

CT_EX 宏集手册

CTEX.ORG

2019/04/07 v2.4.15*

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. cb8bf3c`.

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 CTeX 宏集 的组成

为了适应用户不同的需求,我们将 CTeX 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 CTeX 宏集的组成

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

2.2 CTeX 宏集 的安装和更新

CTeX 宏集依赖的宏包和宏集已被最常见的 TeX 发行版 TeX Live 和 MiKTeX 所收录。如果本地安装 TeX Live 或 MiKTeX 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。

TeX 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 CTeX 宏集⁵。

MiKTeX 的宏包管理器是 `mpm` (MiKTeX Package Manager)。用户可以在开始菜单找到 MiKTeX → 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 CTeX 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 TeX 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。

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

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

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

4.2 中文编码

GBK ☆
UTF8 ☆

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

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

4.3 中文字库

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

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

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

1 Windows Vista 及以后的 Windows 操作系统。

2 Windows XP 及以前的 Windows 操作系统。

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

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

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

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

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

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

zhmap	☆	zhmap = <code><true false zhmCJK></code>
-------	---	--

Updated: 2014-03-08

指定字体映射机制。本选项只在使用 pdf $\LaTeX$ / $\LaTeX$ 编译时有意义。

true 使用 zhmetrics 宏包, 将 CJK 字库通过 \special 命令映射到 .ttf 文件。

false 使用传统的 CJK 字库 (Type 1)⁹。zhmCJK 载入 zhmCJK 宏包¹⁰, 由 zhmCJK 宏包提供从 CJK 字库到 .ttf 的映射。

fontset	fontset =
---------	-----------

New: 2014-03-08

`<none|adobe|fandol|founder|mac|macnew|macold|ubuntu|windows|windowsnew|windowsold|...>`指定 CT $\TeX$ 宏集加载的字库。如果没有指定 fontset 的值, CT $\TeX$ 宏集将自动检测用户使用的操作系统, 配置相应的字体 (参见表 3)。CT $\TeX$ 预定义了以下六种中文字库。adobe 使用 Adobe 公司的四款中文字体, 不支持 pdf $\LaTeX$ 。fandol 使用 Fandol 中文字体, 不支持 pdf $\LaTeX$ 。

founder 使用方正公司的中文字体。

mac 使用 Mac OS X 系统下的字体, 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 $\TeX$ 预定义的中文字库, 可以设置 fontset 为下述值之一。

none 不配置中文字体, 需要用户自己配置。

`<name>` 这里 `<name>` 为自定义的名字。CT $\TeX$ 宏集将载入名为 `ctex-fontset-<name>.def` 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录树下合适位置建立一个名为 `ctex-fontset-<name>.def` 的文件, 在这个文件里面自定义中文字体。然后通过使用 `fontset=<name>` 选项来调用它。字体配置文件的具体写法可以参考 CT $\TeX$ 宏集 fontset 目录下的字体配置文件。注意: 如果希望使用 \ctexset 在导言区指定字库, 则需要先在宏包/文档类选项中指定 `fontset = none`。例如:

例 5

```

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

```

⁹如果需要使用自定义的字体映射文件, 或者希望使用 Type1 字库, 请禁用本选项。¹⁰zhmCJK 宏包基于 zhmetrics 和 CJK 宏包, 提供与 xeCJK 宏包类似的用户接口。¹¹可以使用 zhmetrics 宏包提供的脚本 CTeXFonts.lua。

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

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

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

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

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

`\yahei` 微软雅黑,CJK 等价命令 `\CJKfamily{zh yahei}`。
在 macnew 字库中,还定义了 `\pingfang`。
`\pingfang` 苹方黑体,CJK 等价命令 `\CJKfamily{zh pf}`。

第 5 节 排版格式设定

5.1 文档默认字号

zihao

☆

New: 2015-05-06

`zihao = <-4|5|false>`
将文章默认字号(`\normalsize`)设置为小四号字或五号字,具体情况见表 4。false 禁用本功能。本选项可以用于四个 CT_EX 文档类和 `ctex` 宏包,也可以用于 `ctexsize` 宏包。
`scheme = chinese` 时,对标准文档类默认值为 5,即设置 `\normalsize` 为五号字;对 `beamer` 则为 false,使用文档类原有的设置。

表 4 标准字体命令与字号的对应

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

10pt ☆
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,也可能影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中,那么该选项将会被自动激活。

¹³使用 CT_EX 文档类,或者使用 ctex 宏包且开启该选项时。

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

`punct` `punct = \langle \text{quanjiao} | \text{banjiao} | \text{kaiming} | \text{CCT} | \text{plain} \rangle`

