>
2107 </pdf|xtex|lua|u|up|ap|tex>

```

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> 2108 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.4 用户设置接口

```

<ctexset> 2109 <*class|ctex|ctexheading>
2110 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
2111 </class|ctex|ctexheading>

```

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

`\CTEXoptions`

```

2112 <*class|ctex>
2113 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
2114 {
2115   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
2116   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
2117   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
2118 }
2119 \NewDocumentCommand \CTEXoptions { +o }
2120 {
2121   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
2122   { \ctexset~ {~ #1~ }~ is~ set. }
2123   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
2124 }

```

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。

```

2125 \keys_define:nn { ctex }
2126 {
2127   autoindent .choice: ,
2128   autoindent .default:n = { true } ,
2129   autoindent / true .code:n =
2130   {
2131     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2132     \ctex_select_size:
2133   } ,
2134   autoindent / false .code:n =
2135   { \tl_clear:N \l__ctex_autoindent_tl } ,
2136   autoindent / unknown .code:n =
2137   {
2138     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2139     \ctex_select_size:
2140   }
2141 }

```

`\CTEXsetfont` 无论字体大小是否变化都更新相关信息。

```

2142 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2143 \cs_new_protected_nopar:Npn \ctex_select_size:
2144 { \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。

```

2145 \cs_new_protected_nopar:Npn \ctex_update_size:
2146 {
2147   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2148   {
2149     \ctex_update_stretch:
2150     \ctex_update_parindent:
2151   }
2152   { \ctex_update_ziju: }
2153 }
2154 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2155 \tl_new:N \l__ctex_ziju_tl
2156 \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。

```

2157 \ctex_add_to_selectfont:n
2158 { \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`。

```

2159 \keys_define:nn { ctex }
2160 {
2161   linestretch .code:n =
2162   {
2163     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2164     \ctex_select_size:
2165   } ,
2166   linestretch .value_required:n = true
2167 }
2168 \tl_new:N \l__ctex_line_stretch_tl
2169 \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。

```

2170 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2171 {
2172   \ctex_update_em_unit:
2173   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2174   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2175   {
2176     \ctex_if_ccglue_touched:TF
2177     { \ctex_update_ccwd: }
2178     {
2179       \dim_set:Nn \l__ctex_tmp_dim
2180       { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
2181       \skip_set:Nn \l__ctex_ccglue_skip
2182       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2183       \ctex_update_ccglue:
2184     }
2185   }
2186   {
2187     \int_set:Nn \l__ctex_tmp_int
2188     { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2189     \tex_divide:D \l__ctex_tmp_int \ccwd
2190     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
2191     {
2192       \skip_set:Nn \l__ctex_ccglue_skip
2193       {
2194         \c_zero_dim plus \dim_eval:n
2195         {
2196           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2197           \l__ctex_tmp_int
2198         }
2199       }
2200     }
2201     { \skip_zero:N \l__ctex_ccglue_skip }
2202     \ctex_update_ccglue:
2203   }
2204 }

```

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

```

2205 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2206 {
2207   \tl_if_empty:NF \l__ctex_autoindent_tl
2208   {

```



```

2209         \dim_compare:nNnF \parindent = \c_zero_dim
2210         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2211     }
2212 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2213 \NewDocumentCommand \ziju { m }
2214 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2215 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2216 {
2217     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2218     \ctex_select_size:
2219 }

```

`\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。

```

2220 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2221 {
2222     \ctex_update_em_unit:
2223     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2224     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2225     \dim_compare:nNnTF \ccwd > \c_zero_dim

```

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

```

2226     {
2227         \dim_set:Nn \l__ctex_tmp_dim
2228         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2229         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2230         \tex_divide:D \l__ctex_tmp_int \ccwd
2231         \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。

```

2232         \dim_compare:nNnF \parindent = \c_zero_dim
2233         {
2234             \int_compare:nNnF \l__ctex_tmp_int < 3
2235             { \int_sub:Nn \l__ctex_tmp_int { 2 } }
2236         }
2237         \skip_set:Nn \l__ctex_ccglue_skip
2238         {
2239             \l__ctex_ziju_dim
2240             plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2241             minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2242             { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
2243         }
2244     }
2245     { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2246     \ctex_update_ccglue:

```

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

```

2247     \ctex_update_parindent:
2248 }
2249 \dim_new:N \l__ctex_ziju_dim

```

`\CTEXindent`
`\CTEXnoindent` 过时命令。

```

2250 \NewDocumentCommand \CTEXindent { }

```

```

2251 {
2252   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2253   { \parindent is~ set~ to~ 2\ccwd. }
2254   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2255 }
2256 \NewDocumentCommand \CTEXnoindent { }
2257 {
2258   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2259   { \parindent is~ set~ to~ 0pt. }
2260   \dim_zero:N \parindent
2261 }

```

14.6 中文数字与日期

```

2262 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2263 \RequirePackage { zhnumber }

\chinese 2264 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2265 \cs_new_eq:NN \@chinese \@zhnum
2266 \cs_new_eq:NN \Chinese \chinese
2267 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

```

2268 \ctex_at_end_package:nn { enumitem }
2269 {
2270   \cs_if_free:NF \AddEnumerateCounter
2271   {
2272     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2273     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2274     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2275   }
2276 }

\CTEXnumber 2277 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 2278 { \protected@edef #1 { \zhnumber {#2} } }
2279 \NewDocumentCommand \CTEXdigits { m m }
2280 { \protected@edef #1 { \zhdigits {#2} } }

today 2281 \cs_set_eq:NN \CTEX@todayold \today
2282 \keys_define:nn { ctex }
2283 {
2284   today .choice: ,
2285   today / old .code:n =
2286   { \cs_set_eq:NN \today \CTEX@todayold } ,
2287   today / small .code:n =
2288   {
2289     \cs_set_eq:NN \today \zhtoday
2290     \zhnumsetup { time = Arabic }
2291   } ,
2292   today / big .code:n =
2293   {
2294     \cs_set_eq:NN \today \zhtoday
2295     \zhnumsetup { time = Chinese }
2296   } ,
2297   today / unknown .code:n =
2298   { \msg_error:nxx { ctex } { today-undef } {#1} }
2299 }
2300 \msg_new:nnnn { ctex } { today-undef }
2301 { Today~format~`#1'~is~undefined. }
2302 { 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。

```

2303 \tl_if_exist:NF \proofname

```

```

2304 {
2305   \tl_new:N \proofname
2306   \tl_set:Nn \proofname { Proof }
2307 }

2308 \keys_define:nn { ctex }
2309 {
2310   contentsname .tl_set:N = \contentsname ,
2311   listfigurename .tl_set:N = \listfigurename ,
2312   listtablename .tl_set:N = \listtablename ,
2313   figurename .tl_set:N = \figurename ,
2314   tablename .tl_set:N = \tablename ,
2315   abstractname .tl_set:N = \abstractname ,
2316   indexname .tl_set:N = \indexname ,
2317   appendixname .tl_set:N = \appendixname ,
2318   proofname .tl_set:N = \proofname ,
2319   bibname .tl_set:N = \refname
<article>
<book|report> 2320   bibname .tl_set:N = \bibname
2321 <*beamer>
2322   algorithmname .tl_set:N = \algorithmname ,
2323   bibname .tl_set:N = \bibname ,
2324   refname .tl_set:N = \refname ,
2325   continuation .tl_set:N = \insertcontinuationtext
2326 </beamer>
2327 }

2328 <*ctex>
2329 \msg_new:nnn { ctex } { ctexbibname }
2330 {
2331   Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.~\
2332   The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2333 }
2334 \tl_if_exist:NTF \insertcontinuationtext
2335 {
2336   \keys_define:nn { ctex }
2337   {
2338     algorithmname .tl_set:N = \algorithmname ,
2339     bibname .tl_set:N = \bibname ,
2340     refname .tl_set:N = \refname ,
2341     continuation .tl_set:N = \insertcontinuationtext
2342   }
2343 }
2344 {
2345   \tl_if_exist:NTF \bibname
2346   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2347   {
2348     \tl_if_exist:NTF \refname
2349     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2350     {
2351       \msg_warning:nn { ctex } { ctexbibname }
2352       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2353     }
2354   }
2355 }
2356 </ctex>
2357 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 CTeX 文档类或打开 heading 选项下生效。

```

2358 <*class|heading>

```

14.8.1 定义标题格式选项

保存 \section 级以下标题名字。

```

\c__ctex_section_headings_seq
2359 <*article|book|report>
2360 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
2361 { section , subsection , subsubsection , paragraph , subparagraph }
2362 </article|book|report>

\c__ctex_headings_seq
2363 <*article|book|report>
2364 \seq_new:N \c__ctex_headings_seq
2365 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2366 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2367 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2368 </article|book|report>
2369 <*beamer>
2370 \seq_const_from_clist:Nn \c__ctex_headings_seq
2371 { part , section , subsection }
2372 </beamer>

\__ctex_initial_heading:n
2373 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2374 {
2375   \tl_new:c { CTEX@pre#1 }
2376   \tl_new:c { CTEX@post#1 }
2377   \tl_const:cx { CTEXthe#1 }
2378   {
2379     \exp_not:c { CTEX@pre#1 }
2380     \exp_not:c { CTEX@the#1 }
2381     \exp_not:c { CTEX@post#1 }
2382   }
2383   \tl_const:cx { CTEX@#1name }
2384   {
2385     \group_begin:
2386     \exp_not:c { CTEX@#1nameformat }
2387     {
2388       \exp_not:c { CTEX@pre#1 }
2389       \exp_not:N \tl_if_empty:NTF
2390       \exp_not:c { CTEX@#1numberformat }
2391       { \exp_not:c { CTEX@the#1 } }
2392       {
2393         \group_begin:
2394         \exp_not:c { CTEX@#1numberformat }
2395         \exp_not:c { CTEX@the#1 }
2396         \group_end:
2397       }
2398       \exp_not:c { CTEX@post#1 }
2399     }
2400   \group_end:
2401 }
2402 }

\__ctex_def_heading_keys:n
2403 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2404 {
2405   \tl_put_right:Nx \l__ctex_tmp_tl
2406   {
2407     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2408     #1 / name .code:n =
2409     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2410     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2411     #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1@beforekip } ,
2412     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2413     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2414     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2415     #1 / numbering .initial:n = true ,
2416     #1 / beforekip .initial:n = \c_zero_skip ,

```

```

2417      #1 / afterskip      .initial:n = \c_zero_skip ,
2418      #1 / indent        .initial:n = \c_zero_dim ,
2419      #1 / beforeskip     .value_required:n = true ,
2420      #1 / afterskip      .value_required:n = true ,
2421      #1 / indent        .value_required:n = true ,
2422 <*article|book|report>
2423      #1 / afterindent     .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2424      #1 / fixskip        .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2425      #1 / hang           .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2426      #1 / hang           .initial:n = true ,
2427      #1 / runin          .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2428      #1 / tocline        .code:n =
2429      {
2430          \cs_set:Npn \exp_not:c { CTEX@#1@tocline}
2431          \exp_not:n { #####1####2 } { #####1 }
2432      } ,
2433      \__ctex_plus_key_aux:nn {#1} { break } ,
2434 </article|book|report>
2435      \__ctex_plus_key_aux:nn {#1} { format } ,
2436      \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2437      \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2438      \__ctex_plus_key_aux:nn {#1} { titleformat } ,
2439      \__ctex_plus_key_aux:nn {#1} { aftername } ,
2440      \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2441  }
2442 }
2443 \cs_new_nopar:Npn \__ctex_plus_key_aux:nn #1#2
2444 {
2445     #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2446     #1 / #2 + .code:n =
2447     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2448     #1 / #2 ~ + .code:n =
2449     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2450 }

```

\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 L^AT_EX3 的 clist 总是会自动忽略空元素，所以设置 name={, 章} 后，第一个元素将会是“章”，必须用空的分组保护空元素：name={{} , 章}，这在使用中有些许不便。我们可以改用 seq 或者手写函数解析参数来加以改进。为实现的简单起见，这里用了 xparse 的 \SplitArgument，它带有参数的长度检查。

```

2451 \NewDocumentCommand \ctex_assign_heading_name:nn
2452 { m > { \SplitArgument { 1 } { , } } +m }
2453 { \__ctex_assign_heading_name:nnn {#1} #2 }
2454 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2455 {
2456     \tl_set:cn { CTEX@pre#1 } {#2}
2457     \IfNoValueTF {#3}
2458     { \tl_clear:c { CTEX@post#1 } }
2459     { \tl_set:cn { CTEX@post#1 } {#3} }
2460 }

```

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

只在 ctexbook 和 ctexrep 下有定义。

```

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

```

```

2473 }
2474 </book|report>
2475 <*article|beamer>
2476 \tl_clear:N \l__ctex_tmp_tl
2477 </article|beamer>

```

定义标题键值选项。

```

2478 \seq_map_inline:Nn \c__ctex_headings_seq
2479 {
2480   \__ctex_initial_heading:n {#1}
2481   \__ctex_def_heading_keys:n {#1}
2482 }
2483 \use:x
2484 {
2485   \group_end:
2486   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2487 }
2488 <*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。

```

2489 <*book|report>
2490 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2491 {
2492   \CTEX@fixheadingskip
2493   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2494     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2495 }
2496 </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 `afterkip` 作为修正。

```

2497 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2498 {
2499   \par
2500   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2501   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2502 }
2503 \skip_new:N \l__ctex_heading_skip
2504 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2505 { \skip_set:Nn \l__ctex_heading_skip }
2506 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\partmark` 提供 `\partmark`。

```

2507 \ProvideDocumentCommand \partmark { m }
2508 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2509 \cs_new_eq:NN \CTEXifname \use_ii:nn
2510 \cs_new_protected_nopar:Npn \CTEX@ifnametrue
2511 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2512 \cs_new_protected_nopar:Npn \CTEX@ifnamefalse
2513 { \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。

```

2514 <*book|report>
2515 \cs_new_protected_nopar:Npn \CTEX@addloflotskip #1
2516 {
2517   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2518   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2519   {
2520     \addtocontents { lof }
2521     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2522   }
2523   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2524   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2525   {
2526     \addtocontents { lot }
2527     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2528   }
2529 }
2530 </book|report>

```

`\CTEX@addtocline` 2531 `\cs_new_protected:Npn \CTEX@addtocline #1#2`
 2532 `{ \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }`

`\CTEX@disableautoindent` 禁用自动调整首行缩进。

```

2533 \cs_new_protected_nopar:Npn \CTEX@disableautoindent
2534 { \tl_clear:N \l__ctex_autoindent_tl }

```

14.8.2.1 part 的标题

```

2535 <@@=>
\part 2536 <*article>
2537 \renewcommand\part{%
2538   \if@noskipsec \leavevmode \fi
2539   \par
2540   \CTEX@part@break
2541   % \addvspace{4ex}%
2542   \CTEX@setheadingskip \CTEX@part@beforeskip
2543   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2544   \addvspace \CTEX@headingskip
2545   \ifodd \CTEX@part@afterindent
2546     \@afterindenttrue
2547   \else
2548     \@afterindentfalse
2549   \fi
2550   \secdef\@part\@spart}
2551 </article>
2552 <*book|report>
2553 \renewcommand\part{%
2554   % \if@openright
2555   % \cleardoublepage
2556   % \else
2557   % \clearpage
2558   % \fi
2559   \CTEX@part@break
2560   % \thispagestyle{plain}%
2561   \thispagestyle{\CTEX@part@pagestyle}%
2562   \if@twocolumn
2563     \onecolumn
2564     \@tempwatrue
2565   \else
2566     \@tempwafalse
2567   \fi
2568   % \null\vfil
2569   \CTEX@setheadingskip \CTEX@part@beforeskip

```

```

2570 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2571 \vspace*{\CTEX@headingskip}%
2572 \secdef\@part\@spart}
2573 </book|report>

\@part 2574 <*article>
2575 \def\@part[#1]#2{%
2576 \ifnum \c@secnumdepth >\m@ne
2577 \ifodd \CTEX@part@numbering
2578 \CTEX@ifnametrue
2579 \refstepcounter{part}%
2580 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2581 \else
2582 \CTEX@ifnamefalse
2583 \CTEX@makeanchor{part*}%
2584 % \addcontentsline{toc}{part}{#1}%
2585 \fi
2586 \else
2587 \CTEX@ifnamefalse
2588 \CTEX@makeanchor{part*}%
2589 % \addcontentsline{toc}{part}{#1}%
2590 \fi
2591 \CTEX@getttitle{#1}%
2592 \CTEX@addtocline{part}{#1}%
2593 {\interlinepenalty \@M
2594 % \normalfont \parindent \z@ \raggedright
2595 \CTEX@disableautoindent
2596 \normalfont \CTEX@part@format
2597 % \ifnum \c@secnumdepth >\m@ne
2598 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2599 % \fi
2600 \CTEX@hangindent{part}%
2601 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2602 % \huge\bfseries #2%
2603 \CTEX@part@titleformat{#2}%
2604 % \markboth{}{}%
2605 \partmark{#1}%
2606 \CTEX@part@aftertitle}%
2607 \nobreak
2608 % \vskip 3ex
2609 \CTEX@setheadingskip \CTEX@part@afterskip
2610 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2611 \vskip \CTEX@headingskip
2612 \@afterheading}
2613 </article>
2614 <*book|report>
2615 \def\@part[#1]#2{%
2616 \ifnum \c@secnumdepth >-2\relax
2617 \ifodd \CTEX@part@numbering
2618 \CTEX@ifnametrue
2619 \refstepcounter{part}%
2620 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2621 \else
2622 \CTEX@ifnamefalse
2623 \CTEX@makeanchor{part*}%
2624 % \addcontentsline{toc}{part}{#1}%
2625 \fi
2626 \else
2627 \CTEX@ifnamefalse
2628 \CTEX@makeanchor{part*}%
2629 % \addcontentsline{toc}{part}{#1}%
2630 \fi
2631 \CTEX@getttitle{#1}%
2632 \CTEX@addtocline{part}{#1}%
2633 % \markboth{}{}%
2634 \partmark{#1}%
2635 {\interlinepenalty \@M

```



```

2636 % \normalfont \centering
2637 \CTEX@disableautoindent
2638 \normalfont \CTEX@part@format
2639 % \ifnum \c@secnumdepth >-2\relax
2640 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2641 % \fi
2642 \CTEX@hangindent{part}%
2643 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2644 % \Huge\bfseries #2\par}%
2645 \CTEX@part@titleformat{#2}%
2646 \CTEX@part@aftertitle}%
2647 \endpart}
2648 </book|report>

\@spart 2649 <*article>
2650 \def\@spart#1{%
2651     \CTEX@ifnamefalse
2652     \CTEX@makeanchor@spart{part*}%
2653     \CTEX@getttitle{#1}%
2654     {\interlinepenalty \M
2655 % \normalfont \parindent \z@ \raggedright
2656 \CTEX@disableautoindent
2657 \normalfont \CTEX@part@format
2658 \CTEX@hangindent{part}{}%
2659 % \huge \bfseries #1\par}%
2660 \CTEX@part@titleformat{#1}%
2661 \CTEX@part@aftertitle}%
2662 \nobreak
2663 % \vskip 3ex
2664 \CTEX@setheadingskip \CTEX@part@afterskip
2665 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2666 \vskip \CTEX@headingskip
2667 \@afterheading}
2668 </article>
2669 <*book|report>
2670 \def\@spart#1{%
2671     \CTEX@ifnamefalse
2672     \CTEX@makeanchor@spart{part*}%
2673     \CTEX@getttitle{#1}%
2674     {\interlinepenalty \M
2675 % \normalfont \centering
2676 \CTEX@disableautoindent
2677 \normalfont \CTEX@part@format
2678 \CTEX@hangindent{part}{}%
2679 % \Huge \bfseries #1\par}%
2680 \CTEX@part@titleformat{#1}%
2681 \CTEX@part@aftertitle}%
2682 \endpart}
2683 </book|report>

\@endpart 2684 <*book|report>
2685 \def\@endpart{%
2686 % \vfil
2687 \CTEX@setheadingskip \CTEX@part@afterskip
2688 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2689 \vskip \CTEX@headingskip
2690 \newpage
2691 \if@twoside
2692 \if@openright
2693 \null
2694 \thispagestyle{empty}%
2695 \newpage
2696 \fi
2697 \fi
2698 \if@tempswa
2699 \twocolumn
2700 \fi}

```

2701 `</book|report>`

14.8.2.2 chapter 的标题

2702 `<*book|report>`

```

\chapter 2703 \renewcommand\chapter{%
2704 % \if@openright\cleardoublepage\else\clearpage\fi
2705 % \thispagestyle{plain}%
2706 \CTEX@chapter@break
2707 \thispagestyle{\CTEX@chapter@pagestyle}%
2708 \global\@topnum\z@
2709 % \@afterindentfalse
2710 \ifodd \CTEX@chapter@afterindent
2711 \@afterindenttrue
2712 \else
2713 \@afterindentfalse
2714 \fi
2715 \secdef\@chapter\@schapter}

\@chapter 2716 \def\@chapter[#1]#2{%
2717 \ifnum \c@secnumdepth >\m@ne
2718 <*book>
2719 \if@mainmatter
2720 </book>
2721 \ifodd \CTEX@chapter@numbering
2722 \CTEX@ifnametrue
2723 \refstepcounter{chapter}%
2724 % \typeout{\@chapapp\space\thechapter.}%
2725 \typeout{\CTEXthechapter}%
2726 % \addcontentsline{toc}{chapter}
2727 % {\protect\numberline{\thechapter}#1}%
2728 \else
2729 \CTEX@ifnamefalse
2730 \CTEX@makeanchor{\Hy@chapapp*}%
2731 % \addcontentsline{toc}{chapter}{#1}%
2732 \fi
2733 <*book>
2734 \else
2735 \CTEX@ifnamefalse
2736 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2737 % \addcontentsline{toc}{chapter}{#1}%
2738 \fi
2739 </book>
2740 \else
2741 \CTEX@ifnamefalse
2742 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2743 % \addcontentsline{toc}{chapter}{#1}%
2744 \fi
2745 \CTEX@getttitle{#1}%
2746 \CTEX@addtocline{chapter}{#1}%
2747 \chaptermark{#1}%
2748 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2749 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2750 \CTEX@addloflotskip{chapter}%
2751 \if@twocolumn
2752 \@topnewpage[\@makechapterhead{#2}]%
2753 \else
2754 \@makechapterhead{#2}%
2755 \@afterheading
2756 \fi}

\@schapter 2757 \def\@schapter#1{%
2758 \CTEX@ifnamefalse
2759 \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2760 \CTEX@getttitle{#1}%
2761 \if@twocolumn

```

```

2762 \topnewpage[\@makeschapterhead{#1}]%
2763 \else
2764 \makeschapterhead{#1}%
2765 \afterheading
2766 \fi}

\@makechapterhead 2767 \def\@makechapterhead#1{%
2768 % \vspace*{50\p@}%
2769 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2770 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2771 \vspace*{\CTEX@headingskip}%
2772 % {\normalfont \parindent \z@ \raggedright
2773 {\CTEX@disableautoindent
2774 \normalfont \CTEX@chapter@format
2775 \interlinepenalty\@M
2776 % \ifnum \c@secnumdepth >\m@ne
2777 % \if@mainmatter
2778 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2779 % \fi
2780 % \fi
2781 \CTEX@hangindent{chapter}%
2782 {\CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
2783 % \Huge \bfseries #1\par\nobreak
2784 \CTEX@chapter@titleformat{#1}%
2785 \CTEX@chapter@aftertitle
2786 \nobreak
2787 % \vskip 40\p@
2788 \CTEX@setheadingskip \CTEX@chapter@afterskip
2789 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2790 \vskip \CTEX@headingskip
2791 }}

\@makeschapterhead 2792 \def\@makeschapterhead#1{%
2793 % \vspace*{50\p@}%
2794 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2795 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2796 \vspace*{\CTEX@headingskip}%
2797 % {\normalfont \parindent \z@ \raggedright
2798 {\CTEX@disableautoindent
2799 \normalfont \CTEX@chapter@format
2800 \interlinepenalty\@M
2801 \CTEX@hangindent{chapter}{}%
2802 % \Huge \bfseries #1\par\nobreak
2803 \CTEX@chapter@titleformat{#1}%
2804 \CTEX@chapter@aftertitle
2805 \nobreak
2806 % \vskip 40\p@
2807 \CTEX@setheadingskip \CTEX@chapter@afterskip
2808 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2809 \vskip \CTEX@headingskip
2810 }}

2811 </book|report>

```

14.8.2.3 section 类的标题

```

\@startsection 2812 \def\@startsection#1#2#3#4#5#6{%
2813 \if@noskipsec \leavevmode \fi
2814 \par
2815 % \@tempskipa #4\relax
2816 % \@afterindenttrue
2817 % \ifdim \@tempskipa <\z@
2818 % \@tempskipa -\@tempskipa \@afterindentfalse
2819 % \fi
2820 \CTEX@update@sectionformat@n{#1}%
2821 \ifodd \CTEX@afterindent
2822 \@afterindenttrue

```

```

2823 \else
2824 \afterindentfalse
2825 \fi
2826 \if@nobreak
2827 \everypar{}%
2828 \else
2829 % \addpenalty\@secpenalty\addvspace\@tempskipa
2830 \csname CTEX@#1@break\endcsname
2831 \CTEX@setheadingskip{#4}%
2832 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2833 \addvspace \CTEX@headingskip
2834 \fi
2835 \@ifstar
2836 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2837 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}

\@secntformat 2838 \def\@secntformat#1{%
2839 % \csname the#1\endcsname\quad}%
2840 \csname CTEX@#1name\endcsname
2841 \csname CTEX@#1@aftername\endcsname}

\@sect 2842 \def\@sect#1#2#3#4#5#6[#7]#8{%
2843 \ifnum #2>\c@secnumdepth
2844 \CTEX@ifnamefalse
2845 \CTEX@makeanchor@ssect{#1*}%
2846 \let\@svsec\@empty
2847 \else
2848 \ifodd \csname CTEX@#1@numbering\endcsname
2849 \CTEX@ifnametrue
2850 \refstepcounter{#1}%
2851 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2852 \else
2853 \CTEX@ifnamefalse
2854 \CTEX@makeanchor{#1*}%
2855 \let\@svsec\@empty
2856 \fi
2857 \fi
2858 \CTEX@getttitle{#7}%
2859 % \@tempskipa #5\relax
2860 % \ifdim \@tempskipa>\z@
2861 \unless \ifodd \CTEX@runin
2862 \begingroup
2863 #6{%
2864 \CTEX@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2865 % \interlinepenalty \@M #8\@par}%
2866 \interlinepenalty \@M
2867 \csname CTEX@#1@titleformat\endcsname{#8}%
2868 \csname CTEX@#1@aftertitle\endcsname}%
2869 \endgroup
2870 \csname #1mark\endcsname{#7}%
2871 % \addcontentsline{toc}{#1}{%
2872 % \ifnum #2>\c@secnumdepth \else
2873 % \protect\numberline{\csname the#1\endcsname}%
2874 % \fi
2875 % #7}%
2876 \CTEX@addtocline{#1}{#7}%
2877 \else
2878 \def\@svsechd{%
2879 #6{\hskip\glueexpr #3\relax
2880 % \@svsec #8}%
2881 \@svsec
2882 \csname CTEX@#1@titleformat\endcsname{#8}%
2883 \csname CTEX@#1@aftertitle\endcsname}%
2884 \csname #1mark\endcsname{#7}%
2885 % \addcontentsline{toc}{#1}{%
2886 % \ifnum #2>\c@secnumdepth \else
2887 % \protect\numberline{\csname the#1\endcsname}%

```