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

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

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

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

`true` 总是保留汉字后的空格。此时, 用户需要自行在行尾加上 `%` 处理换行产生的空格¹⁵。
`false` 总是忽略掉汉字后面的空格, 不论汉字后是什么(使用 $(\text{pdf})\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译时); 等同于 `auto` 的效果(使用 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译时)。不建议使用该选项。
`auto` 根据空格后面的情况决定是否保留: 如果空格后面是汉字, 则忽略该空格, 否则保留。

例如, 使用

例 6

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

将得到“汉字 分词 技术 English”; 使用

例 7

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

则会得到“汉字分词技术 English”。

使用 $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 及 $\text{upL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译的时候, 该选项无效: 汉字间的空格以及汉字与西文字符之间的空格总是有效, 不会被忽略, 但可以自动忽略掉由换行产生的空格。

`linespread` ☆ `linespread = \langle \text{数值} \rangle`

New: 2014-04-23

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

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

`scheme = plain` $\text{CT}_{\text{E}}\text{X}$ 宏集默认不调整行距倍数, 文档中的行距由所选文档类和其他宏包或用户设置决定。

`autoindent` `autoindent = \langle \text{true} | \text{false} | \text{数值} | \text{带单位的数值} \rangle`

New: 2014-03-13

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

¹⁴标点符号分为标号与点号。点号分为两类, 一共七种: 句末点号有句号、问号和叹号; 句内点号有逗号、顿号、冒号和分号。

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

〈数值或带单位的数值〉 用于设置段首缩进的长度。如果不带单位,则默认单位是单个汉字字宽 `\ccwd`; 如果带单位,则使用该单位。

`true` 等价于设置 `autoindent = 2`。

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

例 8

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

`linestretch` ★ `linestretch = 〈数值或长度〉`

New: 2014-03-26

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

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

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

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

第 6 节 文档汉化

6.1 日期汉化

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 4 月 7 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 4 月 7 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇一九年四月七日”。使用全汉字的日期格式。

`old` 效果为“April 7, 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。

contentsname *	contentsname = <名字>
	设置目录标题名 \contentsname。中文默认为“目录”。
listfigurename *	listfigurename = <名字>
	设置插图目录标题名 \listfigurename。中文默认为“插图”。
listtablename *	listtablename = <名字>
	设置表格目录标题名 \listtablename。中文默认为“表格”。
figurename *	figurename = <名字>
	设置图片环境标题名 \figurename。中文默认为“图”。
tablename *	tablename = <名字>
	设置表格环境标题名 \tablename。中文默认为“表”。
abstractname *	abstractname = <名字>
	设置摘要 abstract 环境标题名 \abstractname。中文默认为“摘要”。注意 book 类没有摘要,该选项无效。
indexname *	indexname = <名字>
	设置索引标题名 \indexname。中文默认为“索引”。
appendixname *	appendixname = <名字>
	设置附录标题名 \appendixname。中文默认为“附录”。
bibname *	bibname = <名字>
	设置参考文献标题名 \refname(对 article)或 \bibname(对 report、book 和 beamer)。中文默认为“参考文献”。
proofname *	proofname = <名字>
	设置证明环境的名称 \proofname。中文默认为“证明”。
	如果使用 ctexbeamer 文档类或者在 beamer 文档类下使用 ctex 包,还会汉化常用定理类环境的诸如“定义”、“定理”和“引理”等名称。此时,还有下列三个选项。
refname *	refname = <名字>
	设置参考文献标题名 \refname。中文默认为“参考文献”。
algorithmname *	algorithmname = <名字>
	设置算法环境标题名 \algorithmname。中文默认为“算法”。
continuation *	continuation = <名字>
	设置 beamer 可断页的帧在续页标题中的延续标识 \insertcontinuationtext。中文默认为“(续)”。

6.3 页面格式设置与汉化

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, lot skip, lof skip, indent, hang, page style, 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 page style 选项,chapter 一级标题的 format 与 page style 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， $\text{\LaTeX}$ 带星号的章节标题命令（如 $\text{\section*}$ ）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 $\text{\LaTeX}$ 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 $\text{\chapter}$ 标题使用形如“第一章”的名字，而 $\text{\section}$ 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 $\text{\partname}$ 为 Part
chapter	{第, 章}	{\chaptername\space}	原 $\text{\chaptername}$ 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 $\text{\sectionname}$ 为 $\text{\translate{Section}}$
section	同右	{}	
subsection (beamer)	{}	{\subsectionname\space}	原 $\text{\subsectionname}$ 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      *
subsection/runin   *
subsubsection/runin *
paragraph/runin    *
subparagraph/runin *
```

New: 2015-06-27

runin = true|false

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

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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

```
.../afterindent *
```

New: 2015-06-27

afterindent = true|false

afterindent 选项用于设置章节标题后首段的缩进。

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

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

表 13 afterindent 选项的默认设置

标题名	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 *
```