```

2888 %      \fi
2889 %      #7}%
2890      \CTEX@addtocline{#1}{#7}}%
2891      \fi
2892      \@xsect{#5}}

\@ssect 2893 \def\@ssect#1#2#3#4#5{%
2894      \CTEX@ifnamefalse
2895      \CTEX@getttitle{#5}%
2896 % \@tempskipa #3\relax
2897 % \ifdim \@tempskipa>\z@
2898      \unless \ifodd \CTEX@runin
2899          \begingroup
2900              #4{%
2901                  \CTEX@hangfrom{\hskip\glueexpr #1\relax}%
2902 %                  \interlinepenalty \@M #5\@par}%
2903                  \interlinepenalty \@M
2904                  \CTEX@titleformat@n{#5}%
2905                  \CTEX@aftertitle}%
2906              \endgroup
2907      \else
2908 %      \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2909      \def\@svsechd{#4{\hskip\glueexpr #1\relax
2910                      \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2911      \fi
2912      \@xsect{#3}}

\@xsect 2913 \def\@xsect#1{%
2914 % \@tempskipa #1\relax
2915 % \ifdim \@tempskipa>\z@
2916      \unless \ifodd \CTEX@runin
2917          \par \nobreak
2918 % \vskip \@tempskipa
2919          \CTEX@setheadingskip{#1}%
2920          \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2921          \vskip \CTEX@headingskip
2922          \@afterheading
2923      \else
2924          \@nobreakfalse
2925          \global\@noskipsecttrue
2926          \everypar{%
2927              \if@noskipsec
2928                  \global\@noskipsecfalse
2929                  {\setbox\z@\lastbox}%
2930                  \clubpenalty\@M
2931                  \begingroup \@svsechd \endgroup
2932                  \unskip
2933 %                  \@tempskipa #1\relax
2934 %                  \hskip -\@tempskipa
2935                  \hskip\glueexpr #1\relax
2936              \else
2937                  \clubpenalty \@clubpenalty
2938                  \everypar{}}%
2939          \fi}%
2940      \fi
2941      \ignorespaces}

2942 <@@=-ctex>

```

\CTEX@hangindent 用于实现 \part 和 \chapter 标题的 indent 和 hang 选项。

```

2943 \cs_new_protected:Npn \CTEX@hangindent #1#2
2944 {
2945     \dim_set:Nn \parindent { \use:c { CTEX@#1@indent } }
2946     \bool_if:cTF { CTEX@#1@hang }
2947     { \@hangfrom { \__ctex_indent_aux: #2 } }
2948     {#2}
2949 }

```

```

2950 \cs_new_protected_nopar:Npn \__ctex_indent_aux:
2951 {
2952   \dim_compare:nNf \parindent = \c_zero_dim
2953   { \skip_horizontal:n { \parindent } }
2954 }

```

\CTEX@hangfrom hang 选项控制是否采用悬挂缩进。

```

2955 \cs_new_protected_nopar:Npn \CTEX@hangfrom
2956 {
2957   \bool_if:NTF \CTEX@hang
2958   { \@hangfrom }
2959   { \noindent \use:n }
2960 }

```

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

```

2961 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2962 {
2963   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2964   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2965   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2966   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2967   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2968   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2969 }
2970 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2971 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2972 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2973 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2974 \cs_new_eq:NN \CTEX@hang \c_true_bool
2975 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

```

\CTEX@part@tocline 2976 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 2977 {
2978   \CTEXifname
2979   { \CTEXthepart \hspace { 1em } }
2980   { }
2981   #2
2982 }
2983 <*book|report>
2984 \cs_new:Npn \CTEX@chapter@tocline #1#2
2985 {
2986   \CTEXifname
2987   { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2988   { }
2989   #2
2990 }
2991 </book|report>

```

```

\CTEXnumberline 2992 \cs_new_nopar:Npn \CTEXnumberline #1
2993 {
2994   \CTEXifname
2995   { \protect \numberline { \use:c { CTEXthe #1 } } }
2996   { }
2997 }

2998 \int_zero:N \l__ctex_tmp_int
2999 \seq_map_inline:Nn \c__ctex_section_headings_seq
3000 {
3001   \int_incr:N \l__ctex_tmp_int
3002   \cs_gset_protected_nopar:cpx {#1}
3003   {
3004     \exp_not:N \@startsection {#1}
3005     { \int_use:N \l__ctex_tmp_int }
3006     { \exp_not:c { CTEX@#1@indent } }
3007     { \exp_not:c { CTEX@#1@beforeskip } }

```

```

3008         { \exp_not:c { CTEX@#1@afterskip } }
3009         { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
3010     }
3011     \cs_new:cpn { CTEX@#1@tocline } ##1##2
3012     { \CTEXnumberline { ##1 } ##2 }
3013 }

```

14.8.2.4 附录标题

```

appendix/name 3014 \keys_define:nn { ctex }
appendix/number 3015 {
appendix/numbering 3016     appendix .meta:nn = { ctex / appendix } {#1} ,
3017     appendix / name .code:n =
3018     { \ctex_assign_heading_name:nn { appendix } {#1} } ,
3019     appendix / number .tl_set:N = \CTEX@appendix@number ,
3020     appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
3021     appendix / numbering .initial:n = true
3022 }
3023 \tl_new:N \CTEX@preappendix
3024 \tl_new:N \CTEX@postappendix

\appendix 3025 \cs_new_eq:NN \CTEX@save@appendix \appendix
3026 \cs_gset_protected_nopar:Npn \appendix
3027 {
3028     \CTEX@save@appendix
3029     <article>
3030     \gdef \CTEX@presection { \CTEX@preappendix }
3031     \gdef \CTEX@thesection { \CTEX@appendix@number }
3032     \gdef \CTEX@postsection { \CTEX@postappendix }
3033     \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
3034     </article>
3035     <book|report>
3036     \gdef \CTEX@prechapter { \CTEX@preappendix }
3037     \gdef \CTEX@thechapter { \CTEX@appendix@number }
3038     \gdef \CTEX@postchapter { \CTEX@postappendix }
3039     \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3040     </book|report>
3041 }

```

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。

```

3042 \cs_new_protected_nopar:Npn \CTEX@makeanchor #1
3043 { }

```

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

```

3044 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
<article> 3045 { part , spart , sect , ssect }
<book|report> 3046 { part , spart , chapter , schapter , sect , ssect }
3047 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3048 {
3049     \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3050     \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3051 }

```

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

```

3052 \cs_new_protected_nopar:Npn \CTEX@hyperheadinghook
3053 {
3054     \group_begin:

```

```

3055 \ifHy@implicit
3056 \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3057 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3058 {
3059   \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3060   {
3061     \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3062     \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3063   }
3064 }
3065 \else:
3066 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3067 { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3068 \fi:
3069 \group_end:
3070 }

3071 \ctex_at_end_package:nn { hyperref }
3072 {
3073   \cs_gset_protected_nopar:Npn \CTEX@makeanchor #1
3074   {
3075     \Hy@MakeCurrentHrefAuto {#1}
3076     \Hy@raisedlink
3077     {
3078       \hyper@anchorstart { \@currentHref }
3079       \hyper@anchorend
3080     }
3081   }
3082   \CTEX@hyperheadinghook
3083 }

```

14.8.2.6 兼容 nameref 宏包

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

```

3084 \cs_new_protected:Npn \CTEX@getttitle #1
3085 { }
3086 \ctex_at_end_package:nn { nameref }
3087 {
3088   \cs_gset_protected_nopar:Npn \CTEX@getttitle { \NR@getttitle }
3089   \seq_map_inline:Nn \c__ctex_headings_cs_seq
3090   {
3091     \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
3092     { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
3093   }
3094 }

```

14.8.2.7 兼容 titlesec 宏包

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

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

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

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

```

3095 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
3096 {
3097   \@ifpackagewith { titlesec } { explicit }
3098   {
3099     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3100     \__ctex_titlesec_format_explicit:Nn
3101   }
3102   { }
3103   \clist_map_inline:nn
3104   { indentafter , noindentafter , indentfirst , nonindentfirst }
3105   {
3106     \@ifpackagewith { titlesec } { ##1 }
3107     {
3108       \clist_map_break:n
3109       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3110     }
3111     { }
3112   }
3113   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3114 }
3115 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
3116 {
3117   \__ctex_titlesec_format:n {#1}
3118   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3119 }
3120 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
3121 {
3122   \cs_if_free:cF { ttlf@#1 }
3123   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3124 }
3125 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
3126 {
3127   \tl_set:Nx #1
3128   {
3129     \bool_if:cTF { CTEX@#2@runin }
3130     { \exp_not:N \ttlh@runin }
3131     { \exp_not:N \ttlh@hang }
3132     \tl_tail:N #1
3133   }
3134 }
3135 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3136 {
3137   \cs_set_nopar:Npx #1 ##1
3138   {
3139     \bool_if:cTF { CTEX@#2@runin }
3140     { \exp_not:N \ttlh@runin }
3141     { \exp_not:N \ttlh@hang }
3142     \exp_args:No \tl_tail:n { #1 { } }
3143   }
3144 }
3145 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
3146 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3147 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3148 {
3149   \exp_not:n { {#1} {#2} {#3} {#4} }
3150   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3151 }

3152 \@ifpackageloaded { titlesec }
3153 { }

```

```

3154 {
3155   \ctex_at_end_package:nn { titlesec }
3156   {
3157     \@ifpackagewith { titlesec } { loadonly }
3158     { }
3159     { \ctex_titlesec_hook: }
3160   }
3161 }

```

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

```

3162 \ctex_at_end_package:nn { titlesec }
3163 {
3164   <*book|report>
3165   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3166   </book|report>
3167   \cs_if_free:NF \ttl@tocpart
3168   {
3169     \cs_set_protected_nopar:Npn \ttl@tocpart
3170     { \tl_set:Nn \ttl@a { \CTEXthechapter \hspace { 1em } } }
3171   }
3172   \seq_map_inline:Nn \c__ctex_headings_seq
3173   {
3174     \cs_if_exist:cF { ttl@toc #1 }
3175     {
3176       \cs_new_protected_nopar:cpx { ttl@toc #1 }
3177       {
3178         \tl_set:Nn \exp_not:N \ttl@a
3179         {
3180           \exp_not:N \protect
3181           \exp_not:N \numberline { \exp_not:c { CTExthe #1 } }
3182         }
3183       }
3184     }
3185   }
3186 }

```

14.8.2.8 兼容 titleps 宏包

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

```

3187 \group_begin:
3188 \char_set_catcode_other:N \#
3189 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
3190 {
3191   \ctex_patch_cmd:Nnn \ttl@settopmark
3192   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3193   {
3194     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3195     \CTEX@titlepslabel@set {#1}
3196   }
3197   \ctex_patch_cmd:Nnn \ttl@setsubmark
3198   { \protect \@namedef { the#1 } { } }
3199   {
3200     \protect \@namedef { the#1 } { }
3201     \CTEX@titlepslabel@clear {#1}
3202   }
3203   \ctex_patch_cmd:Nnn \ttl@setsubmark
3204   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }

```

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

```

3205     {
3206         \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3207         \CTEX@titlepslabel@set {#2}
3208     }
3209 }
3210 \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。

```

3211 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
3212 {
3213     \cs_if_free:cF { CTEXthe#1 }
3214     { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3215 }
3216 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
3217 {
3218     \cs_if_free:cF { CTEXthe#1 }
3219     { \protect \@namedef { CTEXthe#1 } { } }
3220 }

```

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

```

3221 \ctex_at_end_package:nn { titlesec }
3222 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3223 \ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }

```

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

```

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

```

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

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

```

3224 \ctex_at_end_package:nn { titleps }
3225 {
3226     \cs_set_protected_nopar:Npn \ttl@setifthe #1
3227     {
3228         \exp_args:Nco \cs_set_nopar:Npn { ifthe #1 }
3229         {
3230             \CTEXifname
3231             { \protect \@firstoftwo }
3232             { \protect \@secondoftwo }
3233         }
3234     }
3235     \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3236 }

```

14.8.3 目录标签的宽度

```

\numberline 3237 \cs_new_protected:Npn \CTEX@toc@width@n #1
3238 {
3239     \hbox_set:Nn \l__ctex_tmp_box {#1}
3240     \dim_set:Nn \@tempdima
3241     {
3242         \dim_max:nn { \@tempdima }
3243         { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
3244     }
3245 }
3246 \group_begin:
3247 \char_set_catcode_other:N \#
3248 \use:n
3249 {

```

```

3250 \group_end:
3251 \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3252 { \CTEX@toc@width@n {#1} }
3253 { }
3254 { \ctex_patch_failure:N \numberline }
3255 \@ifpackageloaded { tocloft }
3256 { }
3257 {
3258 \ctex_at_end_package:nn { tocloft }
3259 {
3260 \ctex_preto_cmd:NnnTF \numberline
3261 { \char_set_catcode_letter:n { 64 } }
3262 { \CTEX@toc@width@n {#1} }
3263 { }
3264 { \ctex_patch_failure:N \numberline }
3265 }
3266 }
3267 }

```

14.8.4 页眉信息的修改

```

\ps@headings 3268 <*article>
3269 \if@twoside
3270 \ctex_patch_cmd:Nnn \ps@headings
3271 { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3272 { \CTEXifname { \CTEXthesection \quad } { } }
3273 \ctex_patch_cmd:Nnn \ps@headings
3274 { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3275 { \CTEXifname { \CTEXthesubsection \quad } { } }
3276 \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。

```

3277 \ctex_patch_cmd:Nnn \ps@headings
3278 { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3279 { \CTEXifname { \CTEXthesection \quad } { } }
3280 \fi:
3281 </article>
3282 <*book|report>
3283 \ctex_patch_cmd:Nnn \ps@headings
3284 {
3285 \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3286 \ifnum \c@secnumdepth > \m@ne
3287 \@chapapp \ \thechapter . ~ \ %
3288 \fi
3289 \fi \fi
3290 }
3291 { \CTEXifname { \CTEXthechapter \quad } { } }
3292 \if@twoside
3293 \ctex_patch_cmd:Nnn \ps@headings
3294 { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3295 { \CTEXifname { \CTEXthesection \quad } { } }
3296 \fi:
3297 </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。

```

3298 \ctex_at_end_package:nn { fancyhdr }
3299 {
3300 <*article>
3301 \ctex_patch_cmd:Nnn \ps@fancy
3302 { \ifnum \c@secnumdepth > \z@ \thesection \hspace{1em} \relax \fi }
3303 { \CTEXifname { \CTEXthesection \quad } { } }
3304 \ctex_patch_cmd:Nnn \ps@fancy
3305 { \ifnum \c@secnumdepth > \@ne \thesubsection \hspace{1em} \relax \fi }
3306 { \CTEXifname { \CTEXthesubsection \quad } { } }

```

```

3307 </article>
3308 <*book|report>
3309     \ctex_patch_cmd:Nnn \ps@fancy
3310     { \ifnum \c@secnumdepth > \m@ne \chapapp \thechapter . ~ \ \fi }
3311     { \CTEXifname { \CTEXthechapter \quad } { } }
3312     \ctex_patch_cmd:Nnn \ps@fancy
3313     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3314     { \CTEXifname { \CTEXthesection \quad } { } }
3315 </book|report>
3316 }
3317 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

```

3318 <*beamer>
3319 \ExplSyntaxOff

    对应 \partpage。
3320 \defbeamertemplate*{part page}{CTEX}[1][]{%
3321     \begingroup
3322     \CTEX@disableautoindent
3323     % \centering
3324     % {\usebeamerfont{part name}%
3325     % \usebeamercolor{fg}{part name}\partname~\insertromanpartnumber}
3326     % \vskip1em\par
3327     \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3328     \CTEX@part@format
3329     \parindent \dimexpr \CTEX@part@indent \relax
3330     \ifodd \CTEX@part@numbering
3331     \CTEX@partname \CTEX@part@aftername
3332     \fi
3333     \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3334     % \usebeamerfont{part title}\insertpart\par
3335     \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3336     \end{beamercolorbox}%
3337     \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3338     \endgroup
3339 }

```

对应 \sectionpage。

```

3340 \defbeamertemplate*{section page}{CTEX}[1][]{%
3341     \begingroup
3342     \CTEX@disableautoindent
3343     % \centering
3344     % {\usebeamerfont{section name}%
3345     % \usebeamercolor{fg}{section name}\sectionname~\insertsectionnumber}
3346     % \vskip1em\par
3347     \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3348     \CTEX@section@format
3349     \parindent \dimexpr \CTEX@section@indent \relax
3350     \ifodd \CTEX@section@numbering
3351     \CTEX@sectionname \CTEX@section@aftername
3352     \fi
3353     \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3354     % \usebeamerfont{section title}\insertsection\par
3355     \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3356     \end{beamercolorbox}%
3357     \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3358     \endgroup
3359 }

```

对应 \subsectionpage。

```

3360 \defbeamertemplate*{subsection page}{CTEX}[1][]{%
3361     \begingroup

```

```

3362 \CTEX@disableautoindent
3363 % \centering
3364 % {\usebeamerfont{subsection name}%
3365 % \usebeamerfont{fg}[subsection name]\subsectionname~\insertsubsectionnumber}
3366 % \vskip1em\par
3367 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3368 \CTEX@subsection@format
3369 \parindent \dimexpr \CTEX@subsection@indent \relax
3370 \ifodd \CTEX@subsection@numbering
3371 \CTEX@subsectionname \CTEX@subsection@aftername
3372 \fi
3373 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3374 % \usebeamerfont{subsection title}\insertsubsection\par
3375 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3376 \end{beamercolorbox}%
3377 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3378 \endgroup
3379 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3380 \defbeamertemplatealias{part page}{default}{CTEX}
3381 \defbeamertemplatealias{section page}{default}{CTEX}
3382 \defbeamertemplatealias{subsection page}{default}{CTEX}
3383 \ExplSyntaxOn
3384 </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。

```

3385 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel:n #1
3386 {
3387   \protected@edef \@currentlabel
3388   {
3389     \cs_if_exist:ctf { CTEX@the#1 }
3390     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3391     { \exp_not:o { \@currentlabel } }
3392   }
3393 }

```

`\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 C_TE_X 之后被载入, 我们之前的修改将会被覆盖。

```

3394 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3395 {
3396   \seq_map_inline:Nn \c__ctex_headings_seq
3397   { \ctex_fix_varioref_label:n { ##1 } }
3398 }

```

`__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。

```

3399 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3400 {
3401   \tl_if_empty:ct { p@#1 }
3402   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3403 }

```

如果 `varioref` 已经被载入,则使用它来设置。

```

3404 \@ifpackageloaded { varioref }
3405 { \ctex_varioref_hook: }
3406 {
3407   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3408   \RenewDocumentCommand \refstepcounter { m }
3409   {
3410     \CTEX@save@refstepcounter {#1}
3411     \CTEX@setcurrentlabel@n {#1}
3412   }
3413   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3414 }

```

14.8.7 载入 $\langle scheme \rangle$ 文件

```

3415 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3416 </class|heading>

```

14.8.8 标题格式的 `scheme` 定义

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

```

3417 <*scheme&(article|book|report|beamer)>

3418 \keys_set:nn { ctex / part }
3419 {
3420   aftertitle = \par ,
3421   <*article|book|report>
3422     hang = false ,
3423   </article|book|report>
3424   <*plain>
3425     name = \partname \space ,
3426   <*article|book|report>
3427     number = \thepart ,
3428   </article|book|report>
3429   <*beamer>
3430     number = \insertromanpartnumber ,
3431   </beamer>
3432   </plain>
3433   <*chinese>
3434     number = \chinese { part } ,
3435   </chinese>
3436   <*article>
3437     beforeskip = 4ex ,
3438     afterskip = 3ex ,
3439   <*plain>
3440     format = \raggedright ,
3441     nameformat = \Large \bfseries ,
3442     aftername = \par \nobreak ,
3443     titleformat = \huge \bfseries ,
3444     afterindent = false
3445   </plain>
3446   <*chinese>
3447     format = \Large \bfseries \centering ,
3448     aftername = \quad ,
3449     afterindent = true
3450   </chinese>
3451   </article>
3452   <*book|report>
3453     aftername = \par \vskip 20 \p@ ,
3454     beforeskip = 0pt \@plus 1fil ,
3455     afterskip = 0pt \@plus 1fil ,
3456     pagestyle = plain ,
3457     break = \if@openright \cleardoublepage \else \clearpage \fi ,

```

```

3458 <*plain>
3459     format      = \centering ,
3460     nameformat   = \huge \bfseries ,
3461     titleformat  = \Huge \bfseries
3462 </plain>
3463 <*chinese>
3464     format      = \huge \bfseries \centering
3465 </chinese>
3466 </book|report>
3467 <*beamer>
3468     format      = \centering ,
3469     nameformat   = \usebeamerfont { part ~ name }
3470                  \usebeamercolor [fg] { part ~ name } ,
3471     aftername    = \vskip 1em \par ,
3472     titleformat  = \usebeamerfont { part ~ title }
3473 </beamer>
3474 }

3475 <*book|report>
3476 \keys_set:nn { ctex / chapter }
3477 {
3478     pagestyle    = plain ,
3479     aftertitle   = \par ,
3480     hang         = false ,
3481     beforekip    = 50 \p@ ,
3482     afterskip    = 40 \p@ ,
3483     lofskip      = 10 \p@ ,
3484     lotskip      = 10 \p@ ,
3485     break        = \if@openright \cleardoublepage \else \clearpage \fi ,
3486 <*plain>
3487     name         = \chaptername \space ,
3488     number       = \thechapter ,
3489     format       = \raggedright ,
3490     nameformat   = \huge \bfseries ,
3491     aftername    = \par \nobreak \vskip 20 \p@ ,
3492     titleformat  = \Huge \bfseries ,
3493     afterindent  = false ,
3494     tocline      = \CTEXnumberline {#1} #2
3495 </plain>
3496 <*chinese>
3497     number       = \chinese { chapter } ,
3498     format       = \huge \bfseries \centering ,
3499     aftername    = \quad ,
3500     afterindent  = true
3501 </chinese>
3502 }
3503 </book|report>

3504 <@@@=>

3505 \keys_set:nn { ctex / section }
3506 {
3507 <*article|book|report>
3508     number       = \thesection ,
3509     aftername    = \quad ,
3510     aftertitle   = \@@par ,
3511     beforekip    = 3.5ex \@plus 1ex \@minus .2ex ,
3512     afterskip    = 2.3ex \@plus .2ex ,
3513     runin        = false ,
3514     break        = \addpenalty \@secpenalty ,
3515 <*plain>
3516     format       = \Large \bfseries ,
3517     afterindent  = false
3518 </plain>
3519 <*chinese>
3520     format       = \Large \bfseries \centering ,
3521     afterindent  = true
3522 </chinese>

```



```

3523 </article|book|report>
3524 <*beamer>
3525 <*plain>
3526     name          = \sectionname \space ,
3527 </plain>
3528     format         = \centering ,
3529     number         = \insertsectionnumber ,
3530     nameformat     = \usebeamerfont { section ~ name }
3531                   \usebeamercolor [fg] { section ~ name } ,
3532     aftername      = \vskip 1em \par ,
3533     titleformat    = \usebeamerfont { section ~ title } ,
3534     aftertitle     = \par
3535 </beamer>
3536 }

3537 \keys_set:nn { ctex / subsection }
3538 {
3539 <*article|book|report>
3540     number         = \thesubsection ,
3541     format         = \large \bfseries ,
3542     aftername      = \quad ,
3543     aftertitle     = \@@par ,
3544     beforekip      = 3.25ex \@plus 1ex \@minus .2ex ,
3545     afterskip      = 1.5ex \@plus .2ex ,
3546     runin          = false ,
3547     break          = \addpenalty \@secpenalty ,
3548 <*plain>
3549     afterindent    = false
3550 </plain>
3551 <*chinese>
3552     afterindent    = true
3553 </chinese>
3554 </article|book|report>
3555 <*beamer>
3556 <*plain>
3557     name          = \subsectionname \space ,
3558     number        = \insertsubsectionnumber ,
3559 </plain>
3560 <*chinese>
3561     number        = \arabic { section } . \arabic { subsection } ,
3562 </chinese>
3563     format        = \centering ,
3564     nameformat    = \usebeamerfont { subsection ~ name }
3565                   \usebeamercolor [fg] { subsection ~ name } ,
3566     aftername     = \vskip 1em \par ,
3567     titleformat   = \usebeamerfont { subsection ~ title } ,
3568     aftertitle    = \par
3569 </beamer>
3570 }

3571 <*article|book|report>

3572 \keys_set:nn { ctex / subsubsection }
3573 {
3574     number        = \thesubsubsection ,
3575     format        = \normalsize \bfseries ,
3576     aftername     = \quad ,
3577     aftertitle    = \@@par ,
3578     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3579     afterskip     = 1.5ex \@plus .2ex ,
3580     runin        = false ,
3581     break        = \addpenalty \@secpenalty ,
3582 <*plain>
3583     afterindent   = false
3584 </plain>
3585 <*chinese>
3586     afterindent   = true
3587 </chinese>

```

```

3588 }

3589 \keys_set:nn { ctex / paragraph }
3590 {
3591     number      = \theparagraph ,
3592     format      = \normalsize \bfseries ,
3593     aftername    = \quad ,
3594     beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
3595     break       = \addpenalty \@secpenalty ,
3596     <*plain>
3597     afterindent = false
3598 </plain>
3599     <*chinese>
3600     afterindent = true
3601 </chinese>
3602 }

3603 \keys_set:nn { ctex / subparagraph }
3604 {
3605     number      = \thesubparagraph ,
3606     format      = \normalsize \bfseries ,
3607     aftername    = \quad ,
3608     beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
3609     break       = \addpenalty \@secpenalty ,
3610     <*plain>
3611     afterindent = false
3612 </plain>
3613     <*chinese>
3614     afterindent = true
3615 </chinese>
3616 }

```

处理 sub3section 与 sub4section 的格式。

```

3617 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3618 {
3619     \keys_set:nn { ctex / paragraph }
3620     {
3621         aftertitle = \@@par ,
3622         afterskip  = 1ex \@plus .2ex ,
3623         runin      = false
3624     }
3625 }
3626 {
3627     \keys_set:nn { ctex / paragraph }
3628     {
3629         afterskip  = 1em ,
3630         runin      = true
3631     }
3632 }
3633 \int_compare:nNnTF \g__ctex_section_depth_int > 3
3634 {
3635     \keys_set:nn { ctex / subparagraph }
3636     {
3637         aftertitle = \@@par ,
3638         afterskip  = 1ex \@plus .2ex ,
3639         runin      = false
3640     }
3641 }
3642 {
3643     \keys_set:nn { ctex / subparagraph }
3644     {
3645         afterskip  = 1em ,
3646         runin      = true
3647     }
3648 }
3649 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3650 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }

```

```

3651 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3652 <@@=ctex>

```

处理附录的格式。

```

3653 \keys_set:nn { ctex / appendix }
3654 <*article>
3655 { number = \@Alph \c@section }
3656 </article>
3657 <*book|report>
3658 {
3659     name = \appendixname \space ,
3660     number = \@Alph \c@chapter
3661 }
3662 </book|report>
3663 </article|book|report>
3664 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

```

3665 <*ctex|ctexheading>

```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3666 \clist_map_inline:nn { article , book , report , beamer }
3667 {
3668     \@ifclassloaded {#1}
3669     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3670     { }
3671 }

```

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。

```

3672 \msg_new:nnn { ctex } { not-standard-class }
3673 {
3674     None~of~the~standard~document~classes~was~loaded.\\
3675     Heading~`#1'~is~selected.\\
3676     ctex~may~not~work~as~expected.
3677 }
3678 <ctex> \bool_if:NTF \l__ctex_heading_bool
3679 <ctexheading> \use:n
3680 {
3681     \tl_if_exist:NTF \c__ctex_std_class_tl
3682     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3683     {
3684         \cs_if_exist:NTF \chapter
3685         {
3686             \cs_if_exist:NF \if@mainmatter
3687             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3688             \tl_const:Nn \c__ctex_class_tl { book }
3689         }
3690         { \tl_const:Nn \c__ctex_class_tl { article } }
3691         \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3692     }
3693     \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3694 }
3695 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3696 </ctex|ctexheading>

```

14.8.10 标题配置文件

```

3697 <*name>

```

```

3698 \keys_set_known:nn { ctex }
3699 {
3700     contentsname = 目录 ,
3701     listfigurename = 插图 ,
3702     listtablename = 表格 ,
3703     figurename = 图 ,
3704     tablename = 表 ,
3705     abstractname = 摘要 ,
3706     indexname = 索引 ,
3707     bibname = 参考文献 ,
3708     appendixname = 附录 ,
3709     proofname = 证明 ,
3710     algorithmname = 算法 ,
3711     refname = 参考文献 ,
3712     continuation = (续) ,
3713     part / name = { 第 , 部分 } ,
3714     chapter / name = { 第 , 章 }
3715 }
3716 </name>

```

14.9 chinese 方案的其他设置

```

3717 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

<article|book|report> 3718 \pagestyle { headings }

```

日期格式。

```

3719 \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。

```

3720 \ctex_if_autoindent_touched:F
3721 { \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。

```

3722 <*!generic>
3723 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3724 <*beamer>
3725 {
3726     \uselanguage { ChineseGBK }
3727     \languagealias { chinese } { ChineseGBK }
3728     \ctex_file_input:n { ctex-name-gbk.cfg }
3729 }
3730 {
3731     \uselanguage { ChineseUTF8 }
3732     \languagealias { chinese } { ChineseUTF8 }
3733     \ctex_file_input:n { ctex-name-utf8.cfg }
3734 }

```

让 translator 包优先查找中文翻译。

```

3735 \clist_put_left:Nn \trans@languagepath { chinese }
3736 </beamer>
3737 <*!beamer>
3738 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3739 { \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。

```

3740 \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3741 { \int_gset:Nn \g__ctex_font_size_int { 0 } }

```

对 beamer 以外的文档类, 若用户未设置宏包选项 linespread, 则设置行距初始值为 $1.3 \times 1.2 = 1.56$ 倍字体大小。beamer 不调整行距。

```

3742 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }

```

```

3743 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3744 </!beamer>
3745 </!generic>

```

不使用标题定义时的通用设置。

```

3746 <*generic>
3747 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3748 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3749 {
3750   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3751   {
3752     \uselanguage { ChineseGBK }
3753     \languagealias { chinese } { ChineseGBK }
3754     \ctex_file_input:n { ctex-name-gbk.cfg }
3755   }
3756   {
3757     \uselanguage { ChineseUTF8 }
3758     \languagealias { chinese } { ChineseUTF8 }
3759     \ctex_file_input:n { ctex-name-utf8.cfg }
3760   }
3761   \clist_put_left:Nn \trans@languagepath { chinese }
3762 }
3763 {
3764   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3765   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3766   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3767   \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3768   { \int_gset:Nn \g__ctex_font_size_int { 0 } }
3769   \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3770   { \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。

```

3771   \tl_if_exist:NT \c__ctex_std_class_tl
3772   { \RequirePackage { indentfirst } }
3773 }
3774 </generic>
3775 </scheme&chinese>

```

14.10 中文字号

```

3776 <*class|ctex|ctexsize>
\zihao 3777 \NewDocumentCommand \zihao { m }
3778 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

\ctex_zihao:n 3779 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3780 {
3781   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3782   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3783   { \msg_error:nnn { ctex } { fontsize } {#1} }
3784 }
3785 \msg_new:nnnn { ctex } { fontsize }
3786 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3787 {
3788   The~old~font~size~is~used~if~you~continue.\\
3789   The~available~font~sizes~are~listed~as~follow.\\
3790   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3791 }

```

14.10.1 定义中文字号

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

```

3792 \prop_new:N \c__ctex_font_size_prop
3793 \seq_new:N \c__ctex_font_size_seq
3794 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3795 {
3796   \use:x
3797   {
3798     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3799     {
3800       { \dim_to_decimal:n {#2} }
3801       { \dim_to_decimal:n { (#2) * 6 / 5 } }
3802     }
3803   }
3804   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3805 }
3806 \clist_map_inline:nn
3807 {
3808   { 8 } { 5 bp } ,
3809   { 7 } { 5.5 bp } ,
3810   { -6 } { 6.5 bp } ,
3811   { 6 } { 7.5 bp } ,
3812   { -5 } { 9 bp } ,
3813   { 5 } { 10.5 bp } ,
3814   { -4 } { 12 bp } ,
3815   { 4 } { 14 bp } ,
3816   { -3 } { 15 bp } ,
3817   { 3 } { 16 bp } ,
3818   { -2 } { 18 bp } ,
3819   { 2 } { 22 bp } ,
3820   { -1 } { 24 bp } ,
3821   { 1 } { 26 bp } ,
3822   { -0 } { 36 bp } ,
3823   { 0 } { 42 bp }
3824 }
3825 { \__ctex_save_font_size:nn #1 }

```

```

\ctex_declare_math_sizes:nnnn 3826 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3827 {
3828   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3829   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3830 }

```

```

\__ctex_get_font_sizes:Nn 3831 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3832 {
3833   \tl_clear:N #1
3834   \tl_map_inline:nn {#2}
3835   {
3836     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3837     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3838     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3839   }
3840 }

3841 \clist_map_inline:nn
3842 {
3843   { 8 } { 8 } { 5pt } { 5pt } ,
3844   { 7 } { 7 } { 5pt } { 5pt } ,
3845   { -6 } { -6 } { 5pt } { 5pt } ,
3846   { 6 } { 6 } { 5pt } { 5pt } ,
3847   { -5 } { -5 } { 6pt } { 5pt } ,
3848   { 5 } { 5 } { 7pt } { 5pt } ,

```

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

```

3849 { -4 }{ -4 }{ 8pt }{ 6pt } ,
3850 { 4 }{ 4 }{ 5 }{ 6 } ,
3851 { -3 }{ -3 }{ -4 }{ -5 } ,
3852 { 3 }{ 3 }{ 4 }{ 5 } ,
3853 { -2 }{ -2 }{ -3 }{ -4 } ,
3854 { 2 }{ 2 }{ 3 }{ 4 } ,
3855 { -1 }{ -1 }{ -2 }{ -3 } ,
3856 { 1 }{ 1 }{ 2 }{ 3 } ,
3857 { -0 }{ -0 }{ -1 }{ -2 } ,
3858 { 0 }{ 0 }{ 1 }{ 2 }
3859 }
3860 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3861 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3862 {
3863   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3864   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3865   { \msg_error:nnn { ctex } { fontsize } {#2} }
3866 }
3867 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3868 { \cs_set_protected_nopar:Npn #3 { \setfontsize #3 {#1} {#2} #4 } }

3869 \if_case:w \g__ctex_font_size_int
3870 \ctex_file_input:n { ctex-c5size.clo }
3871 \or:
3872 \ctex_file_input:n { ctex-cs4size.clo }
3873 \fi:
3874 </class>ctex<ctexsize>
3875 <*c5size>
3876 \ctex_set_font_size:Nnn \normalsize { 5 }
3877 {
3878   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3879   \abovedisplayshortskip \z@ \@plus3\p@
3880   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3881   \belowdisplayskip \abovedisplayskip
3882   \let\@listi\@listI
3883 }
3884 \ctex_set_font_size:Nnn \small { -5 }
3885 {
3886   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3887   \abovedisplayshortskip \z@ \@plus2\p@
3888   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3889   \def\@listi{\leftmargin\leftmarginI
3890             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3891             \parsep 2\p@ \@plus\p@ \@minus\p@
3892             \itemsep \parsep}
3893   \belowdisplayskip \abovedisplayskip
3894 }
3895 \ctex_set_font_size:Nnn \footnotesize { 6 }
3896 {
3897   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3898   \abovedisplayshortskip \z@ \@plus\p@
3899   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3900   \def\@listi{\leftmargin\leftmarginI
3901             \topsep 3\p@ \@plus\p@ \@minus\p@
3902             \parsep 2\p@ \@plus\p@ \@minus\p@
3903             \itemsep \parsep}
3904   \belowdisplayskip \abovedisplayskip
3905 }
3906 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3907 \ctex_set_font_size:Nnn \tiny { 7 } { }
3908 \ctex_set_font_size:Nnn \large { -4 } { }
3909 \ctex_set_font_size:Nnn \Large { -3 } { }

```

```

3910 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3911 \ctex_set_font_size:Nnn \huge { 2 } { }
3912 \ctex_set_font_size:Nnn \Huge { 1 } { }
3913 </cs5size>
3914 <*cs4size>
3915 \ctex_set_font_size:Nnn \normalsize { -4 }
3916 {
3917   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3918   \abovedisplayshortskip \z@ \@plus3\p@
3919   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3920   \belowdisplayskip \abovedisplayskip
3921   \let\@listi\@listI
3922 }
3923 \ctex_set_font_size:Nnn \small { 5 }
3924 {
3925   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3926   \abovedisplayshortskip \z@ \@plus3\p@
3927   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3928   \def\@listi{\leftmargin\leftmarginI
3929             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3930             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3931             \itemsep \parsep}
3932   \belowdisplayskip \abovedisplayskip
3933 }
3934 \ctex_set_font_size:Nnn \footnotesize { -5 }
3935 {
3936   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3937   \abovedisplayshortskip \z@ \@plus3\p@
3938   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3939   \def\@listi{\leftmargin\leftmarginI
3940             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3941             \parsep 3\p@ \@plus2\p@ \@minus\p@
3942             \itemsep \parsep}
3943   \belowdisplayskip \abovedisplayskip
3944 }
3945 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3946 \ctex_set_font_size:Nnn \tiny { -6 } { }
3947 \ctex_set_font_size:Nnn \large { -3 } { }
3948 \ctex_set_font_size:Nnn \Large { -2 } { }
3949 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3950 \ctex_set_font_size:Nnn \huge { -1 } { }
3951 \ctex_set_font_size:Nnn \Huge { 1 } { }
3952 </cs4size>
<ctexsize> 3953 \normalsize
3954 <*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`。

```

3955 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3956 {
3957   \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。

```

3958   \group_begin: \footnotesize \exp_args:NNNo \group_end:
3959   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3960 }

```

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

```

3961 \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。

```
3962 \NewDocumentCommand \CTeX { }
3963 { C \TeX }
3964 \ctex_at_end_package:nn { hyperref }
3965 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

`captiondelimitter` 过时选项。

```
3966 \keys_define:nn { ctex }
3967 {
3968   captiondelimitter .code:n =
3969   {
3970     \msg_warning:nnn { ctex } { deprecated-option }
3971     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3972   }
3973 }
3974 </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`

```
3975 <*scheme&chinese&(article|book|report)>
3976 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3977 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3978 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3979 </scheme&chinese&(article|book|report)>
3980 <*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。

```
3981 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3982 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3983 \msg_new:nnn { ctex } { fontset-unavailable }
3984 { 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。

```
3985 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3986 {
3987   \tl_if_empty:NTF \g__ctex_fontset_tl
3988   {
3989     \ctex_detected_platform:
3990     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3991   }
3992   {
3993     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3994     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3995     {
3996       \use:x
3997       {
```

```

3998             \ctex_detected_platform:
3999             \msg_error:nnxx { ctex } { fontset-not-found }
4000             { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
4001         }
4002         \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
4003     }
4004 }
4005 }
4006 \@onlypreamble \ctex_load_fontset:
4007 \msg_new:nnnn { ctex } { fontset-not-found }
4008 {
4009     CTex~fontset~`#1'~could~not~be~found.\\
4010     Fontset~`#2'~will~be~used~instead.
4011 }
4012 { You~may~run~`mktexlsr'~firstly. }

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

4013 \keys_define:nn { ctex }
4014 {
4015     fontset .code:n =
4016     {
4017         \ctex_if_preamble:TF
4018         {
4019             \str_if_eq:eeTF {#1} { none }
4020             { \msg_warning:nnn { ctex } { invalid-value } {#1} }
4021             {
4022                 \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
4023                 {
4024                     \tl_gset:Nx \g__ctex_fontset_tl {#1}
4025                     \ctex_load_fontset:
4026                 }
4027                 {
4028                     \msg_error:nnxx { ctex } { fontset-loaded }
4029                     { \g__ctex_fontset_tl } {#1}
4030                 }
4031             }
4032         }
4033         { \msg_error:nn { ctex } { fontset-only-preamble } }
4034     }
4035 }
4036 \msg_new:nnnn { ctex } { fontset-loaded }
4037 {
4038     CTex~fontset~`#1'~has~been~loaded.
4039     \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
4040 }
4041 { Only~one~fontset~can~be~loaded~in~the~preamble. }
4042 \msg_new:nnn { ctex } { fontset-only-preamble }
4043 {
4044     The~`fontset'~option~can~be~used~only~in~preamble.
4045 }

```

载入中文字库。

```

4046 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
4047 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

4048 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
4049 </class|ctex>
4050 <*config>
4051 %%

```

```
4052 </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。

```
4053 <*ctexopts>
4054 %%
4055 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
4056 </ctexopts>
```

14.15 字体定义文件

14.15.1 传统定义方式

```
4057 <*c19|c70>
4058 %%
4059 %% Chinese characters
4060 %%
4061 <c19> %% character set: GBK (extension of GB 2312)
4062 <c70> %% character set: Unicode
4063 %% font encoding: Unicode
4064 %%
4065 </c19|c70>
```

CJK 宏包使用的字体族。

```
<rm|c19> 4066 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm|c70> 4067 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf|c19> 4068 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf|c70> 4069 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt|c19> 4070 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt|c70> 4071 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}
```

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

```
<rm|jy2> 4072 \DeclareKanjiFamily{JY2}{zhrm}{\}
<rm|jt2> 4073 \DeclareKanjiFamily{JT2}{zhrm}{\}
<sf|jy2> 4074 \DeclareKanjiFamily{JY2}{zhshf}{\}
<sf|jt2> 4075 \DeclareKanjiFamily{JT2}{zhshf}{\}
<tt|jy2> 4076 \DeclareKanjiFamily{JY2}{zhtt}{\}
<tt|jt2> 4077 \DeclareKanjiFamily{JT2}{zhtt}{\}

4078 <*rm>
4079 <*c19>
4080 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
4081 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
4082 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
4083 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
4084 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4085 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4086 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
4087 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
4088 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
4089 </c19>
4090 <*c70>
4091 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
4092 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
4093 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
4094 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
4095 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4096 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4097 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
4098 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
4099 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
4100 </c70>
```

```

4101 <*jy2>
4102 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{}
4103 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{}
4104 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{}
4105 </jy2>
4106 <*jt2>
4107 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{}
4108 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{}
4109 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{}
4110 </jt2>
4111 </rm>
4112 <*sf>
4113 <*c19>
4114 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4115 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4116 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4117 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
4118 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4119 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4120 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4121 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4122 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4123 </c19>
4124 <*c70>
4125 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
4126 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
4127 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4128 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4129 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4130 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4131 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4132 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4133 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4134 </c70>
4135 <*jy2>
4136 \DeclareFontShape{JY2}{zhfsf}{m}{n}{<-> upzhsans-h}{}
4137 \DeclareFontShape{JY2}{zhfsf}{bx}{n}{<-> upzhsansb-h}{}
4138 </jy2>
4139 <*jt2>
4140 \DeclareFontShape{JT2}{zhfsf}{m}{n}{<-> upzhsans-v}{}
4141 \DeclareFontShape{JT2}{zhfsf}{bx}{n}{<-> upzhsansb-v}{}
4142 </jt2>
4143 </sf>
4144 <*tt>
4145 <*c19>
4146 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4147 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4148 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4149 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4150 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4151 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4152 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4153 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4154 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4155 </c19>
4156 <*c70>
4157 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4158 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4159 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4160 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4161 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4162 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4163 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4164 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4165 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4166 </c70>

```

```

4167 <*jy2>
4168 \DeclareFontShape{JY2}{zh tt}{m}{n}{<-> upzhmono-h}{}
4169 </jy2>
4170 <*jt2>
4171 \DeclareFontShape{JT2}{zh tt}{m}{n}{<-> upzhmono-v}{}
4172 </jt2>
4173 </tt>
4174 <*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。)

```

4175 <*windows>
4176 \file_if_exist:nTF { C:/bootfont.bin }
4177 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
4178 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
4179 </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。

```

4180 <*windowsnew>
4181 \tl_new:N \l__ctex_msyh_suffix_tl
4182 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
4183 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
4184 {
4185   \file_if_exist:nF { msyh.ttc }
4186   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
4187 }
4188 </windowsnew>
4189 <*windowsnew|windowsold>
4190 \sys_if_engine_pdfTeX:TF
4191 {
4192   \ctex_zhmap_case:nnn
4193   {
4194     \ctex_punct_set:n { windows }
4195     \setCJKmainfont
4196     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
4197 <*windowsold>
4198     \setCJKsansfont { simhei.ttf }
4199     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4200 </windowsold>
4201 <*windowsnew>
4202     \setCJKsansfont
4203     [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4204     \setCJKfamilyfont { zhyahei }
4205     [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4206     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4207     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
4208 </windowsnew>
4209     \setCJKmonofont { simfang.ttf }
4210     \setCJKfamilyfont { zhkai } { simkai.ttf }
4211     \setCJKfamilyfont { zhfs } { simfang.ttf }
4212     \setCJKfamilyfont { zhsong } { simsun.ttc }
4213     \setCJKfamilyfont { zhhei } { simhei.ttf }
4214     \setCJKfamilyfont { zhli } { simli.ttf }
4215     \setCJKfamilyfont { zhyou } { simyou.ttf }
4216     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4217     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4218     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4219     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4220   }

```

```

4221 {
4222   \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
4223   \ctex_punct_set:n { windows }
4224   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4225   \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4226   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4227 }
4228 {
4229   \tl_set:Nn \CJKrmdefault { rm }
4230   \tl_set:Nn \CJKsfdefault { sf }
4231   \tl_set:Nn \CJKttdefault { tt }
4232 }
4233 }
4234 {
4235   \sys_if_engine_uptex:TF
4236   {
4237     \ctex_set_upfonts:nnnnnn
4238     {simsum.ttc} {simhei.ttf} {simkai.ttf}
<windowsnew> 4239     {msyh\l__ctex_msyh_suffix_tl} {msyhbd\l__ctex_msyh_suffix_tl}
<windowsold> 4240     {simhei.ttf} {simhei.ttf}
4241     {simfang.ttf}
4242     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4243     \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4244     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4245     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
<windowsnew> 4246     \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4247     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4248     \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4249     \ctex_set_upfamily:nnn { zhyou } { upschg } {}
4250     \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4251   }
4252   {
4253     <*windowsold>
4254     \setCJKmainfont
4255     [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
4256     \setCJKsansfont { SimHei }
4257     \setCJKmonofont { FangSong_GB2312 }
4258     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
4259     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
4260   </windowsold>
4261   <*windowsnew>
4262     \setCJKmainfont
4263     [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
4264     \setCJKsansfont
4265     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4266     \setCJKmonofont { FangSong }
4267     \setCJKfamilyfont { zhkai } { KaiTi }
4268     \setCJKfamilyfont { zhfs } { FangSong }
4269   </windowsnew>
4270     \setCJKfamilyfont { zhsong } { SimSun }
4271     \setCJKfamilyfont { zhhei } { SimHei }
4272     \setCJKfamilyfont { zhli } { LiSu }
4273     \setCJKfamilyfont { zhyou } { YouYuan }
4274   <*windowsnew>
4275     \setCJKfamilyfont { zhyahei }
4276     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4277   </windowsnew>
4278   }
4279 }
4280 </windowsnew>|<windowsold>

```

14.15.3 ctex-fontset-adobe.def

```

4281 <*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。

```

4282 \sys_if_engine_pdftex:TF
4283 {
4284   \sys_if_output_pdf:TF
4285   { \ctex_fontset_error:n { adobe } }
4286   {
4287     \ctex_zhmap_case:nnn
4288     {
4289       \setCJKmainfont
4290       [
4291         cmap = UniGB-UTF16-H ,
4292         BoldFont = AdobeHeitiStd-Regular.otf ,
4293         ItalicFont = AdobeKaitiStd-Regular.otf
4294       ] { AdobeSongStd-Light.otf }
4295       \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4296       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4297       \setCJKfamilyfont { zhsong }
4298       [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
4299       \setCJKfamilyfont { zhhei }
4300       [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4301       \setCJKfamilyfont { zhkai }
4302       [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
4303       \setCJKfamilyfont { zhfs }
4304       [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4305       \ctex_punct_set:n { adobe }
4306       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4307       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4308       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4309       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4310       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4311     }
4312     {
4313       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
4314       \ctex_punct_set:n { adobe }
4315       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4316       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4317       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4318     }
4319     { \ctex_fontset_error:n { adobe } }
4320   }
4321 }
4322 {
4323   \sys_if_engine_uptex:TF
4324   {
4325     \ctex_set_upfonts:nnnnnn
4326     {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
4327     {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
4328     {AdobeFangsongStd-Regular.otf}
4329     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4330     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4331     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4332     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4333   }
4334   {
4335     \setCJKmainfont
4336     [
4337       BoldFont = AdobeHeitiStd-Regular ,
4338       ItalicFont = AdobeKaitiStd-Regular
4339     ] { AdobeSongStd-Light }
4340     \setCJKsansfont { AdobeHeitiStd-Regular}
4341     \setCJKmonofont { AdobeFangsongStd-Regular}
4342     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4343     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4344     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4345     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4346   }
4347 }

```

4348 </adobe>
 4349 <*fandol>

14.15.4 ctex-fontset-fandol.def

```

4350 \sys_if_engine_pdftex:TF
4351 {
4352   \sys_if_output_pdf:TF
4353   { \ctex_fontset_error:n { fandol } }
4354   {
4355     \ctex_zhmap_case:nnn
4356     {
4357       \setCJKmainfont
4358       [
4359         cmap = UniGB-UTF16-H ,
4360         BoldFont = FandolSong-Bold.otf ,
4361         ItalicFont = FandolKai-Regular.otf
4362       ] { FandolSong-Regular.otf }
4363       \setCJKsansfont
4364       [
4365         cmap = UniGB-UTF16-H ,
4366         BoldFont = FandolHei-Bold.otf
4367       ] { FandolHei-Regular.otf }
4368       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4369       \setCJKfamilyfont { zhsong }
4370       [
4371         cmap = UniGB-UTF16-H ,
4372         BoldFont = FandolSong-Bold.otf
4373       ] { FandolSong-Regular.otf }
4374       \setCJKfamilyfont { zhhei }
4375       [
4376         cmap = UniGB-UTF16-H ,
4377         BoldFont = FandolHei-Bold.otf
4378       ] { FandolHei-Regular.otf }
4379       \setCJKfamilyfont { zhfs }
4380       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4381       \setCJKfamilyfont { zhkai }
4382       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4383       \ctex_punct_set:n { fandol }
4384       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4385       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4386       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4387       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4388       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4389       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4390     }
4391     {
4392       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4393       \ctex_punct_set:n { fandol }
4394       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4395       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4396       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4397     }
4398     { \ctex_fontset_error:n { fandol } }
4399   }
4400 }
4401 {
4402   \sys_if_engine_uptex:TF
4403   {
4404     \ctex_set_upfonts:nnnnnn
4405     {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4406     {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4407     {FandolFang-Regular.otf}
4408     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4409     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4410     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4411     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}

```



```

4412     }
4413     {
4414         \setCJKmainfont
4415         [
4416             Extension = .otf ,
4417             BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular
4418         ]
4419         { FandolSong-Regular }
4420     \setCJKsansfont
4421     [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4422     \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4423     \setCJKfamilyfont { zhsong }
4424     [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4425     \setCJKfamilyfont { zhhei }
4426     [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4427     \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4428     \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4429     }
4430 }
4431 </fandol>

```

14.15.5 ctex-fontset-mac.def 等

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。

```

4432 <*mac>
4433 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
4434 { \ctex_file_input:n { ctex-fontset-macnew.def } }
4435 { \ctex_file_input:n { ctex-fontset-macold.def } }
4436 </mac>