Updated: 2016-05-10

beforeskip = {<弹性间距>}

beforeskip 选项用于设置章节标题前的垂直间距。

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

表 14 before skip 选项的默认设置

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

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}}{}#2`

```
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 $\text{\LaTeX}$ 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 $\text{\TeX}$ 宏，不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 $\langle \text{number} \rangle$ 转为中文数字，最后将结果存储在 $\backslash \langle \text{macro} \rangle$ 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 $\text{\TeX}$ 社区的北川弘典、前田一贵、八登崇之等人开发的，设计目的主要是在 $\text{Lua}\text{\TeX}$ 引擎下实现日本 $\text{p}\text{\TeX}$ 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 $\text{\LaTeX}$ 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>} [AlternateFont = { {<character range ₁ >} [<alternate font features ₁ >] {<alternate font name ₁ >} {<character range ₂ >} [<alternate font features ₂ >] {<alternate font name ₂ >} }, <base font features>] {<base font name>}
---------------	--

New: 2014-04-14

在设置字体族 *<family>* 的时候,同时设置该字体族在字符范围 *<character range_n>* 内,对应字形的替代字体。

CharRange	\setCJKfamilyfont {<family>} [CharRange = {<character range>} , <alternate font features>] {<alternate font name>}
-----------	--

New: 2014-04-14

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

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

例 22

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

declarecharrange *	\ctexset { declarecharrange = { {<name ₁ >} {<character range ₁ >} , {<name ₂ >} {<character range ₂ >} , ... } }
--------------------	---

New: 2014-04-14

预先声明字符范围。声明字符范围 *<name>* 之后,它的名字 *<name>* 可以用在 AlternateFont 和 CharRange 选项的 *<character range>* 之中,表示对应的字符范围。

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

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

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

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

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

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

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

10.1 修改宏包默认选项

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

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

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

10.2 宏包载入后的配置

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

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

11.1 C_T_EX 0.8a 及以前的版本

在 `ctex-kit` 项目成立之前, C_T_EX 宏包的最后一个版本是 C_T_EX 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。

cs4size c5size	分别相当于 zihao=-4 和 zihao=5, 过时选项。
CCT CCTfont	相关选项已删除。
indent noindent	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。
zhmap nozhmap	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。
winfonts adobefonts nofonts	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)。
punct nopunct	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。
cap nocap	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。
space nospace	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。
fancyhdr	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 CT _E X 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 CT _E X 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 CT _E X 宏集中已失去意义。 此外, 在 pdf _T E _X 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。
captiondelimit	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 CT _E X 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[captiondelimit=:]
\usepackage{caption}
\captionsetup{labelsep=colon}
```

11.3 CT_EX 1.02c 以后的 SVN 开发版

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

CT_EX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 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 zhm-CJK 是可选的。
- xeCJK 宏集, 它还依赖
 - xtemplate 宏包, 它属于 l3packages 宏集。
 - fontspec 宏包。
- environ 宏包, 它还依赖 trimspaces 宏包。
- ➡ 以上是使用 X_YL_AT_EX 编译时的依赖项。

- luatexja 宏包, 它还依赖
 - adobemapping 宏包。
 - lualibs 宏包。
 - luaotfload 宏包。
 - luatexbase 宏包, 它还依赖 ctablestack 宏包。
 - oberdiek 宏集。
 - xkeyval 宏包。
 - etoolbox 宏包。
- fontspec 宏包。
- xunicode-addon 宏包, 属于 xeCJK 宏集, 它还依赖
 - xunicode 宏包, 它还依赖
 - * graphics 宏集。
 - * graphics-cfg 宏包。
 - * graphics-def 宏包。
- ➡ 以上是使用 Lua $\text{\LaTeX}$ 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 $\text{\LaTeX}$ 编译时的依赖项。

出于一些原因, zhmCJK 尚未被收入 $\text{\TeX}$ Live 和 MiK $\text{\TeX}$ 。因此, 若你希望使用 zhmCJK 作为 $\text{\CTEX}$ 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 $\text{\TeX}$ 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 $\text{\TeX}$ 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。

$\text{\CTEX}$ 宏集已被 $\text{\TeX}$ Live 和 MiK $\text{\TeX}$ 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 $\text{\CTEX}$ 宏集的 **TDS 安装包**,
2. 按目录结构将文件复制到 $\text{\TeX}$ 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 $\text{\TeX}$ 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:nnx { ctex } { engine-not-supported } { \c__ctex_engine_str } }