```

macold 的设置参考了 [OS X Mavericks \(10.9\) 预装的主要简体中文字体列表](#)。

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...

```

在 pdfTeX 下生成的 PDF 只有方框²²。华文细黑和华文仿宋的情况类似。这确认为 DVIPDFMx 的 bug²³。

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

由于 Songti SC Light 的字重与 STSong 及 Windows 上的 SimSun 更接近，故默认字重使用 Songti SC Light，而不带后缀的正常字重事实上没有使用。下方关于 zhsong 的字族设

²²<http://www.newsmth.net/bbscon.php?bid=460&id=312640>

²³<https://github.com/clerkma/ptex-ng/blob/master/texk/libdpx/cidtype2.c#L597>

置也有同样的情况。

```

4437 <*macold|macnew>
4438 \sys_if_engine_pdftex:TF
4439 { \ctex_fontset_error:n { mac } }
4440 {
4441   \sys_if_engine_uptex:TF
4442   { \ctex_fontset_error:n { mac } }
4443   {
4444     <*macold>
4445     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4446     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4447   </macold>
4448   <*macnew>
4449   \setCJKmainfont
4450   [
4451     UprightFont = *~Light ,
4452     BoldFont = *~Bold ,
4453     ItalicFont = Kaiti~SC ,
4454     BoldItalicFont = Kaiti~SC~Bold
4455   ] { Songti~SC }
4456   \setCJKsansfont { PingFang~SC }
4457 </macnew>
4458   \setCJKmonofont { STFangsong }
4459 <*macold>
4460   \setCJKfamilyfont { zhsong } { STSong }
4461   \setCJKfamilyfont { zhhei } { STHeiti }
4462 </macold>
4463 <*macnew>
4464   \setCJKfamilyfont { zhsong }
4465   [
4466     UprightFont = *~Light ,
4467     BoldFont = *~Bold ,
4468   ] { Songti~SC }
4469   \setCJKfamilyfont { zhhei } { Heiti~SC }
4470   \setCJKfamilyfont { zhpfs } { PingFang~SC }
4471 </macnew>
4472   \setCJKfamilyfont { zhfs } { STFangsong }
4473 <macold> \setCJKfamilyfont { zhkai } { STKaiti }
4474 <macnew> \setCJKfamilyfont { zhkai } { Kaiti~SC }
4475   }
4476 }
4477 </macold|macnew>

```

14.15.6 ctex-fontset-founder.def

```

4478 <*founder>
4479 \sys_if_engine_pdftex:TF
4480 {
4481   \ctex_zhmap_case:nnn
4482   {
4483     \setCJKmainfont
4484     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4485     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4486     \setCJKmonofont { FZFSK.TTF }
4487     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4488     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4489     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4490     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4491     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4492     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4493     \ctex_punct_set:n { founder }
4494     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4495     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4496     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4497     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }

```

```

4498     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4499     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4500     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4501   }
4502   {
4503     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4504     \ctex_punct_set:n { founder }
4505     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4506     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4507     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4508   }
4509   { \ctex_fontset_error:n { founder } }
4510 }
4511 {
4512   \sys_if_engine_uptex:TF
4513   {
4514     \ctex_set_upfonts:nnnnnn
4515     { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4516     { FZXH1K.TTF } { FZHTK.TTF }
4517     { FZFSK.TTF }
4518     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4519     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4520     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4521     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4522     \ctex_set_upfamily:nnn { zhli } { upschrm } {}
4523     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4524     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4525     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4526   }
4527   {
4528     \setCJKmainfont
4529     [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }
4530     \setCJKsansfont [ BoldFont = FZHei-B01 ] { FZXiHeiI-Z08 }
4531     \setCJKmonofont { FZFangSong-Z02 }
4532     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4533     \setCJKfamilyfont { zhhei } { FZHei-B01 }
4534     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4535     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4536     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4537     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4538   }
4539 }
4540 </founder>
4541 <*ubuntu>

```

14.15.7 ctex-fontset-ubuntu.def

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

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

需要注意的是 uming.ttc 和 ukai.ttc 看起来像有四种字形的样子,但其实只有“令”和“骨”这区区两个字有新字形,其余都取旧字形²⁴。

```

4542 \sys_if_engine_pdftex:TF
4543 {
4544   \ctex_zhmap_case:nnn
4545   {
4546     \setCJKmainfont
4547     [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4548     \setCJKsansfont { wqy-zenhei.ttc }
4549     \setCJKmonofont { uming.ttc }
4550     \setCJKfamilyfont { zhsong } { uming.ttc }
4551     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }

```

²⁴<http://www.freedesktop.org/wiki/Software/CJKUnifonts/Download/>

```

4552     \setCJKfamilyfont { zhkai } { ukai.ttc }
4553     \ctex_punct_set:n { ubuntu }
4554     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4555     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4556     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4557     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4558     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4559 }
4560 {
4561     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4562     \ctex_punct_set:n { ubuntu }
4563     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4564     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4565     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4566 }
4567 { \ctex_fontset_error:n { ubuntu } }
4568 }
4569 {
4570     \sys_if_engine_uptex:TF
4571     {
4572         \ctex_set_upfonts:nnnnnn
4573         { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4574         { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4575         { uming.ttc }
4576         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4577         \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4578         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4579     }
4580     {
4581         \setCJKmainfont
4582         [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4583         \setCJKsansfont { WenQuanYi~Zen~Hei }
4584         \setCJKmonofont { AR~PL~UMing~CN }
4585         \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4586         \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4587         \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4588     }
4589 }
4590 </ubuntu>

```

14.15.8 中文字体命令

```

4591 <!*windows&!mac>
4592 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4593 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
</ubuntu> 4594 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4595 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4596 <*windowsnew|windowsold|founder>
4597 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4598 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4599 </windowsnew|windowsold|founder>
<windowsnew> 4600 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4601 <*macnew>
4602 \NewDocumentCommand \yahei { } { \CJKfamily { zhpj } }
4603 \NewDocumentCommand \pingfang { } { \CJKfamily { zhpj } }
4604 </macnew>
4605 </!windows&!mac>
4606 </fontset>
4607 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4608 \beginingroup\catcode61\catcode48\catcode32=10\relax%

```

```

4609 \catcode 35=6 % #
4610 \catcode123=1 % {
4611 \catcode125=2 % }
4612 \toks0{\endlinechar=\the\endlinechar\relax}%
4613 \toks2{\endlinechar=13 }%
4614 \def\x#1 #2 {%
4615   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4616   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4617 \x 13 5 % carriage return
4618 \x 32 10 % space
4619 \x 35 6 % #
4620 \x 40 12 % (
4621 \x 41 12 % )
4622 \x 45 12 % -
4623 \x 46 12 % .
4624 \x 47 12 % /
4625 \x 58 12 % :
4626 \x 60 12 % <
4627 \x 61 12 % =
4628 \x 64 11 % @
4629 \x 91 12 % [
4630 \x 93 12 % ]
4631 \x 123 1 % {
4632 \x 125 2 % }
4633 \edef\x#1{\endgroup%
4634   \edef\noexpand#1{%
4635     \the\toks0 %
4636     \let\noexpand\noexpand\noexpand#1%
4637     \noexpand\noexpand\noexpand\noexpand\undefined%
4638     \noexpand\noexpand\noexpand\noexpand\endinput}%
4639     \the\toks2}%
4640 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4641 \begingroup\expandafter\endgroup
4642 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4643   \expandafter\ifx\csname ifctexpdf\endcsname\relax
4644     \expandafter\ifx\csname pdfoutput\endcsname\relax
4645       iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4646     \else ifctexpdf\fi
4647 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4648 \begingroup
4649 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4650   \long\def\x#1\ProvidesFile#2[#3]{%
4651     #1%
4652     \immediate\write-1{File: #2 #3}%
4653     \expandafter\xdef\csname ver@#2\endcsname{#3}}
4654   \expandafter\x%
4655 \fi
4656 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4657 <*windows>
4658 \ProvidesFile{zhwindowsfonts.tex}%
4659 [2019/05/29 v2.4.16 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4660
4661 \ifzhmappdf
4662   \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4663   \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4664   \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4665   \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4666   \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4667   \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4668   \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}

```

```

4669
4670 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4671 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4672 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4673 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4674 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4675 \pdfmapline{=unili@Unicode@ <simli.ttf}
4676 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4677
4678 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4679 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4680 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4681 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4682 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4683 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4684
4685 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4686 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4687 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4688 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4689 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4690 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4691
4692 \else
4693 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4694 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4695 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4696 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4697 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4698 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4699 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4700
4701 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4702 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4703 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4704 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4705 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4706 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4707 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4708
4709 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4710 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4711 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4712 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4713 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4714 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4715
4716 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4717 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4718 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4719 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4720 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4721 \special{pdf:mapline uniyouisl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4722
4723 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4724 <*adobe>
4725 \ProvidesFile{zhadobefonts.tex}%
4726 [2019/05/29 v2.4.16 Adobe font map loader for DVIPDFMx (CTEX)]
4727
4728 \ifzhmappdf
4729 %% pdfTeX does not support OTF fonts
4730
4731 \else
4732 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4733 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}

```

```

4734 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4735 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4736 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4737
4738 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4739 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4740 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4741 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4742 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4743
4744 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4745 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4746 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4747 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4748
4749 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4750 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4751 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4752 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4753
4754 </adobe>

```

14.15.9.3 zhfandolfonts.tex

```

4755 <*fandol>
4756 \ProvidesFile{zhfandolfonts.tex}%
4757 [2019/05/29 v2.4.16 Fandol font map loader for DVIPDFMx (CTEX)]
4758
4759 \ifzhmappdf
4760 %% pdfTeX does not support OTF fonts
4761
4762 \else
4763 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4764 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4765 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4766 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4767 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4768
4769 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4770 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4771 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4772 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4773 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4774
4775 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4776 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4777 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4778 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4779
4780 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4781 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4782 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4783 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4784
4785 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4786 <*founder>
4787 \ProvidesFile{zhfounderfonts.tex}%
4788 [2019/05/29 v2.4.16 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4789
4790 \ifzhmappdf
4791 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF>}
4792 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF>}
4793 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF>}
4794 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF>}
4795 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF>}

```



```

4796 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4797 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4798
4799 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4800 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4801 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4802 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4803 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4804 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4805 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4806
4807 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4808 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4809 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4810 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4811 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4812 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4813
4814 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4815 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4816 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4817 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4818 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4819 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4820
4821 \else
4822 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4823 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4824 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4825 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4826 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4827 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4828 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4829
4830 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4831 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4832 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4833 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4834 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4835 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4836 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4837
4838 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4839 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4840 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4841 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4842 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4843 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4844
4845 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4846 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4847 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4848 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4849 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4850 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4851
4852 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4853 <*ubuntu>
4854 \ProvidesFile{zhubuntufonts.tex}%
4855 [2019/05/29 v2.4.16 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4856
4857 \ifzhmappdf
4858 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4859 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4860 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}

```



```

4861 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4862 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4863 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4864
4865 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4866 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4867 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4868 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4869 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4870 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
4871
4872 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4873 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4874 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4875 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4876 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4877
4878 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4879 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4880 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4881 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4882 \pdfmapline{=uniyoussl@Unicode@ <wqy-zenhei.ttc}
4883
4884 \else
4885 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4886 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4887 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4888 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4889 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4890
4891 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4892 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4893 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4894 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4895 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4896
4897 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4898 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4899 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4900 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4901
4902 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4903 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4904 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4905 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4906
4907 </ubuntu>
4908 \fi
4909
4910 \ctex@zhmap@endinput
4911 </zhmap>

```

14.15.10 制作 spa 文件

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

```

4912 <*spa>
4913 <*macro>
4914 \input expl3-generic %
4915 \ExplSyntaxOn
4916 \sys_if_engine_xetex:F
4917 {
4918   \msg_new:nn { ctex } { xetex }
4919   { XeTeX is required to compile this document! }
4920   \msg_fatal:nn { ctex } { xetex }

```

```
4921 }
```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```
4922 \seq_const_from_clist:Nn \c__ctex_punct_seq
4923 {
4924   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4925   "3008 , "300A , "3016 , "3010 ,
4926   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4927   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4928   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4929 }
```

\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。

```
4930 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4931 {
4932   \iow_open:Nn \g__ctex_spa_iow {#1}
4933   \clist_map_inline:nn {#2}
4934     { \__ctex_write_family:nn ##1 }
4935   \iow_close:N \g__ctex_spa_iow
4936 }
4937 \iow_new:N \g__ctex_spa_iow
4938 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

4939 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4940 {
4941   \group_begin:
4942     \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4943     \l__ctex_punct_font
4944     \clist_clear:N \l__ctex_punct_bounds_clist
4945     \seq_map_inline:Nn \c__ctex_punct_seq
4946       { \exp_args:No \__ctex_save_bounds:n { \int_use:N \tex_XeTeXcharglyph:D ##1 } }
4947     \iow_now:Nx \g__ctex_spa_iow
4948     {
4949       \token_to_str:N \ctexspadef {#1}
```

最后这三个逗号对 CJKpunct 来说是必要的。

```
4950       { \l__ctex_punct_bounds_clist , , , }
4951     }
4952   \group_end:
4953 }
4954 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4955 {
4956   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4957     {
4958       \__ctex_calc_bounds:nn { 1 } {#1} ,
4959       \__ctex_calc_bounds:nn { 3 } {#1}
4960     }
4961 }
4962 \clist_new:N \l__ctex_punct_bounds_clist
```

CJKpunct 要求的格式是边界空白宽度与 1 em 的比值的一百倍。

```
4963 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4964 {
4965   \fp_eval:n
4966     {
4967       round
4968       (
4969         \dim_to_decimal_in_unit:nn
4970           { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
4971           { 1 em }
```

```

4972     )
4973   }
4974 }
4975 \ExplSyntaxOff
4976 </macro>

```

下面是 C_T_EX 定义的一些字体。

```

4977 <*make>
4978 \input ctexspamacro %
4979
4980 \MAKESPA {ctexpunct.spa}
4981 {
4982   {adobezhsong}      {AdobeSongStd-Light} ,
4983   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4984   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4985   {adobezhfs}        {AdobeFangsongStd-Regular} ,
4986   {fandolzhsong}     {FandolSong} ,
4987   {fandolzhsongb}    {FandolSong-Bold} ,
4988   {fandolzhhei}      {FandolHei} ,
4989   {fandolzhheib}     {FandolHei-Bold} ,
4990   {fandolzhkai}      {FandolKai} ,
4991   {fandolzhfs}       {FandolFang} ,
4992   {founderzhsong}    {FZShuSong-Z01} ,
4993   {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
4994   {founderzhhei}     {FZHei-B01} ,
4995   {founderzhheil}    {FZXiHeiI-Z08} ,
4996   {founderzhkai}     {FZKai-Z03} ,
4997   {founderzhfs}      {FZFangSong-Z02} ,
4998   {founderzhli}      {FZLiShu-S01} ,
4999   {founderzhyou}     {FZXiYuan-M01} ,
5000   {founderzhyoub}    {FZZhunYuan-M02} ,
5001   {ubuntuzhsong}     {AR PL UMinG CN} ,
5002   {ubuntuzhhei}      {WenQuanYi Zen Hei} ,
5003   {ubuntuzhkai}      {AR PL UKai CN} ,
5004   {windowszhsong}    {SimSun} ,
5005   {windowszhhei}     {SimHei} ,
5006   {windowszhkai}     {KaiTi} ,
5007   {windowszhfs}      {FangSong} ,
5008   {windowszhli}      {LiSu} ,
5009   {windowszhyou}     {YouYuan} ,
5010   {windowszhyawei}    {Microsoft YaHei} ,
5011   {windowszhyaweib}  {Microsoft YaHei Bold}
5012 }
5013
5014 \primitive\end
5015 </make>
5016 </spa>

```

14.16 translator 宏包的中文字典

```

5017 <*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。

```

5018 <*theorem>
5019 \providetranslation{Comments}{评论}
5020 \providetranslation{comments}{评论}
5021 \providetranslation{Comment}{评论}
5022 \providetranslation{comment}{评论}
5023 \providetranslation{Corollaries}{推论}
5024 \providetranslation{corollaries}{推论}
5025 \providetranslation{Corollary}{推论}
5026 \providetranslation{corollary}{推论}
5027 \providetranslation{Definitions}{定义}
5028 \providetranslation{definitions}{定义}

```

```

5029 \providetranslation{Definition}{定义}
5030 \providetranslation{definition}{定义}
5031 \providetranslation{Examples}{例}
5032 \providetranslation{examples}{例}
5033 \providetranslation{Example}{例}
5034 \providetranslation{example}{例}
5035 \providetranslation{Exercises}{练习}
5036 \providetranslation{exercises}{练习}
5037 \providetranslation{Exercise}{练习}
5038 \providetranslation{exercise}{练习}
5039 \providetranslation{Facts}{事实}
5040 \providetranslation{facts}{事实}
5041 \providetranslation{Fact}{事实}
5042 \providetranslation{fact}{事实}
5043 \providetranslation{Key Lemmas}{关键引理}
5044 \providetranslation{key lemmas}{关键引理}
5045 \providetranslation{Key Lemma}{关键引理}
5046 \providetranslation{key lemma}{关键引理}
5047 \providetranslation{Key Observations}{关键观察}
5048 \providetranslation{key observations}{关键观察}
5049 \providetranslation{Key Observation}{关键观察}
5050 \providetranslation{key observation}{关键观察}
5051 \providetranslation{Lemmas}{引理}
5052 \providetranslation{lemmas}{引理}
5053 \providetranslation{Lemma}{引理}
5054 \providetranslation{lemma}{引理}
5055 \providetranslation{Main Theorems}{主要定理}
5056 \providetranslation{main theorems}{主要定理}
5057 \providetranslation{Main Theorem}{主要定理}
5058 \providetranslation{main theorem}{主要定理}
5059 \providetranslation{Observations}{观察}
5060 \providetranslation{observations}{观察}
5061 \providetranslation{Observation}{观察}
5062 \providetranslation{observation}{观察}
5063 \providetranslation{Problems}{问题}
5064 \providetranslation{problems}{问题}
5065 \providetranslation{Problem}{问题}
5066 \providetranslation{problem}{问题}
5067 \providetranslation{Proofs}{证明}
5068 \providetranslation{proofs}{证明}
5069 \providetranslation{Proof}{证明}
5070 \providetranslation{proof}{证明}
5071 \providetranslation{Proof Sketch}{证明提要}
5072 \providetranslation{Proof sketch}{证明提要}
5073 \providetranslation{proof sketch}{证明提要}
5074 \providetranslation{Proof Sketches}{证明提要}
5075 \providetranslation{Proof sketches}{证明提要}
5076 \providetranslation{proof sketches}{证明提要}
5077 \providetranslation{Sketch of Proof}{证明提要}
5078 \providetranslation{Sketch of Proofs}{证明提要}
5079 \providetranslation{Sketch of proof}{证明提要}
5080 \providetranslation{Sketch of proofs}{证明提要}
5081 \providetranslation{sketch of proof}{证明提要}
5082 \providetranslation{sketch of proofs}{证明提要}
5083 \providetranslation{Propositions}{命题}
5084 \providetranslation{propositions}{命题}
5085 \providetranslation{Proposition}{命题}
5086 \providetranslation{proposition}{命题}
5087 \providetranslation{Remarks}{注}
5088 \providetranslation{remarks}{注}
5089 \providetranslation{Remark}{注}
5090 \providetranslation{remark}{注}
5091 \providetranslation{Solutions}{解}
5092 \providetranslation{solutions}{解}
5093 \providetranslation{Solution}{解}
5094 \providetranslation{solution}{解}
5095 \providetranslation{Theorems}{定理}

```

```

5096 \providetranslation{theorems}{定理}
5097 \providetranslation{Theorem}{定理}
5098 \providetranslation{theorem}{定理}
5099 </theorem>
5100 </dict>

```

14.17 ctexcap 宏包

```
5101 <*ctexcap>
```

ctexcap 是过时宏包。

```

5102 \clist_new:N \l__ctex_ctexcap_options_clist
5103 \clist_set:Nx \l__ctex_ctexcap_options_clist
5104 { \exp_not:v { opt@ \@currname . \@currentext } , heading }
5105 \msg_new:nnn { ctexcap } { deprecated }
5106 {
5107   Package~`ctexcap'~is~deprecated.\\
5108   Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
5109   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5110 }
5111 \msg_warning:nnx { ctexcap } { deprecated }
5112 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

5113 \PassOptionsToPackage { heading = true } { ctexcap }
5114 \RequirePackageWithOptions { ctex }
5115 </ctexcap>

```

14.18 ctexhook 宏包

```
5116 <*ctexhook>
```

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

5117 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
5118 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
5119 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
5120 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
5121 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
5122 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5123 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
5124 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5125 \cs_set_nopar:Npx \document
5126 {
5127   \CTEX@document@left@hook
5128   \exp_not:o { \document }
5129   \CTEX@document@right@hook
5130 }
5131 \tl_new:N \g__ctex_end_preamble_hook_tl
5132 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

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。

```

5133 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
5134 {
5135   \@ifpackageloaded {#1}
5136   {#2}
5137   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
5138 }

```

给钩子附加内容。

```

5139 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
5140 {

```

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n

```

```
\ctex_at_end_package:nn
```

```

\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn

```

```

5141 \tl_if_exist:NF #1 { \tl_new:N #1 }
5142 \tl_gput_right:Nn #1 {#2}
5143 }
5144 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

\ctex_package_end_hook:n
\ctex_package_end_hook:o

宏包末尾钩子,只执行一次,用后清除。

```

5145 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
5146 {
5147   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
5148   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
5149 }
5150 \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。

```

5151 \tl_put_left:Nn \@popfilename
5152 {
5153   \cs_if_eq:NNT \@currentx \pkgextension
5154   { \ctex_package_end_hook:o { \@currname } }
5155 }
5156 </ctexhook>

```

14.19 ctexpatch 宏包

```

5157 <*ctexpatch>
5158 \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。

```

5159 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
5160 {
5161   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5162   { once } {#2} { \use_i:nn } { \use_ii:nn }
5163 }

```

\ctex_patch_cmd_all:NnnnTF

替换所有匹配到的文本。

```

5164 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5165 {
5166   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5167   { all } {#2} { \use_i:nn } { \use_ii:nn }
5168 }

```

\ctex_patch_cmd:Nnn

快捷方式,在补丁的时候关闭 L^AT_EX3 语法和设置 @ 为字母类,补丁失败时给出警告。

```

5169 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5170 {
5171   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5172   { once }
5173   {
5174     \ExplSyntaxOff
5175     \char_set_catcode_letter:n { 64 }
5176   }
5177   { }
5178   { \ctex_patch_failure:N #1 }
5179 }
5180 \cs_new_protected:Npn \ctex_patch_failure:N #1
5181 { \msg_warning:nxx { ctex } { patch-failure } { \token_to_str:N #1 } }
5182 \msg_new:nnn { ctex } { patch-failure }
5183 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

\ctex_preto_cmd:NnnTF

在宏的原本定义前面增加钩子。

```

5184 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2

```

```

5185 {
5186   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnnw #1
5187   { left } {#2} { \use_i:nn } { \use_ii:nn }
5188 }

```

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```

5189 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5190 {
5191   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnnw #1
5192   { right } {#2} { \use_i:nn } { \use_ii:nn }
5193 }

```

\ctex_patch_boot:NnnTF 参数记号 # 作为宏的参数被读入时,总是会双写,会影响随后的字符串替换。需要先将它转换为普通符号。

```

5194 \cs_new_protected:Npn \ctex_patch_boot:NnnTF #1#2#3#4#5#6
5195 {
5196   \tl_set:Nn \__ctex_patch_true:w {#5}
5197   \tl_set:Nn \__ctex_patch_false:w {#6}
5198   \group_begin:
5199     \char_set_catcode_other:n { 35 }
5200     \ctex_parse_name:NN #1 #2 {#3} {#4}
5201   }
5202   \tl_new:N \__ctex_patch_true:w
5203   \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_\_ \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。

```

5204 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5205 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5206 \group_begin:
5207 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5208 {
5209   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5210   {
5211     \bool_lazy_or:nnTF
5212     { \cs_if_exist_p:c { ##3 ~ } }
5213     { \cs_if_exist_p:c { #1##3 } }
5214     {
5215       \group_begin:
5216       \use:x
5217       {
5218         \group_end:
5219         \__ctex_parse_name:nNNNnN
5220         { \cs_replacement_spec:N ##2 }
5221         \exp_not:N ##2
5222         \exp_not:c { ##3 ~ }
5223         \exp_not:c { #1##3 }
5224       } {##3} ##1

```

```

5225     }
5226     { ##1##2 }
5227 }
5228 \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5229 {
5230     \exp_args:Nc ##6
5231     {
5232         \str_case:nnTF {##1}
5233         {
5234             { \protect ##3 } { }
5235             { \x@protect ##2 \protect ##3 } { }
5236         }
5237         {
5238             \str_if_eq:eeTF
5239             { \exp_not:n { #1@protected@ ##3 #1##3 } }
5240             {
5241                 \exp_last_unbraced:Nf \__ctex_parse_name:w
5242                 \cs_replacement_spec:N ##3 #3 ~ #2 \q_stop
5243             }
5244             { #1##5 ~ } { ##5 ~ }
5245         }
5246     }
5247     {
5248         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5249         {
5250             { #1@protected@ ##2 ##4 } { }
5251             { #1@ ##4 } { }
5252         }
5253         { #1##5 } { ##5 }
5254     }
5255 }
5256 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5257 }
5258 \use:x
5259 {
5260     \__ctex_tmp:w
5261     { \c_backslash_str }
5262     { \c_left_brace_str }
5263     { \tl_to_str:n { testopt } }
5264 }
5265 \group_end:
5266 \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。

```

5267 \str_new:N \l__ctex_prefix_str
5268 \str_new:N \l__ctex_parameter_str
5269 \str_new:N \l__ctex_replacement_str

```

`\ctex_get_macro_meaning:N`
`__ctex_get_macro_meaning:w`

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

```

5270 \group_begin:
5271 \cs_set_protected:Npn \__ctex_tmp:w #1
5272 {
5273     \prg_new_protected_conditional:Npnn
5274     \ctex_get_macro_meaning:N ##1 { TF }
5275     {
5276         \exp_after:wN \__ctex_get_macro_meaning:w
5277         \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5278     }
5279     \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5280     ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5281     {
5282         \tl_if_empty:nTF { ##4 }
5283         { \prg_return_false: }
5284         {
5285             \str_set:Nn \l__ctex_prefix_str { ##1 }

```



```

5286         \str_set:Nn \l__ctex_parameter_str { ##2 }
5287         \str_set:Nn \l__ctex_replacement_str { ##3 }
5288         \prg_return_true:
5289     }
5290 }
5291 }
5292 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5293 \group_end:

```

\ctex_if_rescanable:NnTF 检查宏是否可以重建。

```

5294 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5295 {
5296     \ctex_get_macro_meaning:NnTF #1
5297     {
5298         \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5299         \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5300     }
5301     {#4}
5302 }
5303 \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。

```

5304 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5305 {
5306     \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
5307     \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
5308     \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5309     \use:x
5310     {
5311         \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5312         \exp_not:o { \l__ctex_parameter_tl }
5313         { \exp_not:o { \l__ctex_replacement_tl } }
5314     }
5315 }
5316 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5317 {
5318     \str_if_empty:NnTF #2
5319     { \tl_clear:N #1 }
5320     { \tl_set_rescan:Nno #1 {#3} {#2} }
5321 }
5322 \tl_new:N \l__ctex_prefix_tl
5323 \tl_new:N \l__ctex_parameter_tl
5324 \tl_new:N \l__ctex_replacement_tl

```

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

```

5325 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5326 {
5327     \group_end:
5328     \ctex_if_rescanable:NnTF #1 {#3}
5329     {
5330         \use:x
5331         {
5332             \__ctex_patch_replace:nnnTF {#2}
5333             { \tl_to_str:n {#4} }
5334             { \tl_to_str:n {#5} }
5335         }
5336         {
5337             \__ctex_patch_rebuild:Nn #1 {#3}
5338             \__ctex_patch_true:w
5339         }
5340         { \__ctex_patch_false:w }
5341     }
5342     { \__ctex_patch_false:w }
5343 }

```

替换前先检查原文本是否存在。

```

5344 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5345 {
5346   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5347   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5348 }

```

在宏的前/后附加钩子。

```

5349 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5350 {
5351   \group_end:
5352   \ctex_get_macro_meaning:NTF #1
5353   {
5354     \str_if_empty:NTF \l__ctex_parameter_str
5355     { \__ctex_hookto_cmd_parameterless:Nnnnw }
5356     { \__ctex_hookto_cmd_parameter:Nnnnw }
5357     #1 {#2} {#3} {#4}
5358   }
5359   { \__ctex_patch_false:w }
5360 }

```

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。

```

5361 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5362 {
5363   \str_if_empty:NF \l__ctex_prefix_str
5364   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5365   \tex_edef:D #1
5366   {
5367     \use:c { __ctex_ #2 _hook_aux:nn }
5368     { \exp_not:o {#1} }
5369     { \exp_not:n {#4} }
5370   }
5371   \__ctex_patch_true:w
5372 }
5373 \cs_generate_variant:Nn \tl_rescan:nn { no }
5374 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5375 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

如果宏有参数,需要在字符串中进行附加,然后再重建。

```

5376 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5377 {
5378   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5379   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5380   {
5381     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5382     \__ctex_patch_rebuild:Nn #1 {#3}
5383     \__ctex_patch_true:w
5384   }
5385   { \__ctex_patch_false:w }
5386 }
5387 </ctexpatch>

```

版本历史

v2.0 (2014/03/06 – 2015/05/06)

General: c5size, cs4size 是过时选项。	38
captiondelimiter 是过时选项。	113
fancyhdr 成为过时选项, 原选项功能总是打开。	41
fntef 成为过时选项, 原选项功能总是打开。	41
hyperref 成为过时选项, 原选项功能总是打开。	41
indent, noindent 是过时选项。	39
nofonts, adobefonts, winfonts 是过时选项。	39
nopunct 是过时选项。	40
nospace 是过时选项。	40
nozhmap 是过时选项。	40
punct 选项可以设置标点格式。	40
ctex 宏包新增 heading 选项。	41
\CTEXindent, \CTEXnoindent 是过时命令。	81
\CTEXsetup, \CTEXoptions 是过时命令。	14, 79
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; \CTEXfilltwosides 是过时环境。	74
标题设置新增 pagestyle 选项。	85
调整 \footnotesep 的大小, 以适合行距的变化。	112
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。	42
将标题汉化功能加入 ctex.sty。	82
将中文字号功能提取到可以独立使用的 ctexsize。	109
解决 etoolbox 与 breqn 关于 \end 的冲突。	113
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	74
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jan 宏包支持 Lua [®] TeX。	49
新增 autoindent 选项。	39
新增 fontset 选项。	39
新增 linespread 选项。	38
新增 linestretch 选项。	79
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。	41
新增 zhmCJK 支持选项。	40
新增 zihao 选项。	38
新增统一设置接口 \ctexset。	78
应用 L [®] TeX3 重新整理代码。	1
中文字号不再采用近似值。	110
自动检测操作系统, 载入对应的字体配置。	71
\CJK@surr: 解决与 \nouppercase 的冲突。	45

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 ctex 宏包后章节标题后第一段无段首缩进的问题。	109
---	-----

v2.1 (2015/05/18 – 2015/06/19)

General: format+, nameformat+ 等带加号的选项, 加号与前面的文字之间可以有可选的空格。	84
nameformat 可以接受章节名字为参数。	84
不依赖 ifpdf 宏包。	37
不再设置 hyperref 宏包的 colorlinks 选项。	72
给 pdf [®] TeX 下的非 UTF8 编码 CJK 字体族加上 CMap。	43

将章节标题设置功能提取到可以独立使用的宏包

ctexheading 中。	1
新的标题格式选项 aftertitle。	84
修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。	103
\ctex_if_macosx:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	72
hyperref: 补充定义 \hypersetup。	41

v2.2 (2015/06/21 – 2015/06/30)

General: before skip 和 after skip 选项的符号不再有特殊意义。	84
before skip、after skip 和 indent 选项支持表达式。	87
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/before skip 和 part/after skip 选项有意义。	87
给 enumitem 宏包注册 \chinese 和 \zhnum。	82
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。	133
将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。	103
删去 etoolbox 与 breqn 的兼容补丁。	113
新的标题格式选项 afterindent。	84
新的标题格式选项 numbering。	84
新的标题格式选项 runin。	84
新增子宏包 ctexpatch 实现给宏打补丁功能。	134

v2.3 (2015/09/17 – 2016/01/05)

General: .value_required: 和 .value_forbidden: 已过时。	38
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。	37
更新 LuaTeX-jan 支持 (20150922.0)。	49
更新 unicode-math 宏包补丁。	50
兼容 titleps 宏包。	98
修复 nameformat 作用域问题。	84
与 L [®] TeX3 (2015/12/20) 同步。	35

v2.4 (2015/02/19 – 2016/04/25)

General: 初步支持 up [®] TeX。	68
加强 beamer 宏包支持。	1
提供 translator 宏包的中文定理名称翻译。	131
提供 up [®] TeX 的 NFSS 字体定义。	115
正确更新 CJK 包的 \CJKfamilydefault。	70
正确设置 up [®] TeX 下字体命令。	69
\ctex_parse_name:NN: 修复宏名解析错误。	135

v2.4.1 (2016/04/26 – 2016/05/14)

General: beamer 不调整默认行距。	108
beamer 不调整默认字体大小。	108
使用 bootfont.bin 判断 Windows XP 以避免权限问题。	117
随字体更新 up [®] TeX 的 \xkanjiskip。	77
新的标题格式选项 part/fixbefore skip 和 chapter/fixbefore skip。	85
正确更新 up [®] TeX 的 \CJKfamilydefault。	70
\ccwd: 正确设置 up [®] TeX 下的 \ccwd。	75
\chinese: 支持 \pagenumbering。	82
zihao: 不允许无参 zihao 选项。	38

v2.4.2	(2016/05/15)	v2.4.10	(2017/07/19 – 2017/07/23)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	50	General: 常数 <code>\c_minus_one</code> 已过时。	35
<code>\em</code> : 兼容 <code>upL^AT_EX 2016/05/07u00</code> 的定义。	69	定义 <code>\cht</code> , <code>\cdp</code> 和 <code>\c wd</code> 。	77
v2.4.3	(2016/06/03 – 2016/08/26)	使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 (<code>L^AT_EX 3 2017/07/19</code>)。	35
General: 更新 <code>unicode-math</code> 补丁。	55	v2.4.11	(2017/08/17 – 2017/11/21)
简化 <code>fontspec</code> 补丁。	50	General: 不把 <code>Enclosed Alphanumerics</code> 设置为 <code>J_Achar</code> 。	49
确保 <code>\proofname</code> 非空。	82	不把希腊和西里尔字母设置为 <code>J_Achar</code> 。	49
删除选项 <code>part/fixbeforeskip</code> 和 <code>chapter/fixbeforeskip</code> 。	85	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
新的标题格式选项 <code>fixskip</code> 。	84	<code>\ps@headings</code> : 补充页眉空格。	100
v2.4.4	(2016/09/09 – 2016/09/19)	v2.4.12	(2017/12/05 – 2018/01/27)
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	48	General: 同步 <code>L^AT_EX 3 2017/12/16</code> 。	1
改进 <code>hyperref</code> 宏包的标题锚点设置。	95	修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。	78
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	44	正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	68
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	98	<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	113
提供 <code>\CTEXifname</code> 。	86	v2.4.13	(2018/03/23)
提供 <code>\partmark</code> 。	86	General: 修正导言区 <code>\selectfont</code> 钩子位置。	76
新的标题格式选项 <code>break</code> 。	84	v2.4.14	(2018/05/01)
v2.4.5	(2016/10/01 – 2016/10/25)	General: 配置 <code>macnew</code> 的默认字体设置。	121
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	85	区分 <code>macold</code> 及 <code>macnew</code> 。	121
新的标题格式选项 <code>hang</code> 。	84	为 <code>macnew</code> 配置字体命令。	124
新的标题格式选项 <code>tocline</code> 。	84	v2.4.15	(2019/01/29 – 2019/04/05)
<code>\ps@headings</code> : 修复补丁失败。	100	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。	87
v2.4.6	(2016/10/31 – 2016/11/20)	将 <code>JY2</code> 和 <code>JT2</code> 编码的字体定义提取到单独的文件中。	115
General: <code>\CTEXifname</code> 初始为假。	86	将 <code>upL^AT_EX</code> 的默认字体由 <code>mc</code> 改为 <code>zhrm</code> , 并启用 <code>\jfam</code> 。	69
支持字体属性可选项在后的新语法。	60	局部指定 <code>autoindent</code> 为 <code>false</code> , 并交换 <code>\CTEX@XXX@indent</code> 与 <code>\CTEX@XXX@format</code> 的顺序。	101
重新初始化 <code>\ifthechapter</code> 等。	99	同步 <code>L^AT_EX 3 2019/03/05</code> 。	1
v2.4.7	(2016/12/23 – 2016/12/27)	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。	122
General: 依赖 <code>pxeverysel</code> 宏包。	70	显式补丁 <code>upL^AT_EX</code> 的 <code>\rmfamily</code> 等字体命令。	68
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	100	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。	87
v2.4.8	(2017/02/23)	v2.4.16	(2019/05/11 – 2019/05/29)
<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: 更好地兼容 <code>nameref</code> 宏包。	96
v2.4.9	(2017/02/27)	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法, 在其标题内部禁用 <code>autoindent</code> 。	87
General: 调整 <code>unicode-math</code> 补丁的代码顺序。	55	允许设置 <code>autoindent</code> 为 0。	37

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	
\	502, 3287, 3294, 3310, 3313
\#	468, 3188, 3247
.../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	21
.../name	15
.../nameformat	17
.../nameformat+	17
.../number	16
.../numberformat	17
.../numberformat+	17
.../titleformat	19
.../titleformat+	19
.../tocline	23
\/	1638
\\	15, 33, 34, 103, 139, 141, 143, 200, 210, 666, 1096, 1178, 2331, 3674, 3675, 3788, 3789, 4009, 4039, 5107, 5108, 5109, 5183
\{	1107, 1108, 5109
\}	1107, 1108, 5109
10pt	9, 361
11pt	9, 361
12pt	9, 361
A	
\abovedisplayshortskip	3879, 3887, 3898, 3918, 3926, 3937
\abovedisplayskip	3878, 3881, 3886, 3893, 3897, 3904, 3917, 3920, 3925, 3932, 3936, 3943
\abstractname	2315
abstractname	12
\addCJKfontfeature	1178, 1222
\addCJKfontfeatures	1195
\addcontentsline	2532, 2580, 2584, 2589, 2620, 2624, 2629, 2726, 2731, 2737, 2743, 2871, 2885
\AddEnumerateCounter	2270, 2272, 2273, 2274
\addpenalty	2829, 3514, 3547, 3581, 3595, 3609
\addtocontents	2520, 2526, 2748, 2749
\addvspace	2521, 2527, 2541, 2544, 2748, 2749, 2829, 2833, 3327, 3337, 3347, 3357, 3367, 3377
adobefonts	30
\algorithmname	2322, 2338
algorithmname	12
AlternateFont	27, 1334
\appendix	3025
appendix/name	24, 3014
appendix/number	24, 3014
appendix/numbering	23, 3014
\appendixname	2317, 3659
appendixname	12
\arabic	3561
\AtBeginDocument	569, 1627
\AtBeginDvi	427
\AtBeginShipoutFirst	429, 1846, 1861
\AtBeginUTFCommand	713
\AtEndOfClass	129
\AtEndOfPackage	130
\AtEndUTFCommand	719
autoindent	10, 179, 2125
B	
\baselinestretch	2180
\begin	3333, 3353, 3373
\begingroup	2862, 2899, 2931, 3321, 3341, 3361, 4608, 4641, 4648
\belowdisplayshortskip	3880, 3888, 3899, 3919, 3927, 3938
\belowdisplayskip	3881, 3893, 3904, 3920, 3932, 3943
\bfdefault	960, 966, 1324, 1331
\bfseries	2598, 2602, 2640, 2644, 2659, 2679, 2778, 2783, 2802, 3441, 3443, 3447, 3460, 3461, 3464, 3490, 3492, 3498, 3516, 3520, 3541, 3575, 3592, 3606
\bibname	2320, 2323, 2331, 2339, 2345, 2346
bibname	12
bool commands:	
\bool_gset_false:N	260, 265
\bool_gset_true:N	255
\bool_if:NTF	70, 1136, 2946, 2957, 3129, 3139, 3150, 3678
\bool_lazy_or:nnTF	5211
\bool_new:N	125, 1173
\bool_set_false:N	1148
\bool_set_true:N	1166
\c_false_bool	2973, 2975
\c_true_bool	2972, 2974
box commands:	
\box_ht:N	3959
\box_new:N	60
\box_wd:N	1932, 3243
C	
c5size	30
cap	30
captiondelimiter	31, 3966

- \catcode 4608, 4609, 4610, 4611, 4615, 4616
- CCT 30
- CCTfont 30
- \ccwd 21, 26, 121, 183, 1928, 1975,
1976, 1977, 2131, 2169, 2188, 2189, 2196, 2223, 2224,
2225, 2228, 2230, 2231, 2242, 2253, 2254, 3976, 3977, 3978
- \cdp 2043
- \centering 2636, 2675, 3323,
3343, 3363, 3447, 3459, 3464, 3468, 3498, 3520, 3528, 3563
- \chapter 2703, 3684
- chapter/beforeskip 32
- chapter/fixbeforeskip 32
- chapter/hang 22
- chapter/lofskip 21, 2461
- chapter/lotskip 21, 2461
- chapter/numbering 15
- chapter/pagestyle 22, 2461
- \chaptermark 2747
- \chaptername 3487
- char commands:
 - \char_set_catcode:nn 73, 506, 507, 542, 543
 - \char_set_catcode_active:N 1359
 - \char_set_catcode_comment:n 501
 - \char_set_catcode_escape:n 502
 - \char_set_catcode_letter:n 67, 1630, 3261, 5175
 - \char_set_catcode_math_superscript:n 537
 - \char_set_catcode_other:N 468, 868, 1342, 3188, 3247
 - \char_set_catcode_other:n 536, 5199
 - \char_set_catcode_space:n 2049
 - \char_value_catcode:n 73, 506, 507, 542, 543
- CharRange 27, 1334
- \Chinese 31, 2266
- \chinese 26, 2264, 2274, 3434, 3497
- \cht 2043
- \CJK 1897
- \CJKbold
4087, 4088, 4098, 4099, 4115, 4116, 4118, 4119, 4121,
4122, 4126, 4127, 4129, 4130, 4132, 4133, 4147, 4148,
4150, 4151, 4153, 4154, 4158, 4159, 4161, 4162, 4164, 4165
- \CJKfamily 516, 520, 524,
528, 1032, 1055, 1270, 1277, 1278, 1279, 1280, 1428,
1672, 4592, 4593, 4594, 4595, 4597, 4598, 4600, 4602, 4603
- \CJKfamilydefault 510,
511, 528, 584, 1257, 1258, 1270, 1280, 1288, 1290,
1294, 1301, 1314, 1649, 1650, 1736, 1740, 1748, 1761, 1763
- \CJKfilltwosides 1905
- \CJKfontspec 1195
- \CJKglue 1931, 1945, 1959, 1972
- \CJKhook 599
- \CJKnormal 4080, 4081, 4082, 4083, 4084, 4085, 4086,
4091, 4092, 4093, 4094, 4095, 4096, 4097, 4114, 4117,
4120, 4125, 4128, 4131, 4146, 4149, 4152, 4157, 4160, 4163
- \CJKpunctmapfamily
632, 633, 634, 635, 636, 637, 641, 642, 643, 644, 645, 646
- \CJKrmdefault 491,
511, 512, 516, 1103, 1114, 1226, 1258, 1259, 1277, 1290,
1293, 1294, 1646, 1649, 1655, 1744, 4216, 4218, 4219,
4224, 4225, 4226, 4229, 4306, 4309, 4310, 4315, 4316,
4317, 4384, 4387, 4388, 4394, 4395, 4396, 4494, 4497,
4498, 4505, 4506, 4507, 4554, 4557, 4558, 4563, 4564, 4565
- \CJKsfdefault 492, 513,
520, 1104, 1115, 1233, 1260, 1278, 1647, 1662, 1745,
4199, 4206, 4207, 4230, 4307, 4385, 4389, 4495, 4499, 4555
- \CJKtilde 585
- \CJKttdefault 493, 514, 524, 1105, 1116, 1239, 1261,
1279, 1648, 1669, 1746, 4217, 4231, 4308, 4386, 4496, 4556
- clearalternatefont 28, 1520
- \cleardoublepage 2555, 2704, 3457, 3485
- \clearpage 2557, 2704, 3457, 3485
- clist commands:
 - \clist_clear:N 981, 4944
 - \clist_concat:NNN 1001
 - \clist_const:Nn 612
 - \clist_gput_right:Nn 377, 389, 399, 401
 - \clist_gset:Nn 1249
 - \clist_if_empty:NTF 1347, 1366, 1606
 - \clist_map_break:n 3108, 3669
 - \clist_map_function:nN 1523, 1525, 1557, 1571
 - \clist_map_inline:Nn 602, 1459
 - \clist_map_inline:nn
362, 624, 1564, 1889, 1915, 3103, 3666, 3806, 3841, 4933
 - \clist_new:N 132, 998, 1250, 1401, 4962, 5102
 - \clist_put_left:Nn 1002, 3735, 3761
 - \clist_put_right:Nn 1165, 1447, 4956
 - \clist_set:Nn 985, 1384, 1385, 5103
 - \clist_use:Nn 5112
- \clubpenalty 2930, 2937
- \contentsname 2310
- contentsname 12
- continuation 12
- cs commands:
 - \cs:w 3402
 - \cs_end: 3402
 - \cs_generate_variant:Nn
93, 444, 456, 945, 1067, 1154, 1174,
1175, 1380, 1417, 1519, 1557, 1565, 5144, 5150, 5266, 5373
 - \cs_gset_eq:NN 256,
261, 266, 476, 763, 767, 1141, 1144, 3061, 3062, 3067, 3092
 - \cs_gset_nopar:Npx 764
 - \cs_gset_protected_nopar:Npn 3026, 3073, 3088
 - \cs_gset_protected_nopar:Npx
802, 1028, 1421, 1515, 3002
 - \cs_if_eq:NNTF 904, 3059, 3091, 5153, 5299, 5379
 - \cs_if_exist:NTF 44, 353, 453, 758, 770, 800,
838, 923, 1420, 1512, 2007, 3174, 3389, 3684, 3686, 5158
 - \cs_if_exist_p:N 5212, 5213
 - \cs_if_exist_use:N 1051, 1052
 - \cs_if_exist_use:NTF 4, 741, 1073, 1534, 5147
 - \cs_if_free:NTF 438, 448, 604, 619,
1138, 1323, 2144, 2158, 2270, 3122, 3167, 3213, 3218, 3222
 - \cs_new:Npn 113, 120,
859, 865, 869, 977, 1054, 2976, 2984, 3011, 3147, 5256, 5374

- \cs_new_eq:NN . . . 6, 7, 8, 9, 20, 23, 107, 108, 109, 127,
128, 354, 445, 589, 650, 779, 804, 880, 911, 1222, 1229,
1757, 1910, 1911, 1912, 1924, 1972, 1983, 1989, 2265,
2266, 2267, 2506, 2509, 2970, 2971, 2972, 2973, 2974,
2975, 3025, 3049, 3050, 3407, 3682, 3687, 4938, 5303, 5375
- \cs_new_nopar:Npn 819, 1046, 1099, 1110, 2264, 2443, 2992, 3211, 3216, 4963
- \cs_new_protected:Npn 62, 111, 913, 978,
1181, 1345, 1436, 1450, 1601, 1978, 2454, 2531, 2943,
3084, 3237, 3861, 3867, 4939, 5117, 5119, 5133, 5139,
5159, 5164, 5169, 5180, 5184, 5189, 5194, 5204, 5209,
5228, 5279, 5294, 5304, 5316, 5325, 5344, 5349, 5361, 5376
- \cs_new_protected_nopar:Npn 77, 79, 129,
130, 425, 436, 446, 451, 457, 489, 497, 532, 549, 591,
597, 600, 617, 622, 630, 639, 648, 739, 748, 761, 766,
768, 780, 794, 821, 836, 849, 873, 882, 899, 938, 955,
999, 1007, 1026, 1047, 1057, 1082, 1120, 1132, 1134,
1156, 1158, 1267, 1273, 1284, 1309, 1318, 1364, 1375,
1402, 1411, 1418, 1424, 1444, 1457, 1476, 1490, 1497,
1509, 1529, 1545, 1563, 1566, 1568, 1581, 1593, 1635,
1640, 1642, 1696, 1709, 1723, 1734, 1766, 1784, 1799,
1815, 1826, 1830, 1867, 1874, 1878, 1901, 1907, 1928,
1942, 1974, 1990, 2022, 2067, 2143, 2145, 2170, 2205,
2215, 2220, 2373, 2403, 2490, 2497, 2504, 2510, 2512,
2515, 2533, 2950, 2955, 2961, 3042, 3052, 3095, 3115,
3120, 3125, 3135, 3145, 3189, 3385, 3394, 3399, 3779,
3794, 3826, 3831, 3981, 3985, 4930, 4954, 5121, 5123, 5145
- \cs_new_protected_nopar:Npx . . . 461, 1205, 1894, 3176
- \cs_replacement_spec:N 5220, 5242
- \cs_set:Npn 1343, 1986, 2430
- \cs_set_eq:NN 110, 184, 189, 194, 529,
547, 567, 594, 598, 606, 620, 773, 774, 881, 912, 930,
975, 1032, 1033, 1420, 1428, 1438, 1439, 1440, 1512,
1513, 1514, 1638, 1739, 2281, 2286, 2289, 2294, 2511,
2513, 2963, 2964, 2965, 2966, 2967, 2968, 3056, 3099, 3109
- \cs_set_nopar:Npn 1695, 3228
- \cs_set_nopar:Npx 1925, 3137, 5125
- \cs_set_protected:Npn 720, 916, 5207, 5271
- \cs_set_protected_nopar:Npn
. 593, 931, 1945, 3169, 3226, 3868
- \cs_set_protected_nopar:Npx 1980
- \cs_to_str:N 54, 1204, 5205
- \cs_undefine:N . 1011, 1012, 1016, 1017, 1441, 1994, 5148
- cs4size 30
- \csname 2830, 2839, 2840, 2841, 2848, 2867, 2868, 2870, 2873,
2882, 2883, 2884, 2887, 4640, 4642, 4643, 4644, 4649, 4653
- \CTeX 26, 3962
- ctex commands:
 - \ctex_add_cmap:n 441, 446
 - \ctex_add_cmap:Nn 449, 451, 456
 - \ctex_add_to_selectfont:n 1978, 2013, 2041, 2069, 2157
 - \ctex_after_end_preamble:n 37, 110, 5117
 - \ctex_appto_cmd:NnnTF 472, 1629, 5189
 - \ctex_assign_heading_name:nn 2409, 2451, 3018
 - \ctex_at_end:n 129, 288, 351, 1972, 4048
 - \ctex_at_end_package:nn 134,
428, 475, 728, 921, 1840, 1842, 1860, 2268, 3071, 3086,
3155, 3162, 3221, 3223, 3224, 3258, 3298, 3413, 3964, 5133
 - \ctex_at_end_preamble:n 356, 574, 581, 1283, 1683, 5117
 - \ctex_auto_ignorespaces: 588, 598, 2079
 - \ctex_CJK_input:n 532
 - \ctex_declare_math_sizes:nnnn 3826, 3860
 - \ctex_detected_platform: 1765, 3989, 3998
 - \ctex_family_cmap:nn 436, 445
 - \ctex_file_input:n 77, 392, 651, 2108,
3693, 3728, 3733, 3738, 3739, 3754, 3759, 3765, 3766,
3870, 3872, 3990, 3994, 4002, 4048, 4177, 4178, 4434, 4435
 - \ctex_file_wrapper:nnn 62, 78, 81, 499, 534
 - \ctex_fix_varioref_label:n 3397, 3399
 - \ctex_fontset_error:n
. 3981, 4285, 4319, 4353, 4398, 4439, 4442, 4509, 4567
 - \ctex_gadd_hook:Nn 599, 5137, 5139
 - \ctex_get_macro_meaning:N 5274
 - \ctex_get_macro_meaning:NTF 5270, 5296, 5352
 - \ctex_hypersetup:n
. 72, 354, 1823, 1835, 1838, 1853, 1856, 1859
 - \ctex_if_autoindent_touched:TF 127, 184, 189, 194, 3720
 - \ctex_if_ccglue_touched:TF 1956, 2176
 - \ctex_if_ccglue_touched_p: 1956
 - \ctex_if_macosx:TF 1806, 1807, 1809, 1815
 - \ctex_if_preamble:TF 109, 4017
 - \ctex_if_rescanable:NnTF 5294, 5328
 - \ctex_ignorespaces_case:N 591, 2076, 2079, 2082
 - \ctex_load_fontset: 3985, 4025, 4047
 - \ctex_load_zhmap:nnnn 489, 4222, 4313, 4392, 4503, 4561
 - \ctex_ljt_add_font_features:n 1156, 1219
 - \ctex_ljt_add_font_features:nn 1156, 1157
 - \ctex_ljt_char_range_key:nn 1598, 1601
 - \ctex_ljt_clear_alternate_font:n 1523, 1529
 - \ctex_ljt_declare_alternate_shape:nnnnnn 1430, 1450
 - \ctex_ljt_declare_char_range:n 1560, 1563, 1565
 - \ctex_ljt_declare_char_range:nn 1567, 1568
 - \ctex_ljt_def_char_range_key:n 1573, 1593
 - \ctex_ljt_ensure_default_family: 1284, 1753
 - \ctex_ljt_extract_font: 773, 780
 - \ctex_ljt_family_if_exist:nNTF
. 1059, 1068, 1288, 1293, 1311, 1314, 1532
 - \l_ctex_ljt_family_tl
1061, 1066, 1123, 1124, 1139, 1143, 1146, 1157, 1526, 1527
 - \ctex_ljt_fontspec:nn 1120, 1132, 1167, 1214
 - \ctex_ljt_get_and_define_fonts:nN 873
 - \ctex_ljt_get_and_define_fonts_al:nN 877, 880
 - \ctex_ljt_get_and_define_fonts_ja:nN 876, 882
 - \ctex_ljt_if_alternate_shape_exist:nTF
. 783, 796, 814, 823
 - \ctex_ljt_if_jfont:nTF 55, 859, 870, 875
 - \ctex_ljt_if_jfont_math:NTF 865, 906
 - \ctex_ljt_math_group_hook: 54, 55, 905, 911, 931
 - \ctex_ljt_pickup_font: . . . 754, 768, 843, 885, 887, 889
 - \ctex_ljt_reset_alternate_font:n 1525, 1545
 - \ctex_ljt_save_alternate_family:Nnnn 1395, 1415, 1418

- `\ctex_ltj_save_alterdate_seq:Nn` 1370, 1375
`\ctex_ltj_save_alterdate_seq:Nnnwn` 1375
`\ctex_ltj_save_alterdate_seq:Nnnwnw` 1378, 1381
`\ctex_ltj_save_char_range:n` 1571, 1578
`\ctex_ltj_save_char_range:nn` 1580, 1581
`\ctex_ltj_select_alterdate_font:` 821, 2016
`\ctex_ltj_select_font:` 739, 2015
`\ctex_ltj_set_alterdate_family:nn` 1373, 1402
`\ctex_ltj_set_alterdate_family:nnn` . 1414, 1422, 1424
`\ctex_ltj_set_alterdate_family:Nnnnn` 1406, 1411, 1417
`\ctex_ltj_set_alterdate_family:nnTF` 986, 1364
`\ctex_ltj_set_alterdate_seq:n` 1336, 1341
`\ctex_ltj_set_alterdate_shape:n` ... 1478, 1490, 1501
`\ctex_ltj_set_alterdate_shape:nnN` 1463, 1497
`\ctex_ltj_set_alterdate_shape:nnn` 1466, 1473
`\ctex_ltj_set_alterdate_shape:nnnn` 1475, 1476
`\ctex_ltj_set_alterdate_shape:Nnnnnnn` . 1453, 1457
`\ctex_ltj_set_family:nnn`
..... 978, 1151, 1198, 1208, 1226, 1233, 1239, 1245
`\ctex_ltj_set_kanjiskip:N` 1640, 1950
`\ctex_ltj_set_math_letter:NN` 913, 925, 933
`\ctex_ltj_set_xkanjiskip:N` 1640, 2029
`\ctex_ltj_subst_font:` 774, 794
`\ctex_ltj_swap_cs:NN` 1429, 1436
`\ctex_ltj_switch_family:n`
..... 1056, 1057, 1067, 1124, 1152, 1206
`\ctex_ltj_update_mathfont:` 1306, 1309
`\ctex_ltj_update_mathfont:n` 1312, 1315, 1318
`\ctex_ltj_use_math_group:Nn` 899, 930
`\ctex_make_spa:nn` 4930
`\ctex_mono_jfm:n` 938, 946, 2099
`\ctex_package_end_hook:n` 5145, 5154
`\ctex_parse_name:NN` 1997, 5200, 5204
`\ctex_parse_name:NNn` 5205, 5209, 5266
`\ctex_patch_boot:NnnTF`
..... 5161, 5166, 5171, 5186, 5191, 5194
`\ctex_patch_cmd:Nnn`
..... 1691, 3191, 3197, 3203, 3270, 3273, 3277,
3283, 3293, 3301, 3304, 3309, 3312, 3976, 3977, 3978, 5169
`\ctex_patch_cmd_all:NnnTF` 5164
`\ctex_patch_cmd_once:NnnTF` 1685, 1999, 5159
`\ctex_patch_failure:N` 478,
518, 522, 526, 530, 1633, 2004, 3254, 3264, 5178, 5180
`\ctex_plane_to_utfxvibe:Nn` 548
`\ctex_preto_cmd:NnnTF` 515, 519, 523, 527, 3251, 3260, 5184
`\ctex_punct_map_bfseries:nn`
..... 622, 4207, 4219, 4225, 4310,
4316, 4388, 4389, 4395, 4498, 4499, 4500, 4506, 4558, 4564
`\ctex_punct_map_family:nn` 617, 4199, 4206,
4216, 4217, 4224, 4306, 4307, 4308, 4315, 4384, 4385,
4386, 4394, 4494, 4495, 4496, 4505, 4554, 4555, 4556, 4563
`\ctex_punct_map_itshape:nn` 639,
4218, 4226, 4309, 4317, 4387, 4396, 4497, 4507, 4557, 4565
`\ctex_punct_map_series:nnn` 626, 627, 630
`\ctex_punct_set:n` 600,
4194, 4223, 4305, 4314, 4383, 4393, 4493, 4504, 4553, 4562
`\ctex_punct_space:nn` 648
`\ctex_scheme_input:n` 79, 3415, 3695
`\ctex_select_size:` .. 2132, 2139, 2142, 2143, 2164, 2218
`\ctex_set_default_ccwd:Nn` 111, 193, 2138, 2163
`\ctex_set_font_size:Nnn` 3861, 3876,
3884, 3895, 3906, 3907, 3908, 3909, 3910, 3911, 3912,
3915, 3923, 3934, 3945, 3946, 3947, 3948, 3949, 3950, 3951
`\ctex_set_ignorespaces:` 591, 599
`\ctex_set_upfamily:nnn`
..... 1696, 4242, 4243, 4244, 4245, 4246,
4247, 4249, 4329, 4330, 4331, 4332, 4408, 4409, 4410,
4411, 4518, 4519, 4520, 4521, 4522, 4524, 4576, 4577, 4578
`\ctex_set_upfonts:nnnnnn`
..... 1723, 4237, 4325, 4404, 4514, 4572
`\ctex_set_upmap:nnn`
... 1709, 1725, 1726, 1727, 1728, 4248, 4250, 4523, 4525
`\ctex_set_zhmap:n` 424, 494, 1711
`\ctex_titleps_hook:` 3187, 3222, 3223
`\ctex_titlesec_hook:` 3095, 3159
`\ctex_update_ccglue:` 1942, 2183, 2202, 2246
`\ctex_update_ccwd:` 1928, 2177, 2254
`\ctex_update_default_family:` 47, 574, 1283, 1683, 1733
`\ctex_update_em_unit:` 1974, 2172, 2222
`\ctex_update_kanjisize:` 2067, 2069
`\ctex_update_parindent:` 2150, 2205, 2247
`\ctex_update_size:` 2144, 2145
`\ctex_update_stretch:` 2149, 2170
`\ctex_update_xkanjiskip:` 2021, 2041
`\ctex_update_ziju:` 2152, 2220
`\ctex_varioref_hook:` 3394, 3405, 3413
`\ctex_zhmap_case:nnn`
..... 128, 256, 261, 266, 4192, 4287, 4355, 4481, 4544
`\ctex_zhmap_input:n` 494, 497
`\ctex_zihao:n` 3778, 3779
`\ctex_ziju:n` 2214, 2215
ctex internal commands:
`\g__ctex_after_end_preamble_hook_tl` 5120, 5124, 5132
`__ctex_assign_heading_name:nnn` 2451
`\l__ctex_autoindent_tl`
... 126, 183, 188, 193, 2131, 2135, 2138, 2207, 2210, 2534
`__ctex_calc_bounds:nn` 4958, 4959, 4963
`__ctex_ccglue:` 1959, 1972
`\l__ctex_ccglue_skip` 1946,
1950, 1953, 1955, 1963, 1967, 2181, 2192, 2201, 2237, 2245
`\c__ctex_class_tl`
... 84, 86, 404, 409, 414, 419, 3682, 3688, 3690, 3691, 3693
`__ctex_clear_fntef_color:n` ... 1867, 1874, 1878, 1893
`\c__ctex_cmap_encoding_seq` 434, 440
`\l__ctex_ctexcap_options_clist` 5102, 5103, 5112
`__ctex_def_heading_keys:n` 2403, 2481
`__ctex_default_ccwd_aux:n` 112, 113
`__ctex_default_ccwd_aux:w` 116, 120
`\l__ctex_encoding_tl` 122, 216,
217, 482, 485, 576, 584, 1836, 2105, 2262, 3723, 3750, 3764
`\g__ctex_end_preamble_hook_tl` 5118, 5122, 5131
`\c__ctex_engine_file_str` 43, 2108

`\c_ctex_engine_str` [43](#)
`\l_ctex_family_default_init_tl` [1736](#), [1756](#)
`_ctex_family_default_wrap:n` [1739](#), [1757](#), [1760](#)
`_ctex_fix_varioref_label:n` [3399](#)
`\g_ctex_font_size_int` [145](#),
[156](#), [157](#), [158](#), [372](#), [376](#), [398](#), [3740](#), [3741](#), [3767](#), [3768](#), [3869](#)
`\c_ctex_font_size_prop` [3781](#), [3792](#), [3836](#), [3863](#)
`\c_ctex_font_size_seq` [3790](#), [3793](#), [3804](#)
`\l_ctex_font_size_tl` [3781](#), [3782](#), [3828](#), [3829](#), [3863](#), [3864](#)
`\g_ctex_fontset_tl` [220](#), [1768](#), [1786](#), [1811](#), [1818](#), [1819](#),
[3987](#), [3990](#), [3993](#), [3994](#), [4000](#), [4002](#), [4022](#), [4024](#), [4029](#), [4046](#)
`_ctex_get_font_sizes:Nn` [3828](#), [3831](#)
`_ctex_get_macro_meaning:w` [5270](#)
`\l_ctex_heading_bool` [299](#), [3678](#)
`\l_ctex_heading_skip` [2494](#),
[2501](#), [2503](#), [2505](#), [2506](#), [2517](#), [2518](#), [2521](#), [2523](#), [2524](#), [2527](#)
`\c_ctex_headings_cs_seq` [3044](#), [3057](#), [3066](#), [3089](#)
`\c_ctex_headings_seq` ... [2363](#), [2478](#), [3172](#), [3235](#), [3396](#)
`_ctex_hookto_cmd:Nnnnw` [5186](#), [5191](#), [5349](#)
`_ctex_hookto_cmd_parameter:Nnnnw` [5356](#), [5376](#)
`_ctex_hookto_cmd_parameterless:Nnnnw` . [5355](#), [5361](#)
`_ctex_indent_aux:` [2947](#), [2950](#)
`_ctex_initial_heading:n` [2373](#), [2480](#)
`\g_ctex_kanjisize_func` [2047](#), [2057](#), [2068](#)
`_ctex_left_hook_aux:nn` [5374](#)
`\l_ctex_line_spread_fp`
..... [112](#), [176](#), [3742](#), [3743](#), [3769](#), [3770](#), [3955](#), [3957](#)
`\l_ctex_line_stretch_tl` [2163](#), [2168](#), [2169](#), [2173](#)
`\l_ctex_ltj_add_alternate_bool` [1136](#), [1148](#), [1166](#), [1173](#)
`_ctex_ltj_alternate_cs:n`
..... [1012](#), [1015](#), [1016](#), [1017](#), [1051](#), [1052](#), [1054](#), [1139](#),
[1142](#), [1143](#), [1145](#), [1146](#), [1370](#), [1407](#), [1469](#), [1494](#), [1534](#), [1551](#)
`\l_ctex_ltj_alternate_family_tl` [1432](#), [1435](#)
`\l_ctex_ltj_alternate_options_clist`
..... [1385](#), [1386](#), [1397](#), [1401](#)
`\l_ctex_ltj_alternate_seq`
..... [982](#), [1354](#), [1355](#), [1363](#), [1369](#), [1377](#), [1625](#)
`\l_ctex_ltj_base_CJKfamily_tl`
..... [983](#), [997](#), [1049](#), [1469](#), [1494](#)
`\l_ctex_ltj_base_family_tl` [65](#),
[1013](#), [1044](#), [1050](#), [1413](#), [1454](#), [1513](#), [1532](#), [1537](#), [1538](#), [1550](#)
`_ctex_ltj_change_encoding:` [947](#), [1031](#), [1427](#)
`\l_ctex_ltj_char_range_clist` [981](#),
[1338](#), [1347](#), [1366](#), [1384](#), [1396](#), [1408](#), [1431](#), [1453](#), [1604](#), [1606](#)
`_ctex_ltj_char_range_parse_feature:w` . [1609](#), [1613](#)
`\g_ctex_ltj_char_range_prop` [1461](#), [1568](#)
`\l_ctex_ltj_char_range_tl`
..... [1461](#), [1464](#), [1570](#), [1572](#), [1574](#), [1576](#), [1583](#)
`_ctex_ltj_check_family:n` [1007](#), [1368](#)
`\l_ctex_ltj_current_font_tl`
..... [51](#), [741](#), [745](#), [746](#), [758](#), [800](#), [805](#)
`\l_ctex_ltj_current_shape_tl` [823](#), [828](#), [833](#), [834](#)
`_ctex_ltj_declare_char_range:nn` [1564](#), [1566](#)
`\g_ctex_ltj_default_features_clist` [1001](#), [1249](#), [1250](#)
`_ctex_ltj_family_csname:n` ... [1011](#), [1028](#), [1046](#), [1073](#)
`\g_ctex_ltj_family_font_name_prop`
..... [988](#), [1004](#), [1009](#), [1084](#), [1160](#), [1286](#), [1298](#)
`\g_ctex_ltj_family_font_options_prop` [989](#), [1004](#), [1163](#)
`\g_ctex_ltj_family_int` . [1126](#), [1128](#), [1446](#), [1447](#), [1449](#)
`\g_ctex_ltj_family_name_prop`
..... [1004](#), [1013](#), [1036](#), [1070](#), [1413](#)
`_ctex_ltj_family_unknown_warning:n` [1064](#), [1082](#), [1542](#)
`\l_ctex_ltj_font_options_clist` [985](#),
[990](#), [991](#), [992](#), [998](#), [1035](#), [1164](#), [1165](#), [1168](#), [1404](#), [1405](#), [1409](#)
`_ctex_ltj_fontspec:nnn` [1127](#), [1134](#), [1154](#)
`\g_ctex_ltj_fontspec_family_tl`
..... [1034](#), [1037](#), [1038](#), [1039](#), [1045](#), [1050](#), [1075](#)
`\g_ctex_ltj_fontspec_prop` [1122](#), [1150](#), [1155](#)
`_ctex_ltj_gset_family_cs:nn` [993](#), [1026](#)
`_ctex_ltj_if_jfont_math:w` [866](#), [869](#)
`\l_ctex_ltj_jfm_tl` [938](#), [959](#), [961](#), [1002](#)
`\c_ctex_ltj_math_fam_int` [967](#), [968](#)
`\c_ctex_ltj_math_family_tl`
..... [1320](#), [1322](#), [1324](#), [1327](#), [1331](#)
`\c_ctex_ltj_math_tl`
..... [962](#), [963](#), [965](#), [967](#), [1245](#), [1311](#), [1321](#), [1326](#), [1330](#)
`_ctex_ltj_msg_def_family_map:n` [1097](#), [1099](#)
`_ctex_ltj_msg_family_map:n` [1025](#), [1096](#), [1110](#)
`_ctex_ltj_noembed_wrap:n` [975](#), [977](#)
`_ctex_ltj_pass_args:nnnn`
..... [1181](#), [1197](#), [1207](#), [1213](#), [1225](#), [1232](#), [1238](#), [1244](#)
`_ctex_ltj_patch_external_font:w` [786](#), [819](#)
`_ctex_ltj_pop_fontname:` [757](#), [766](#), [845](#)
`_ctex_ltj_post_arg:w` [1184](#), [1190](#)
`_ctex_ltj_push_fontname:n` [753](#), [761](#), [842](#)
`\g_ctex_ltj_reset_alternate_prop`
..... [1018](#), [1536](#), [1548](#), [1556](#)
`_ctex_ltj_save_alternate_shape:Nnn` [1468](#), [1493](#), [1509](#)
`_ctex_ltj_save_fontname:w` [763](#), [767](#)
`_ctex_ltj_select_font_aux:` [742](#), [748](#)
`_ctex_ltj_set_alternate_family:n` [1040](#), [1047](#)
`_ctex_ltj_tmp:w` [1343](#), [1360](#), [1438](#), [1440](#), [1441](#)
`\l_ctex_ltj_tmp_seq` [1351](#), [1352](#), [1355](#), [1362](#)
`\l_ctex_ltj_tmp_tl`
..... [984](#), [985](#), [1009](#), [1020](#), [1023](#), [1161](#), [1169](#),
[1203](#), [1206](#), [1208](#), [1288](#), [1293](#), [1311](#), [1312](#), [1314](#), [1315](#),
[1349](#), [1350](#), [1351](#), [1388](#), [1390](#), [1391](#), [1398](#), [1608](#), [1616](#), [1625](#)
`\g_ctex_ltj_unknown_family_seq` ... [1086](#), [1088](#), [1093](#)
`_ctex_ltj_update_family_aux:` [1271](#), [1273](#)
`_ctex_ltj_update_family_uid:N` [991](#), [1404](#), [1443](#)
`_ctex_ltj_use_global_options:N` [992](#), [999](#), [1386](#), [1405](#)
`\c_ctex_macosx_file_str` [1817](#), [1821](#)
`\l_ctex_msyh_suffix_tl`
..... [4181](#), [4182](#), [4186](#), [4203](#), [4205](#), [4239](#)
`\l_ctex_parameter_str` [5267](#), [5286](#), [5307](#), [5354](#)
`\l_ctex_parameter_tl` [5307](#), [5312](#), [5323](#)
`_ctex_parse_name:nNNNnN` [5219](#), [5228](#)
`_ctex_parse_name:w` [5241](#), [5247](#), [5256](#)
`_ctex_patch_cmd:Nnnnw` [5161](#), [5166](#), [5171](#), [5325](#)
`_ctex_patch_false:w` [5197](#), [5203](#), [5340](#), [5342](#), [5359](#), [5385](#)
`_ctex_patch_rebuild:Nn` [5298](#), [5304](#), [5337](#), [5378](#), [5382](#)

<code>__ctex_patch_replace:nnnTF</code>	5332, 5344	<code>\l__ctex_ziju_tl</code>	2147, 2155, 2156, 2217, 2223
<code>__ctex_patch_rescan:NNn</code>	5306, 5307, 5308, 5316	<code>\ctexbibname</code>	2332, 2352
<code>__ctex_patch_true:w</code>	5196, 5202, 5338, 5371, 5383	<code>\CTEXcounter</code>	2267
<code>__ctex_plus_key_aux:nn</code>		<code>\CTEXdigits</code>	26, 2277
.....	2433, 2435, 2436, 2437, 2438, 2439, 2440, 2443	<code>\CTEXfilltwosides</code>	31, 1901, 1911
<code>\l__ctex_prefix_str</code>	5267, 5285, 5306, 5363, 5364	<code>\CTEXifname</code> 16, 2509, 2601, 2643, 2782, 2978, 2986, 2994,	
<code>\l__ctex_prefix_tl</code>	5306, 5311, 5322	3230, 3272, 3275, 3279, 3291, 3295, 3303, 3306, 3311, 3314	
<code>\l__ctex_punct_bounds_clist</code> ...	4944, 4950, 4956, 4962	<code>\CTEXindent</code>	31, 2250
<code>\c__ctex_punct_family_clist</code>	602, 612	<code>\CTEXnoindent</code>	31, 2250
<code>\l__ctex_punct_font</code>	4942, 4943	<code>\CTEXnumber</code>	26, 2277
<code>\c__ctex_punct_seq</code>	4922, 4945	<code>\CTEXnumberline</code>	2992, 3012, 3494
<code>\l__ctex_punct_tl</code>		<code>\CTEXoptions</code>	31, 2112
.....	275, 571, 572, 658, 946, 2096, 2097, 2098, 2099	<code>\ctexset</code>	5, 2109, 2116, 2122
<code>__ctex_rebuild_cmd:w</code> ...	5298, 5299, 5303, 5378, 5379	<code>\CTEXsetfont</code>	31, 2142
<code>\l__ctex_replacement_str</code>		<code>\CTEXsetup</code>	31, 2112
.....	5267, 5287, 5308, 5346, 5347, 5381	<code>\CTEXsout</code>	31
<code>\l__ctex_replacement_tl</code>	5308, 5313, 5324	<code>\ctexspadef</code>	648, 4949
<code>__ctex_restore_selectfont:</code>	1990, 1994, 2003	<code>\CTEXthechapter</code>	16, 2725, 2987, 3165, 3291, 3311
<code>__ctex_right_hook_aux:nn</code>	5375	<code>\CTEXtheparagraph</code>	16
<code>__ctex_save_bounds:n</code>	4946, 4954	<code>\CTEXthepart</code>	16, 2979, 3170
<code>__ctex_save_cmap:Nn</code>	453, 457	<code>\CTEXthesection</code>	16, 3272, 3279, 3295, 3303, 3314
<code>__ctex_save_font_size:nn</code>	3792	<code>\CTEXthesubparagraph</code>	16
<code>\l__ctex_scheme_tl</code>	310, 3415, 3695	<code>\CTEXthesubsection</code>	16, 3275, 3306
<code>\g__ctex_section_depth_int</code>		<code>\CTEXthesubsubsection</code>	16
.....	94, 304, 306, 3617, 3633, 3649	<code>\CTEXunderdblline</code>	31
<code>\c__ctex_section_headings_seq</code> .	2359, 2365, 2999, 3113	<code>\CTEXunderdot</code>	31
<code>__ctex_set_font_size:nnNn</code>	3864, 3867	<code>\CTEXunderline</code>	31
<code>\g__ctex_spa_iow</code>	4932, 4935, 4937, 4947	<code>\CTEXunderwave</code>	31
<code>\c__ctex_std_class_tl</code> ...	3666, 3681, 3682, 3748, 3771	<code>\CTEXxout</code>	31
<code>\g__ctex_std_options_clist</code>		<code>\CurrentOption</code>	389
.....	131, 377, 389, 399, 401, 405, 410, 415, 420	<code>\cwd</code>	2043
<code>__ctex_titlesec_format:n</code>	3109, 3117, 3120		
<code>__ctex_titlesec_format:Nn</code>	3099, 3123, 3125		
<code>__ctex_titlesec_format_explicit:Nn</code>	3100, 3135		
<code>__ctex_titlesec_hook:n</code>	3109, 3113, 3115		
<code>__ctex_titlesec_spacing:Nn</code>	3118, 3145		
<code>__ctex_titlesec_spacing:nnnnn</code>	3146, 3147		
<code>__ctex_tmp:N</code>	1986, 1997		
<code>__ctex_tmp:w</code>	5207, 5260, 5271, 5292		
<code>\l__ctex_tmp_box</code>	58, 1931, 1932, 3239, 3243		
<code>\l__ctex_tmp_dim</code>	58,		
	2173, 2174, 2179, 2182, 2188, 2227, 2229, 2231, 2240, 2242		
<code>\l__ctex_tmp_int</code>	58, 552,		
	554, 555, 557, 561, 563, 2187, 2189, 2190, 2196, 2197,		
	2229, 2230, 2231, 2234, 2235, 2240, 2242, 2998, 3001, 3005		
<code>\l__ctex_tmp_tl</code>			
.....	58, 361, 368, 383, 384, 459, 460, 551, 553,		
	1988, 1999, 2405, 2463, 2476, 2486, 3747, 3748, 3836, 3837		
<code>\c__ctex_upper_case_file_str</code>	1805, 1813		
<code>\c__ctex_version_tl</code>	3, 6, 7, 8, 9, 20, 23		
<code>__ctex_write_family:nn</code>	4934, 4939		
<code>\l__ctex_xkanjiskip_skip</code>	2021		
<code>\l__ctex_xkanjiskip_tl</code>	2028, 2033, 2034		
<code>\c__ctex_zero_tl</code>	2147, 2154, 2156		
<code>\g__ctex_zhmCJK_bool</code>	125, 255, 260, 265, 481, 575		
<code>\l__ctex_ziju_dim</code> 2223, 2224, 2228, 2239, 2241, 2245, 2249			

D

<code>declarecharrange</code>	27, 1558
<code>\DeclareErrorKanjiFont</code>	1677
<code>\DeclareFontEncoding</code>	948
<code>\DeclareFontFamily</code>	
.....	467, 957, 4066, 4067, 4068, 4069, 4070, 4071
<code>\DeclareFontShape</code>	
.....	958, 960, 1700, 1701, 1704, 1705, 4080, 4081,
	4082, 4083, 4084, 4085, 4086, 4087, 4088, 4091, 4092,
	4093, 4094, 4095, 4096, 4097, 4098, 4099, 4102, 4103,
	4104, 4107, 4108, 4109, 4114, 4115, 4116, 4117, 4118,
	4119, 4120, 4121, 4122, 4125, 4126, 4127, 4128, 4129,
	4130, 4131, 4132, 4133, 4136, 4137, 4140, 4141, 4146,
	4147, 4148, 4149, 4150, 4151, 4152, 4153, 4154, 4157,
	4158, 4159, 4160, 4161, 4162, 4163, 4164, 4165, 4168, 4171
<code>\DeclareFontSubstitution</code>	951
<code>\DeclareKanjiFamily</code>	
.....	1698, 1699, 4072, 4073, 4074, 4075, 4076, 4077
<code>\DeclareKanjiSubstitution</code>	1678, 1679
<code>\DeclareMathSizes</code>	3829
<code>\DeclareSymbolFont</code>	963, 1321, 1680
<code>\def</code>	2575, 2615, 2650, 2670,
	2685, 2716, 2757, 2767, 2792, 2812, 2838, 2842, 2878,
	2893, 2908, 2909, 2913, 3889, 3900, 3928, 3939, 4614, 4650

`\defaultCJKfontfeatures` [1223](#)
`\defbeamertemplate` [3320, 3340, 3360](#)
`\defbeamertemplatealias` [3380, 3381, 3382](#)
dim commands:
`\dim_abs:n` [2241](#)
`\dim_add:Nn` [2224](#)
`\dim_compare:nNnTF` .. [2174, 2209, 2225, 2232, 2493, 2952](#)
`\dim_eval:n` [2194, 2240](#)
`\dim_max:nn` [3242](#)
`\dim_min:nn` [2241](#)
`\dim_new:N` [61, 1941, 2044, 2045, 2046, 2249](#)
`\dim_set:Nn` [1932, 1975, 1976, 1977,](#)
[2173, 2179, 2210, 2223, 2227, 2254, 2500, 2945, 3240, 3959](#)
`\dim_sub:Nn` [2231](#)
`\dim_to_decimal:n` [3800, 3801, 3838](#)
`\dim_to_decimal_in_unit:nn` [4969](#)
`\dim_use:N` [117, 3959](#)
`\dim_zero:N` [2260](#)
`\c_max_dim` [2174, 2493](#)
`\c_zero_dim` [2182, 2194, 2209, 2225, 2232, 2418, 2952, 3650](#)
`\dimexpr` [3329, 3349, 3369](#)
`\directlua` [699, 700](#)
`\document` [5125, 5128](#)

E

`\edef` [4633, 4634](#)
`\else` . [1687, 1688, 1692, 1693, 2547, 2556, 2565, 2581, 2586,](#)
[2621, 2626, 2704, 2712, 2728, 2734, 2740, 2753, 2763,](#)
[2823, 2828, 2847, 2852, 2872, 2877, 2886, 2907, 2923,](#)
[2936, 3457, 3485, 4645, 4646, 4692, 4731, 4762, 4821, 4884](#)

else commands:

`\else:` [484, 724, 817, 863, 1960, 3065, 3276](#)
`\em` [1685](#)
`\eminnershape` [1687, 1688, 1693, 1695](#)
`\end` [3336, 3356, 3376, 5014](#)
`\endCJKfilltwosides` [1907](#)
`\endcsname` [2830,](#)
[2839, 2840, 2841, 2848, 2867, 2868, 2870, 2873, 2882,](#)
[2883, 2884, 2887, 4640, 4642, 4643, 4644, 4647, 4649, 4653](#)
`\endCTEXfilltwosides` [1907, 1912](#)
`\endgroup` [2869, 2906, 2931, 3338, 3358, 3378, 4633, 4641, 4656](#)
`\endinput` [4638](#)
`\endlinechar` [4612, 4613](#)
`\everypar` [2827, 2926, 2938](#)
`\EverySelectfont` [2007, 2010](#)

exp commands:

`\exp_after:wN`
[116, 770, 786, 788, 863, 866, 3146, 3782, 3829, 3864, 5276](#)
`\exp_args:cc` [3390](#)
`\exp_args:Nc` [3118, 3123, 5230](#)
`\exp_args:Nco` [3228](#)
`\exp_args:Nnc` [102](#)
`\exp_args:NNf` [1616](#)
`\exp_args:NNNo` [3958](#)
`\exp_args:Nno` [3402](#)
`\exp_args:No` [553, 1814, 3142, 4946, 5292](#)

`\exp_args:Nx` [288, 581, 655, 2214, 3778, 3957](#)
`\exp_last_unbraced:Nn` [5241](#)
`\exp_last_unbraced:NNn` [1998](#)
`\exp_last_unbraced:NNNNn` [119](#)
`\exp_not:N`
[463, 583, 584, 585, 732, 733, 735, 805, 841, 951, 1036,](#)
[1037, 1038, 1039, 1395, 1516, 1744, 1745, 1746, 1760,](#)
[1896, 1897, 1898, 1925, 2379, 2380, 2381, 2386, 2388,](#)
[2389, 2390, 2391, 2394, 2395, 2398, 2410, 2411, 2412,](#)
[2413, 2414, 2423, 2424, 2425, 2427, 2430, 2445, 2447,](#)
[2449, 3004, 3006, 3007, 3008, 3009, 3130, 3131, 3140,](#)
[3141, 3178, 3180, 3181, 3798, 4000, 5221, 5222, 5223, 5311](#)
`\exp_not:n` [69, 115,](#)
[383, 734, 736, 1032, 1033, 1034, 1035, 1168, 1169, 1396,](#)
[1397, 1398, 1422, 1516, 1739, 1761, 1981, 2431, 2486,](#)
[3149, 3391, 5104, 5128, 5239, 5311, 5312, 5313, 5368, 5369](#)
`\exp_stop_f:` [722, 723](#)
`\expandafter`
[4615, 4616, 4640, 4641, 4642, 4643, 4644, 4649, 4653, 4654](#)
`\ExplSyntaxOff` [66, 72, 472,](#)
[515, 519, 523, 527, 678, 1686, 2000, 3251, 3319, 4975, 5174](#)
`\ExplSyntaxOn` [71, 710, 3383, 4915](#)

F

`\familydefault` [1269, 1280, 1742](#)
`fancyhdr` [30, 336](#)
`\fangsong` [4594](#)
`\fi` .. [2538, 2543, 2549, 2558, 2567, 2570, 2585, 2590, 2599,](#)
[2610, 2625, 2630, 2641, 2665, 2688, 2696, 2697, 2700,](#)
[2704, 2714, 2732, 2738, 2744, 2756, 2766, 2770, 2779,](#)
[2780, 2789, 2795, 2808, 2813, 2819, 2825, 2832, 2834,](#)
[2856, 2857, 2874, 2888, 2891, 2911, 2920, 2939, 2940,](#)
[3271, 3274, 3278, 3288, 3289, 3294, 3302, 3305, 3310,](#)
[3313, 3332, 3352, 3372, 3457, 3485, 4645, 4646, 4655, 4908](#)

fi commands:

`\fi:` [402, 531,](#)
[568, 587, 726, 817, 863, 1960, 2005, 3068, 3280, 3296, 3873](#)
`\figurename` [2313](#)
`figurename` [12](#)

file commands:

`\g_file_curr_name_str` [670, 1814](#)
`\file_if_exist:nTF`
[48, 1801, 1803, 1805, 1817, 3993, 4176, 4183, 4185, 4433](#)
`\file_if_exist_input:nTF` [86](#)
`\file_input:n` [78, 87, 89, 504, 540](#)
`fntef` [31, 336](#)
`\font` [4066, 4067, 4068, 4069, 4070, 4071](#)
`\fontfamily` [1262](#)
`fontset` [7, 220, 4013](#)
`\fontsize` [3782](#)

fontspec commands:

`\g_fontspec_encoding_tl` [956](#)
`\l_fontspec_family_tl` [1455](#)
`\fontspec_set_family:Nnn` [1034, 1432](#)
`\fontspec_visible_space:` [720](#)
`\fontspec_visible_space_fallback:` [725](#)

fontspec internal commands:

__fontspec_fontname_wrap:n 56, 975
 \footnotesep 3958
 \footnotesize 3895, 3934, 3958
 fp commands:
 \fp_compare:nNnTF 3742, 3769, 3955
 \fp_eval:n 2217, 4965
 \fp_set:Nn 3743, 3770
 \fp_use:N 2154, 3957
 \c_nan_fp 177
 \c_zero_fp 2154, 3742, 3769, 3955

G

GBK 6, 216
 \gdef 3030, 3031, 3032, 3033, 3036, 3037, 3038, 3039
 \global 2708, 2925, 2928
 \globaljfont 788
 \glueexpr 2864, 2879,
 2901, 2908, 2909, 2935, 3327, 3337, 3347, 3357, 3367, 3377
 group commands:
 \group_begin: 467,
 715, 750, 772, 798, 840, 867, 915, 980, 1030, 1341, 1426,
 1511, 1531, 1547, 1738, 1985, 2048, 2385, 2393, 2461,
 3054, 3187, 3246, 3958, 4941, 5122, 5198, 5206, 5215, 5270
 \group_end: 471, 480, 719, 755,
 776, 811, 844, 871, 919, 995, 1041, 1361, 1433, 1517,
 1543, 1554, 1750, 1998, 2066, 2396, 2400, 2485, 3069,
 3210, 3250, 3958, 4952, 5122, 5218, 5265, 5293, 5327, 5351
 \group_insert_after:N 808, 809
 \gtfamily 1687, 1692

H

hbox commands:

\hbox_set:Nn 1931, 3239
 heading 9, 299
 \heiti 4593
 \hskip ... 2864, 2879, 2901, 2908, 2909, 2934, 2935, 3302, 3305
 \hspace 2580, 2620, 2979, 2987, 3170
 \Huge 2644, 2679, 2783, 2802, 3461, 3492, 3912, 3951
 \huge 2602,
 2640, 2659, 2778, 3443, 3460, 3464, 3490, 3498, 3911, 3950
 hyperref 31, 336
 \hypersetup 353, 354, 1827
 \hyphenchar 4066, 4067, 4068, 4069, 4070, 4071

I

if commands:

\if_bool:N 481, 575
 \if_case:w 398, 3869
 \if_cs_exist:N 548, 1984
 \if_false: 108
 \if_meaning:w 1959
 \if_true: 107
 \ifctexpdf 106
 \ifdim 2817, 2860, 2897, 2915
 \IfNoValueF 2117, 2123
 \IfNoValueTF 1183, 1204, 1480, 1585, 1618, 1620, 2457

\ifnum 2576,
 2597, 2616, 2639, 2717, 2776, 2843, 2872, 2886, 3271,
 3274, 3278, 3285, 3286, 3294, 3302, 3305, 3310, 3313, 4645
 \ifodd 2543, 2545, 2570, 2577, 2610,
 2617, 2665, 2688, 2710, 2721, 2770, 2789, 2795, 2808,
 2821, 2832, 2848, 2861, 2898, 2916, 2920, 3330, 3350, 3370
 \ifx 4643, 4644, 4649
 \ifzhmappdf 4661, 4728, 4759, 4790, 4857
 \ignorespaces 2941
 \immediate 4652
 indent 30, 196
 \indexname 2316
 indexname 12
 \input 4914, 4978
 \insertcontinuationtext 2325, 2334, 2341
 \insertpart 3334, 3335
 \insertromanpartnumber 3325, 3430
 \insertsection 3354, 3355
 \insertsectionnumber 3345, 3529
 \insertsubsection 3374, 3375
 \insertsubsectionnumber 3365, 3558

int commands:

\int_compare:nNnTF
 554, 2190, 2234, 3617, 3633, 3649, 3740, 3767
 \int_const:Nn 967
 \int_div_truncate:nn 80, 561
 \int_eval:n 1481, 1483, 1484, 1586, 1588, 1589
 \int_from_hex:n 553
 \int_gincr:N 1126, 1446
 \int_gset:Nn 96, 156, 157, 158, 304, 306, 372, 376, 3741, 3768
 \int_incr:N 3001
 \int_mod:nn 563
 \int_new:N 59, 95, 146, 1449
 \int_set:Nn 147, 538, 544, 552, 2187, 2229
 \int_sub:Nn 557, 2235
 \int_to_Hex:n 555, 560, 562
 \int_use:N ... 464, 544, 855, 1128, 1447, 2057, 3005, 4946
 \int_zero:N 2998
 \c_zero_int 2190
 \interlinepenalty
 2593, 2635, 2654, 2674, 2775, 2800, 2865, 2866, 2902, 2903

iow commands:

\iow_close:N 4935
 \iow_indent:n 5109
 \iow_new:N 4937
 \iow_now:Nn 4947
 \iow_open:Nn 4932
 \itemsep 3892, 3903, 3931, 3942
 \itshape 1687, 1688, 1692, 1693

J

\jfam 907, 968, 1682

K

\kaishu 4595
 \kanjifamily 1655, 1662, 1669, 1674
 \kanjifamilydefault 1650

kernel internal commands:

\l_kernel_expl_bool 70

keys commands:

\l_keys_choice_tl 289
 \keys_define:nn 150, 383, 386, 972,
 1334, 1443, 1520, 1558, 1597, 2071, 2092, 2125, 2159,
 2282, 2308, 2336, 2346, 2349, 2352, 2486, 3014, 3966, 4013
 \keys_if_exist:nnTF 1595
 \l_keys_key_tl 135, 137, 139
 \keys_set:nn 29, 166,
 172, 228, 237, 246, 272, 282, 289, 296, 326, 332, 2110,
 2117, 2123, 3418, 3476, 3505, 3537, 3572, 3589, 3603,
 3619, 3627, 3635, 3643, 3650, 3651, 3653, 3719, 3721, 4055
 \keys_set_known:nn 3698
 \keys_set_known:nnN 984

L

\labelformat 3402
 \languagealias 3727, 3732, 3753, 3758
 \LARGE 3910, 3949
 \Large 2598, 3441, 3447, 3516, 3520, 3909, 3948
 \large 3541, 3908, 3947
 \lastbox 2929
 \leavevmode 2538, 2813

left commands:

\c_left_brace_str 5262
 \leftmargin 3889, 3900, 3928, 3939
 \leftmargini 3889, 3900, 3928, 3939
 \let 1993, 2846, 2855, 3882, 3921, 4636, 4642
 \linespread 3957
 linespread 10, 176
 linestretch 11, 2159
 \linewidth 2188, 2196, 2228
 \lishu 4597
 \listfigurename 2311
 listfigurename 12
 \listtablename 2312
 listtablename 12
 \LoadClass 406, 411, 416, 421
 \long 4650
 \lstlistingname 733, 734
 \lstlistlistingname 735, 736
 \ltjalchar 723
 \ltjdefcharrange 679, 680, 681, 684, 688, 689, 692, 695, 696
 LTJFONTUID 1443
 \ltjgetparameter 1936, 1963, 2025, 2039

ltjitaliccorr internal commands:

_ctex_ltjitaliccorr 1637
 \ltjsetkanjiskip 1641
 \ltjsetmathletter 917
 \ltjsetparameter 698, 704, 1636
 \ltjsetxkanjiskip 1643

lua commands:

\lua_escape:e
 .. 816, 828, 855, 862, 1471, 1486, 1487, 1504, 1505, 1514

\lua_now:e 716, 790,
 816, 820, 825, 830, 852, 861, 954, 1492, 1516, 1770, 2050

M

\MAKESPA 4938, 4980
 \markboth 2508, 2604, 2633
 \mathgroup 907
 \mathrm 1653
 \mathsf 1660
 \mathtt 1667
 \mcfamily 1692
 \mddefault 952, 958, 964, 1322, 1327

mode commands:

\mode_if_math:TF 901

msg commands:

\msg_critical:nnn 53, 670, 3982
 \msg_error:nn 4033
 \msg_error:nnn 19, 22, 38, 41, 661, 674, 677, 2298, 3783, 3865
 \msg_error:nnnn 3999, 4028
 \msg_fatal:nn 4920
 \msg_new:nn 4918
 \msg_new:nnn 134, 136, 138, 140, 142, 664,
 1024, 1094, 1176, 1883, 2329, 3672, 3983, 4042, 5105, 5182
 \msg_new:nnnn 12, 30, 45, 100, 2300, 3785, 4007, 4036
 \msg_warning:nn 1171, 1887, 2089, 2100, 2351
 \msg_warning:nnn
 164, 170, 198, 208, 223, 232, 241, 270, 280, 294,
 324, 330, 338, 343, 349, 1089, 3691, 3970, 4020, 5111, 5181
 \msg_warning:nnnn 1020, 1896, 1903, 2115, 2121, 2252, 2258

N

\newCJKfontfamily 1195
 \NewDocumentCommand
 1055, 1190, 1195, 1201, 1211, 1217, 1223, 1230, 1236,
 1242, 1248, 1381, 1473, 1578, 1613, 1672, 2110, 2113,
 2119, 2142, 2213, 2250, 2256, 2277, 2279, 2451, 3777,
 3962, 4592, 4593, 4594, 4595, 4597, 4598, 4600, 4602, 4603
 \newfontfeature 969, 970, 971
 \newluafunction 2047
 \newpage 2690, 2695
 \ngostype 44, 1788
 \nobreak 2598,
 2607, 2662, 2778, 2783, 2786, 2802, 2805, 2917, 3442, 3491
 \nobreakspace 2598, 2640
 nocap 30
 \noexpand 4634, 4636, 4637, 4638
 nofonts 30
 \noindent 2959
 noindent 30
 nopunct 30
 \normalem 1866
 \normalfont 527, 529,
 530, 1227, 1234, 1240, 1305, 1684, 2594, 2596, 2636,
 2638, 2655, 2657, 2675, 2677, 2772, 2774, 2797, 2799, 3009
 \normalsize 3575, 3592, 3606, 3876, 3915, 3953, 3961
 nospace 30
 nozhmap 30

\null 2568, 2693
 \numberline 2727, 2873, 2887, 2987, 2995, 3181, 3237

O

\onecolumn 2563
 or commands:
 \or: 400, 3871

P

\pagestyle 3718
 \par 2499, 2539, 2598, 2640, 2644, 2659,
 2679, 2778, 2783, 2802, 2814, 2917, 3326, 3327, 3334,
 3337, 3346, 3347, 3354, 3357, 3366, 3367, 3374, 3377,
 3420, 3442, 3453, 3471, 3479, 3491, 3532, 3534, 3566, 3568
 paragraph/afterskip 32
 paragraph/beforeskip 32
 paragraph/hang 22
 paragraph/numbering 15
 paragraph/runin 20
 \parindent 2209, 2210, 2232, 2253, 2254, 2259, 2260, 2594,
 2655, 2772, 2797, 2945, 2952, 2953, 3329, 3349, 3369, 3651
 \parsep 3891, 3892, 3902, 3903, 3930, 3931, 3941, 3942
 \part 2536
 part/beforeskip 32
 part/fixbeforeskip 32
 part/hang 22
 part/numbering 15
 part/pagestyle 22, 2461
 \partmark 2507, 2605, 2634
 \partname 2598, 2640, 3325, 3425
 \PassOptionsToClass 405, 410, 415, 420
 \PassOptionsToPackage 482, 1831, 2262, 5113
 \pdfmapline 4662,
 4663, 4664, 4665, 4666, 4667, 4668, 4670, 4671, 4672,
 4673, 4674, 4675, 4676, 4678, 4679, 4680, 4681, 4682,
 4683, 4685, 4686, 4687, 4688, 4689, 4690, 4791, 4792,
 4793, 4794, 4795, 4796, 4797, 4799, 4800, 4801, 4802,
 4803, 4804, 4805, 4807, 4808, 4809, 4810, 4811, 4812,
 4814, 4815, 4816, 4817, 4818, 4819, 4858, 4859, 4860,
 4861, 4862, 4863, 4865, 4866, 4867, 4868, 4869, 4870,
 4872, 4873, 4874, 4875, 4876, 4878, 4879, 4880, 4881, 4882
 \pdfoutput 4645
 \pdfstringdefDisableCommands 3965
 \pingfang 4603
 prg commands:
 \prg_do_nothing:
 911, 1420, 1512, 1912, 1983, 2076, 2971, 5303
 \prg_generate_conditional_variant:Nnn 1081
 \prg_new_conditional:Npnn 814, 1956
 \prg_new_protected_conditional:Npnn 1068, 5273
 \prg_return_false: ... 817, 1078, 1960, 1964, 1968, 5283
 \prg_return_true: 817, 1071, 1076, 1960, 1964, 1968, 5288
 \primitive 5014
 \ProcessKeysOptions 395
 \proofname 2303, 2318
 proofname 12

prop commands:

\prop_get:NnN 1163
 \prop_get:NnNTF
 1070, 1122, 1160, 1413, 1461, 3781, 3836, 3863
 \prop_gpop:NnNTF 1009, 1013, 1548
 \prop_gput:Nnn ... 988, 989, 1036, 1150, 1536, 1572, 3798
 \prop_gremove:Nn 1018
 \prop_if_empty:NTF 1084, 1286
 \prop_map_break:n 1300
 \prop_map_inline:Nn 1298
 \prop_new:N 1004, 1005, 1006, 1155, 1556, 1577, 3792
 \protect 2521, 2527, 2727,
 2748, 2749, 2873, 2887, 2987, 2995, 3180, 3192, 3194,
 3198, 3200, 3204, 3206, 3214, 3219, 3231, 3232, 5234, 5235
 \ProvideDocumentCommand 2507
 \ProvidesFile 4650, 4658, 4725, 4756, 4787, 4854
 \providetranslation
 5019, 5020, 5021, 5022, 5023, 5024, 5025,
 5026, 5027, 5028, 5029, 5030, 5031, 5032, 5033, 5034,
 5035, 5036, 5037, 5038, 5039, 5040, 5041, 5042, 5043,
 5044, 5045, 5046, 5047, 5048, 5049, 5050, 5051, 5052,
 5053, 5054, 5055, 5056, 5057, 5058, 5059, 5060, 5061,
 5062, 5063, 5064, 5065, 5066, 5067, 5068, 5069, 5070,
 5071, 5072, 5073, 5074, 5075, 5076, 5077, 5078, 5079,
 5080, 5081, 5082, 5083, 5084, 5085, 5086, 5087, 5088,
 5089, 5090, 5091, 5092, 5093, 5094, 5095, 5096, 5097, 5098
 punct 10, 30, 275, 2092
 \punctstyle 572, 2097

Q

\quad 2839,
 3271, 3272, 3274, 3275, 3278, 3279, 3291, 3295, 3303,
 3306, 3311, 3314, 3448, 3499, 3509, 3542, 3576, 3593, 3607

quark commands:

\q_mark 5277, 5280
 \q_stop 117, 120, 866,
 869, 1378, 1382, 1609, 1614, 5242, 5247, 5256, 5277, 5280
 \quotation 3975

R

\raggedright 2594, 2655, 2772, 2797, 3440, 3489
 \refname 2319, 2324, 2331, 2340, 2348, 2349
 refname 12
 \refstepcounter ... 2579, 2619, 2723, 2850, 3385, 3407, 3408
 \relax 2616, 2639, 2815,
 2851, 2859, 2864, 2879, 2896, 2901, 2908, 2909, 2914,
 2933, 2935, 3302, 3305, 3327, 3329, 3337, 3347, 3349,
 3357, 3367, 3369, 3377, 4608, 4612, 4615, 4643, 4644, 4649
 \renewcommand 2537, 2553, 2703
 \RenewDocumentCommand 1262, 1651, 1658, 1665, 3408
 \RequirePackage 27,
 56, 57, 345, 356, 483, 486, 487, 488, 654, 672, 675,
 712, 732, 1731, 1840, 1849, 1865, 1871, 2263, 3772, 5158
 \RequirePackageWithOptions 5114
 resetalternatfont 28, 1520
 reverse commands:
 \reverse_if:N 575

\rmdefault 1277, 1654, 1744
 \rmfamily 515, 518, 1651, 1653
 \romanfamily 1654, 1661, 1668

S

scan commands:

\scan_stop:
 ... 65, 117, 788, 907, 1513, 1514, 2180, 2188, 4942, 5124
 scheme 9, 310
 \scriptsize 3906, 3945
 \secdef 2550, 2572, 2715
 section/afterskip 32
 section/beforeskip 32
 section/hang 22
 section/numbering 15
 section/runin 20
 \sectionname 3345, 3526
 \selectfont 1062, 1539,
 1552, 1656, 1663, 1670, 1675, 1997, 2004, 2018, 2144, 3782

seq commands:

\seq_clear:N 982
 \seq_concat:NNN 1354
 \seq_const_from_clist:Nn . 434, 2360, 2370, 3044, 4922
 \seq_gput_left:Nn 2366, 2367
 \seq_gput_right:Nn 1088, 3804
 \seq_gset_eq:NN 2365
 \seq_if_empty:NTF 1369
 \seq_if_in:NnTF 440, 1086
 \seq_map_function:NN 3113, 3235
 \seq_map_inline:Nn
 1377, 2478, 2999, 3047, 3057, 3066, 3089, 3172, 3396, 4945
 \seq_new:N 1093, 1362, 1363, 2364, 3793
 \seq_put_right:Nn 1625
 \seq_set_filter:NNn 1352
 \seq_set_split:Nnn 1351
 \seq_use:Nnnn 3790
 \setbox 2929
 \setCJKfamilyfont 1107, 1195, 4204, 4210, 4211,
 4212, 4213, 4214, 4215, 4258, 4259, 4267, 4268, 4270,
 4271, 4272, 4273, 4275, 4297, 4299, 4301, 4303, 4342,
 4343, 4344, 4345, 4369, 4374, 4379, 4381, 4423, 4425,
 4427, 4428, 4460, 4461, 4464, 4469, 4470, 4472, 4473,
 4474, 4487, 4488, 4489, 4490, 4491, 4492, 4532, 4533,
 4534, 4535, 4536, 4537, 4550, 4551, 4552, 4585, 4586, 4587
 \setCJKmainfont 1103, 1223, 4195, 4254, 4262,
 4289, 4335, 4357, 4414, 4445, 4449, 4483, 4528, 4546, 4581
 \setCJKmathfont 1223
 \setCJKmonofont 1105, 1223, 4209, 4257,
 4266, 4296, 4341, 4368, 4422, 4458, 4486, 4531, 4549, 4584
 \setCJKromanfont 1229, 1255
 \setCJKsansfont 1104, 1223, 4198, 4202, 4256, 4264,
 4295, 4340, 4363, 4420, 4446, 4456, 4485, 4530, 4548, 4583
 \SetSymbolFont 965, 1326, 1330, 1681
 \sfdefault 1278, 1661, 1745
 \sffamily 519, 522, 1658, 1660

skip commands:

\skip_horizontal:N 1946
 \skip_horizontal:n 2953
 \skip_if_eq:nnTF 1963, 1967, 2024, 2518, 2524
 \skip_new:N 1955, 2037, 2503
 \skip_set:Nn 1936,
 1939, 2028, 2038, 2181, 2192, 2237, 2245, 2505, 2517, 2523
 \skip_set_eq:NN 1953, 2030
 \skip_sub:Nn 2494, 2501
 \skip_use:N 2521, 2527
 \skip_zero:N 2201
 \c_zero_skip 2416, 2417, 2469, 2470, 2518, 2524
 \small 3884, 3923
 \songti 4592
 \space 2724, 2778, 3425, 3487, 3526, 3557, 3659
 space 10, 30, 285, 2071
 \special 1713, 1714, 1717,
 1718, 1847, 1861, 4693, 4694, 4695, 4696, 4697, 4698,
 4699, 4701, 4702, 4703, 4704, 4705, 4706, 4707, 4709,
 4710, 4711, 4712, 4713, 4714, 4716, 4717, 4718, 4719,
 4720, 4721, 4732, 4733, 4734, 4735, 4736, 4738, 4739,
 4740, 4741, 4742, 4744, 4745, 4746, 4747, 4749, 4750,
 4751, 4752, 4763, 4764, 4765, 4766, 4767, 4769, 4770,
 4771, 4772, 4773, 4775, 4776, 4777, 4778, 4780, 4781,
 4782, 4783, 4822, 4823, 4824, 4825, 4826, 4827, 4828,
 4830, 4831, 4832, 4833, 4834, 4835, 4836, 4838, 4839,
 4840, 4841, 4842, 4843, 4845, 4846, 4847, 4848, 4849,
 4850, 4885, 4886, 4887, 4888, 4889, 4891, 4892, 4893,
 4894, 4895, 4897, 4898, 4899, 4900, 4902, 4903, 4904, 4905
 \SplitArgument 1474, 1579, 2452
 str commands:
 \c_backslash_str 5261
 \c_colon_str 893, 894, 895
 \str_case:nnTF 1742, 1788, 5232, 5247
 \str_case_e:nn 1275
 \str_case_e:nnTF 1101, 1112
 \str_const:Nn 43, 50, 1813, 1821
 \str_if_empty:NTF 5318, 5354, 5363
 \str_if_eq:eeTF 571, 1290, 4019, 5238
 \str_if_eq:nnTF 485, 576,
 940, 1269, 1836, 1844, 3723, 3750, 3764, 4022, 4039, 4046
 \str_lower_case:n 459
 \str_new:N 5158, 5267, 5268, 5269
 \str_set:Nn 5285, 5286, 5287
 \str_upper_case:n 1814
 \strutbox 3959
 sub3section 9, 302
 sub4section 9, 302
 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	\@minus	3511, 3544, 3578, 3594, 3608, 3878, 3880, 3886, 3888, 3890, 3891, 3897, 3899, 3901, 3902, 3917, 3919, 3925, 3927, 3929, 3930, 3936, 3938, 3940, 3941
subsection/runin	20	\@namedef ..	3192, 3194, 3198, 3200, 3204, 3206, 3214, 3219
\subsectionname	3365, 3557	\@nameuse	3192, 3194, 3204, 3206, 3214
subsubsection/afterskip	32	\@ne	3150, 3274, 3305
subsubsection/beforeskip	32	\@nil	841
subsubsection/hang	22	\@nobreakfalse	2924
subsubsection/numbering	15	\@noskipsecfalse	2928
subsubsection/runin	20	\@noskipsectrue	2925
\symmincho	1682	\@onlypreamble	37, 431, 496, 1251, 1252, 1253, 1254, 1255, 1256, 1708, 1722, 1730, 4006
sys commands:		\@part	2550, 2572, 2574
\c_sys_engine_str	44	\@pkgextension	6, 7, 8, 9, 20, 23, 671, 5153
\sys_if_engine_pdftex:TF	124, 4190, 4282, 4350, 4438, 4479, 4542	\@plus	3454, 3455, 3511, 3512, 3544, 3545, 3578, 3579, 3594, 3608, 3622, 3638, 3878, 3879, 3880, 3886, 3887, 3888, 3890, 3891, 3897, 3898, 3899, 3901, 3902, 3917, 3918, 3919, 3925, 3926, 3927, 3929, 3930, 3936, 3937, 3938, 3940, 3941
\sys_if_engine_uptex:TF	4235, 4323, 4402, 4441, 4512, 4570	\@popfilename	134, 5151
\sys_if_engine_xetex:TF	4916	\@schapter	2715, 2757
\sys_if_output_pdf:TF	106, 469, 1839, 4284, 4352	\@seccntformat	2838, 2851
T			
\tablename	2314	\@secondoftwo	3232
tablename	12	\@secpenalty	2829, 3514, 3547, 3581, 3595, 3609
\TeX	3963	\@sect	2837, 2842
TeX and L ^A T _E X 2 _ε commands:		\@setfontsize	3868
\@italiccorr	1638	\@spart	2550, 2572, 2649
\@par	2865, 2902, 3510, 3543, 3577, 3621, 3637	\@ssect	2836, 2893
\@afterheading	2612, 2667, 2755, 2765, 2922	\@startsection	94, 96, 103, 2812, 3004
\@afterindentfalse	2548, 2709, 2713, 2818, 2824	\@svsec	2846, 2851, 2855, 2864, 2880, 2881
\@afterindenttrue	2546, 2711, 2816, 2822	\@svsechd	2878, 2908, 2909, 2931
\@Alph	3655, 3660	\@tempdima	3240, 3242
\@chapapp	2724, 2778, 3287, 3310	\@tempskipa	2815, 2817, 2818, 2829, 2859, 2860, 2896, 2897, 2914, 2915, 2918, 2933, 2934
\@chapter	2715, 2716	\@tempswafalse	2566
\@chinese	2265, 2274	\@tempswatrue	2564
\@clubpenalty	2937	\@topnewpage	2752, 2762
\@currentHref	3078	\@topnum	2708
\@currentlabel	3387, 3391	\@xsect	2892, 2912, 2913
\@currentx	4, 5104, 5153	\@zhdig	2273
\@currname	4, 5104, 5154	\@zhnum	2265, 2272
\@dblarg	2837	\abstractname	12
\@empty	102, 2846, 2855	\addCJKfontfeature	65
\@endpart	2647, 2682, 2684	\AfterEndPreamble	133
\@EverySelectfont@Init	1984, 1992	\algorithmname	12
\@firstoftwo	3231	\appendixname	12, 24
\@hangfrom	2947, 2958	\AtBeginDocument	37, 46
\@ifclassloaded	3668	\AtBeginDvi	43
\@ifpackagelater	37, 40, 660, 673, 676, 1872	\AtBeginShipoutFirst	43
\@ifpackageloaded	18, 21, 669, 1824, 3152, 3255, 3404, 5135	\AtEndOfClass	37
\@ifpackagewith	3097, 3106, 3157	\AtEndOfPackage	37
\@ifstar	2835	\AtEndOfPackageFile*	133
\@listI	3882, 3921	\AtEndPreamble	133
\@listi	3882, 3889, 3900, 3921, 3928, 3939	\baselineskip	10, 11, 77, 80
\@M	2593, 2635, 2654, 2674, 2775, 2800, 2865, 2866, 2902, 2903, 2930	\bfseries	48
\@makechapterhead	2752, 2754, 2767	\bibname	12
\@makeschapterhead	2762, 2764, 2792		

<code>\c@chapter</code>	3660	<code>\CTEX@chapter@numbering</code>	2721, 3039
<code>\c@secnumdepth</code>	2576, 2597, 2616, 2639, 2717, 2776, 2843, 2872, 2886, 3271, 3274, 3278, 3285, 3286, 3294, 3302, 3305, 3310, 3313	<code>\CTEX@chapter@pagestyle</code>	2466, 2707
<code>\c@section</code>	3655	<code>\CTEX@chapter@titleformat</code>	2784, 2803
<code>\caption</code>	31	<code>\CTEX@chapter@tocline</code>	2976
<code>\catcode</code>	36, 45, 45, 124, 134	<code>\CTEX@chaptername</code>	2782
<code>\ccwd</code> ...	11, 11, 25, 26, 31, 31, 37, 75, 75, 76, 79, 80, 80, 81	<code>\CTEX@disableautoindent</code>	2533, 2595, 2637, 2656, 2676, 2773, 2798, 3322, 3342, 3362
<code>\chapter</code>	9, 15, 16, 22, 24, 24, 86, 93, 107	<code>\CTEX@document@left@hook</code>	5121, 5127
<code>\chaptermark</code>	13	<code>\CTEX@document@right@hook</code>	5123, 5129
<code>\chaptername</code>	15	<code>\CTEX@Family@CMap</code>	445, 473, 476
<code>\char</code>	50	<code>\CTEX@fixheadingskip</code>	2492, 2497, 2543, 2610, 2665, 2688, 2789, 2808, 2832, 2920
<code>\CheckCommand</code>	76	<code>\CTEX@fixskip</code>	2832, 2920, 2966, 2973
<code>\Chinese</code>	31	<code>\CTEX@fixtopskip</code>	2489, 2570, 2770, 2795
<code>\chinese</code>	26, 31, 82	<code>\CTEX@fontfamily</code>	1033, 1262
<code>\CJK@@ignorespaces</code>	47, 589	<code>\CTEX@getttitle</code>	2591, 2631, 2653, 2673, 2745, 2760, 2858, 2895, 3084
<code>\CJK@encoding</code>	55, 57, 64, 747, 751, 835, 947, 1321, 1324, 1326, 1330, 1471, 1486, 1487, 1504, 1505	<code>\CTEX@hang</code>	2957, 2967, 2974
<code>\CJK@envStart</code>	46, 46, 47, 583	<code>\CTEX@hangfrom</code>	2864, 2901, 2955
<code>\CJK@family</code>	50, 739, 752, 799, 835, 1059, 1538, 1548, 1550, 2018, 2019	<code>\CTEX@hangindent</code>	2600, 2642, 2658, 2678, 2781, 2801, 2943
<code>\CJK@ignorespaces</code>	47, 47, 594, 598	<code>\CTEX@headingskip</code>	2506, 2544, 2571, 2611, 2666, 2689, 2771, 2790, 2796, 2809, 2833, 2921
<code>\CJK@input</code>	46, 532	<code>\CTEX@hyperheadinghook</code>	3052, 3082
<code>\CJK@loadBinding</code>	45, 46, 579	<code>\CTEX@ifnamefalse</code>	2509, 2582, 2587, 2622, 2627, 2651, 2671, 2729, 2735, 2741, 2758, 2844, 2853, 2894
<code>\CJK@loadEncoding</code>	45	<code>\CTEX@ifnametrue</code>	2509, 2578, 2618, 2722, 2849
<code>\CJK@makeActive</code>	46, 578	<code>\CTEX@makeanchor</code>	2583, 2588, 2623, 2628, 2730, 2854, 3042, 3050, 3062, 3067, 3073
<code>\CJK@plane</code>	45, 76, 448, 449, 459, 2007	<code>\CTEX@makeanchor@chapter</code>	2736, 2742
<code>\CJK@surr</code>	45, 548	<code>\CTEX@makeanchor@schapter</code>	2759
<code>\CJK@upperReset</code>	47	<code>\CTEX@makeanchor@sect</code>	2845
<code>\CJKfamily</code>	57	<code>\CTEX@makeanchor@spart</code>	2652, 2672
<code>\CJKfamilydefault</code>	46, 47, 61, 61, 69, 70, 70, 71, 71	<code>\CTEX@makeanchor@ssect</code>	2836
<code>\CJKglue</code>	80, 112	<code>\CTEX@part@afterindent</code>	2545
<code>\CJKhook</code>	47, 47	<code>\CTEX@part@aftername</code>	2601, 2643, 3331
<code>\CJKrmdefault</code>	44	<code>\CTEX@part@afterskip</code>	2609, 2664, 2687, 3337
<code>\CJKsymbol</code>	76	<code>\CTEX@part@aftertitle</code> ...	2606, 2646, 2661, 2681, 3335
<code>\CJKunderdotbasesep</code>	31	<code>\CTEX@part@beforeskip</code>	2542, 2569, 3327
<code>\CJKunderline</code>	31	<code>\CTEX@part@break</code>	2540, 2559
<code>\clearpage</code>	47	<code>\CTEX@part@fixskip</code>	2543, 2570, 2610, 2665, 2688
<code>\contentsname</code>	11, 12	<code>\CTEX@part@format</code>	2596, 2638, 2657, 2677, 3328
<code>\CTEX@addloflotskip</code>	2514, 2750	<code>\CTEX@part@indent</code>	3329
<code>\CTEX@addtocline</code> ...	2531, 2592, 2632, 2746, 2876, 2890	<code>\CTEX@part@numbering</code>	2577, 2617, 3330
<code>\CTEX@afterindent</code>	2821, 2965, 2972	<code>\CTEX@part@pagestyle</code>	2465, 2561
<code>\CTEX@aftertitle</code>	2905, 2910, 2964, 2971	<code>\CTEX@part@titleformat</code> ..	2603, 2645, 2660, 2680, 3335
<code>\CTEX@appendix@number</code>	3019, 3031, 3037	<code>\CTEX@part@tocline</code>	2976
<code>\CTEX@appendix@numbering</code>	3020, 3033, 3039	<code>\CTEX@partname</code>	2601, 2643, 3331
<code>\CTEX@chapter@afterindent</code>	2710	<code>\CTEX@postappendix</code>	3024, 3032, 3038
<code>\CTEX@chapter@aftername</code>	2782	<code>\CTEX@postchapter</code>	3038
<code>\CTEX@chapter@afterskip</code>	2788, 2807	<code>\CTEX@postsection</code>	3032
<code>\CTEX@chapter@aftertitle</code>	2785, 2804	<code>\CTEX@preappendix</code>	3023, 3030, 3036
<code>\CTEX@chapter@beforeskip</code>	2769, 2794	<code>\CTEX@prechapter</code>	3036
<code>\CTEX@chapter@break</code>	2706	<code>\CTEX@presection</code>	3030
<code>\CTEX@chapter@fixskip</code>	2770, 2789, 2795, 2808	<code>\CTEX@runin</code>	2861, 2898, 2916, 2968, 2975
<code>\CTEX@chapter@format</code>	2774, 2799	<code>\CTEX@save@appendix</code>	3025, 3028
<code>\CTEX@chapter@lofskip</code>	2467	<code>\CTEX@save@refstepcounter</code>	3407, 3410
<code>\CTEX@chapter@lotskip</code>	2468		

\CTEX@section@aftername	3351	\external@font	785, 786, 788
\CTEX@section@afterskip	3357	\extract@font	51, 773
\CTEX@section@aftertitle	3355	\f@baselineskip	2180
\CTEX@section@beforeskip	3347	\f@encoding	751, 791, 904
\CTEX@section@format	3348	\f@family	50, 752, 791, 799, 1264, 1275
\CTEX@section@indent	3349	\f@series	747, 835
\CTEX@section@numbering	3033, 3350	\f@shape	747, 835
\CTEX@section@titleformat	3355	\f@size	747, 753, 830, 838, 841, 842, 1932, 1975, 3243
\CTEX@sectionname	3351	\fangsong	8
\CTEX@selectfont@hook	1978, 2007, 2010	\figurename	12
\CTEX@selectfont@save	1989, 1993	\font	52
\CTEX@setcurrentlabel@n	3385, 3411	\font@name	50, 52, 756, 763, 764, 767, 770, 788, 789, 806, 884, 885, 886, 887, 888, 895
\CTEX@setheadingskip	2504, 2542, 2569, 2609, 2664, 2687, 2769, 2788, 2794, 2807, 2831, 2919	\fontcharwd	76
\CTEX@subsection@aftername	3371	\fontfamily	61
\CTEX@subsection@afterskip	3377	\footnotesep	112, 112
\CTEX@subsection@aftertitle	3375	\footnotesize	112
\CTEX@subsection@beforeskip	3367	\get@external@font	782
\CTEX@subsection@format	3368	\getanddefine@fonts	873
\CTEX@subsection@indent	3369	\globaljfont	51
\CTEX@subsection@numbering	3370	\H@old@chapter	3056
\CTEX@subsection@titleformat	3375	\hbox	86
\CTEX@subsectionname	3371	\heiti	8
\CTEX@thechapter	3037	\hrule	86
\CTEX@thesection	3031	\Hy@chapapp	2730, 2736, 2742, 2759
\CTEX@titleformat@n	94, 2904, 2910, 2963, 2970	\Hy@driver	1844
\CTEX@titleslabel@clear	3201, 3211	\Hy@MakeCurrentHrefAuto	3075
\CTEX@titleslabel@set	3195, 3207, 3211	\Hy@org@chapter	3056
\CTEX@toc@width@n	3237, 3252, 3262	\Hy@raisedlink	3076
\CTEX@todayold	2281, 2286	\Hy@unicodetrue	73
\CTEX@update@sectionformat@n	2820, 2961	\hyper@anchorend	3079
\CTEX@verbatim@font@hook	1631, 1635	\hyper@anchorstart	3078
\ctex@zhmap@endinput	4910	\HyPsd@ConvertToUnicode	73
\CTEXdigits	26	\HyPsd@LoadUnicode	73
\CTEXifname	16	\HyPsd@pdfencoding	73
\CTEXnumber	26, 26	\HyPsd@ToBigChars	73
\CTEXnumberline	23	\if@mainmatter	2719, 2777, 3285, 3686, 3687
\CTEXoptions	31, 31, 79	\if@nobreak	2826
\ctexset	5, 5, 7, 11, 14, 27, 28, 31, 31, 114	\if@noskipsec	2538, 2813, 2927
\CTEXsetup	31	\if@openright	2554, 2692, 2704, 3457, 3485
\CTEXthechapter	13, 16, 98	\if@tempwa	2698
\CTEXthesection	13, 16, 100	\if@twocolumn	2562, 2751, 2761
\CTEXunderdotbasesep	31	\if@twoside	2691, 3269, 3292
\CTEXunderline	31	\iffalse	54
\curr@fontshape	753, 783, 792, 796, 842	\ifHy@implicit	3055
\DeclareAlternateKanjiFont	64	\ifin@	54, 863
\DeclareFontEncoding	54	\iftrue	54
\DeclareFontFamily	43, 52	\indexname	12
\DeclareFontShape	52	\insertcontinuationtext	12
\DeclareFontShape@	64, 1430	\itdefault	52
\DeclareRobustCommand	135	\itshape	48
\define@newfont	51, 775	\jfam	69
\do@subst@correction	51, 52, 774	\kaishu	8
\em	69	\kanjiskip	77
\escapechar	54	\labelformat	102
\EverySelectfont	76	\leftmark	13

\linewidth	80	\pingfang	8
\lishu	8	\ProcessKeysOptions	29
\listfigurename	12	\proofname	12, 82
\listtablename	12	\protected	99
\ltj@does@alt@set	52	\protected@edef	2278, 2280, 2851, 3387
\ltj@getjfontnumber	53, 851	\ProvidesFile	125
\ltj@set@stackfont	893, 894, 895	\ps@fancy	3298
\ltj@allalchar	717	\ps@headings	3268
\ltj@curjfnt	51, 76	\refname	12, 12
\ltj@pickup@altfont@auxy	836	\refstepcounter	102
\ltj@pickup@altfont@copy	849	\relax	45, 79
\ltj@setpar@global	68, 892, 1641, 1643	\reset@font	529
\ltj@tempcntc	53, 855	\rightmark	13
\ltjalchar	50	\rmfamily	48
\ltjsetkanjiskip	68	\roman	26
\ltjsetparameter	68	\scantokens	135
\ltjsetxkanjiskip	68	\scriptfont@name	887, 894
\m@ne	2576, 2597, 2717, 2776, 3278, 3285, 3286, 3310, 4066, 4067, 4068, 4069, 4070, 4071	\section	9, 14, 15, 20, 20, 21, 21, 22, 23, 24, 24, 32, 84, 100
\markboth	13	\section*	15
\markright	13	\sectionmark	13
\math@bgroup	903	\sectionname	15
\math@egroup	908	\sectionpage	14, 101
\math@fonts	890	\selectfont	53, 70, 76, 77, 79
\mathbb	113	\setCJKfamilyfont	27, 27
\maxdimen	11, 80	\setCJKmainfont	27
\meaning	135, 136, 136, 137	\sf@size	886
\newcommand	135	\sffamily	48
\newrobustcmd	135	\size@update	79, 80, 2001, 2002, 2144, 2158
\newtitlemark	99	\sldefault	52
\nfss@catcodes	55	\songti	8
\ngostype	71	\special	7, 72
\normalsize	8, 108	\split@name	841
\not@math@alphabet	1653, 1660, 1667	\SplitArgument	85
\nouppercase	45	\ssf@size	888
\NR@getttitle	3088	\strutbox	112
\p@	1932, 1975, 2640, 2748, 2749, 2768, 2778, 2787, 2793, 2806, 3243, 3453, 3481, 3482, 3483, 3484, 3491, 3878, 3879, 3880, 3886, 3887, 3888, 3890, 3891, 3897, 3898, 3899, 3901, 3902, 3917, 3918, 3919, 3925, 3926, 3927, 3929, 3930, 3936, 3937, 3938, 3940, 3941	\subparagraph	9, 20, 36
\pagenumbering	26	\subsectionname	15
\pagestyle	10	\subsectionpage	14, 101
\paragraph	9, 20, 36	\subst@correction	802, 809
\parindent	10, 31, 79, 81, 112	\tablename	12
\parskip	86	\textfont@name	885, 893
\part	14, 20, 21, 22, 86, 93	\tf@size	884
\partmark	86	\thechapter	13, 16, 24
\partname	15	\thesection	13, 16, 24, 100
\partpage	14, 101	\titleformat	96
\PassOptionsToClass	43	\titlespacing	96
\PassOptionsToPackage	72	\today	11, 11
\pdfmapline	118, 121	\trans@languagepath	3735, 3761
\pdfstringdef	73	\ttfamily	48
\pickup@font	50	\ttl@a	3170, 3178
\pickup@jfont	779	\ttl@chapterout	3165
		\ttl@extract	97
		\ttl@setifthe	3224
		\ttl@setsubmark	98, 3197, 3203
		\ttl@settopmark	98, 3191, 3222
		\ttl@tocpart	3167, 3169

- \ttlh@hang 3131, 3141
- \ttlh@runin 3130, 3140
- \uppercase 45, 46
- \use@mathgroup 899, 930
- \usepackage 30, 31
- \verbatim@font 1629, 1633
- \wrong@fontshape 51
- \x@protect 5235
- \xdef 99
- \XeTeXglyphbounds 129
- \xkanjiskip 77
- \yahei 8
- \youyuan 8
- \z@ 704, 707, 2594, 2655,
2708, 2772, 2797, 2817, 2860, 2897, 2915, 2929, 3150,
3271, 3294, 3302, 3313, 3879, 3887, 3898, 3918, 3926, 3937
- \zhdig 82
- \zhdigits 26
- \zhnum 82
- \zhnumber 26, 26
- \zihao 25
- \ziju 25
- \zw 76
- tex commands:
- \tex_def:D 5311
- \tex_dimexpr:D 117, 2188
- \tex_divide:D 80, 2189, 2230
- \tex_edef:D 5365
- \tex_endlinechar:D 538, 544
- \tex_font:D 463, 722, 4942
- \tex_glueexpr:D 2180
- \tex_iffontchar:D 722
- \tex_iftrue:D 3687
- \tex_ignorespaces:D
..... 1056, 1215, 1220, 2082, 2214, 3778, 5124
- \tex_immediate:D 460
- \tex_kanjiskip:D 1939, 1953, 1967
- \tex_luafunction:D 2068
- \tex_pagegoal:D 2493
- \tex_parskip:D 2501
- \tex_pdffontattr:D 463
- \tex_pdflastobj:D 464
- \tex_pdfobj:D 460
- \tex_prevdepth:D 2500
- \tex_topskip:D 2494
- \tex_XeTeXcharglyph:D 4946
- \tex_XeTeXglyphbounds:D 4970
- \tex_xkanjiskip:D 2026, 2030, 2040
- \the 4612, 4615, 4616, 4635, 4639
- \thechapter 2724, 2727, 2778, 3287, 3310, 3488
- \theparagraph 3591
- \thepart 2580, 2598, 2620, 2640, 3427
- \thesection 3271, 3278, 3294, 3302, 3313, 3508
- \thesubparagraph 3605
- \thesubsection 3274, 3305, 3540
- \thesubsubsection 3574
- \thispagestyle 2560, 2561, 2694, 2705, 2707
- \tiny 3907, 3946
- tl commands:
- \c_space_tl 464
- \tl_clear:N 188, 384, 1570,
1574, 1868, 1879, 2019, 2135, 2458, 2476, 2534, 3833, 5319
- \tl_clear_new:N 58, 361
- \tl_const:Nn
... 3, 404, 409, 414, 419, 511, 649, 671, 947, 962, 1258,
1259, 1260, 1261, 1320, 2154, 2377, 2383, 3669, 3688, 3690
- \tl_gput_right:Nn 441, 5118, 5120, 5142
- \tl_gset:Nn 555, 558, 884,
886, 888, 1294, 1740, 1768, 1786, 1811, 1818, 1819, 4024
- \tl_gset_eq:NN 1038, 1763
- \tl_gset_rescan:Nnn 1301
- \tl_head:N 3837
- \tl_if_blank:nTF 1387, 1483, 1484, 1588, 1589, 1603, 1621
- \tl_if_blank_p:n 1353
- \tl_if_empty:NTF 742, 2207, 2389, 3401, 3987
- \tl_if_empty:nTF 121, 1702, 1715, 5282
- \tl_if_eq:NNTF 1736, 2147, 3748
- \tl_if_exist:NTF 84, 510, 512, 513, 514, 1257,
1259, 1260, 1261, 2303, 2334, 2345, 2348, 3681, 3771, 5141
- \tl_if_in:NnTF 5346
- \tl_map_inline:Nn 1499
- \tl_map_inline:nn 3834
- \tl_new:N 122, 126, 512, 513, 514,
744, 745, 833, 944, 997, 1023, 1044, 1045, 1066, 1435,
1576, 1756, 2033, 2155, 2168, 2305, 2375, 2376, 3023,
3024, 4181, 5131, 5132, 5141, 5202, 5203, 5322, 5323, 5324
- \tl_put_left:Nn 1992, 5151
- \tl_put_right:Nn
..... 368, 890, 1583, 1616, 2405, 2447, 2449, 3837, 3838
- \tl_replace_all:Nnn 1350, 1391
- \tl_rescan:nn 137, 5364, 5373
- \tl_set:Nn 112, 123,
183, 216, 217, 459, 491, 492, 493, 551, 733, 735, 746,
785, 834, 941, 942, 983, 1049, 1061, 1203, 1264, 1349,
1388, 1390, 1431, 1604, 1608, 1646, 1647, 1648, 1649,
1650, 1758, 1988, 2018, 2034, 2096, 2105, 2131, 2169,
2217, 2306, 2456, 2459, 2463, 3127, 3146, 3165, 3170,
3178, 3747, 3965, 4182, 4186, 4229, 4230, 4231, 5196, 5197
- \tl_set_eq:NN
751, 752, 799, 885, 887, 956, 1050, 1075, 1538, 1550, 2156
- \tl_set_rescan:Nnn 5320
- \tl_tail:N 3132
- \tl_tail:n 3142
- \tl_to_str:n 120, 5263, 5292, 5333, 5334
- \tl_trim_spaces:n 1567
- \today 2281, 2286, 2289, 2294
- today 11, 2281
- token commands:
- \token_to_meaning:N 5277
- \token_to_str:N
... 54, 866, 875, 884, 886, 888, 1103, 1104, 1105, 1107,
1114, 1115, 1116, 1178, 2331, 2332, 3786, 4949, 5109, 5181