54 </class|ctex>

55 <*class|ctex|ctexheading>

```

56 \RequirePackage { ctexhook , ctexpatch }
<!\ctexheading> 57 \RequirePackage { fix-cm , 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:Npn \ctex_set_default_ccwd:Nn #1#2
112 {
113   \dim_compare:nNnTF
114     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
115     { \tl_set:Nn #1 {#2} }
116     { \tl_set:Nn #1 { #2 \ccwd } }
117 }
```

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

```
118 \cs_new:Npn \ctex_default_pt:n #1
119 {
120   \exp_after:wN \__ctex_default_pt:w
121   \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop
122 }
123 \use:x
124 {
125   \cs_new:Npn \exp_not:N \__ctex_default_pt:w
126     ##1 \tl_to_str:n { pt } ##2 \exp_not:N \q_stop
127     { ##1 \tl_to_str:n { pt } }
128 }
```

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

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

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

```
132 \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。

```
133 \tl_new:N \l__ctex_autoindent_tl
```

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

```
134 \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。

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

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

```
<class> 136 \cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
<ctex> 137 \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。

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

对无效选项给出警告。

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

对过时选项或命令给出警告。

```
145 \msg_new:nnn { ctex } { deprecated-option }
146 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
147 \msg_new:nnn { ctex } { deprecated-command }
148 { Command~ #1 is~ deprecated.\\ #2 }
149 \msg_new:nnn { ctex } { deprecated-environment }
150 { Environment~ `#1'~ is~ deprecated.\\ #2 }
151 </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。

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

14.2 宏包选项

```
156 <*class|style>
157 \keys_define:nn { ctex / option }
158 {
159 </class|style>
160 <*class|ctex|ctexsize>
161   zihao .choice: ,
162   zihao .value_required:n = true ,
163   zihao / 5 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 0 } } ,
164   zihao / -4 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 1 } } ,
165   zihao / false .code:n = { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
<ctexsize> 166 }
167 </class|ctex|ctexsize>
168 <*class|ctex>
169   c5size .code:n =
170   {
171     \msg_warning:nnn { ctex } { deprecated-option }
172     { Option~`zihao=5'~ is~ set. }
173     \keys_set:nn { ctex / option } { zihao = 5 }
174   } ,
175   cs4size .code:n =
176   {
177     \msg_warning:nnn { ctex } { deprecated-option }
```

```

178     { Option~ `zihao=-4'~ is~ set. }
179     \keys_set:nn { ctex / option } { zihao = -4 }
180   } ,
181   c5size .value_forbidden:n = true ,
182   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。

```

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

```

autoindent 自动调整段落的首行缩进功能。

```

186   autoindent .choice: ,
187   autoindent .default:n = { true } ,
188   autoindent / true .code:n =
189   {
190     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
191     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
192   } ,
193   autoindent / false .code:n =
194   {
195     \tl_clear:N \l__ctex_autoindent_tl
196     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
197   } ,
198   autoindent / unknown .code:n =
199   {
200     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
201     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
202   } ,

```

indent 仅为兼容性保留, 已过时。

```

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

```

```

GBK 223   GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
UTF8 224   UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
225   GBK .value_forbidden:n = true ,
226   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。

```

227   fontset .tl_gset:N = \g__ctex_fontset_tl ,

```

```

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

```

```

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

```

punct 设置标点符号输出格式。

```

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

```

```

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

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

heading 306   heading .bool_set:N = \l__ctex_heading_bool ,

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

sub3section 309 <!*beamer>
sub4section 310   sub3section .code:n =
311     { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
312   sub4section .code:n =
313     { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
314   sub3section .value_forbidden:n = true ,
315   sub4section .value_forbidden:n = true ,
316 </!beamer>

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

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

329   cap .code:n =
330     {
331       \msg_warning:nnn { ctex } { deprecated-option }
332       { Option~`scheme = chinese' ~ is~ set. }
333       \keys_set:nn { ctex / option } { scheme = chinese }
334     } ,
335   nocap .code:n =
336     {
337       \msg_warning:nnn { ctex } { deprecated-option }
338       { Option~`scheme = plain' ~ is~ set. }
339       \keys_set:nn { ctex / option } { scheme = plain }
340     } ,
341   cap .value_forbidden:n = true ,
342   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 343   fntef .code:n =

```

```

344     {
345       \msg_warning:nnn { ctex } { deprecated-option }
346       { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
347     } ,
348     fancyhdr .code:n =
349     {
350       \msg_warning:nnn { ctex } { deprecated-option }
351       { `fancyhdr'~ package~ is~ loaded. }
352       \RequirePackage { fancyhdr }
353     } ,
354     hyperref .code:n =
355     {
356       \msg_warning:nnn { ctex } { deprecated-option }
357       { `hyperref'~ package~ will~ be~ loaded. }
358       \ctex_at_end:n
359       {
360         \cs_if_exist:NF \hypersetup
361         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
362       }
363       \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
364     } ,
365   }

366 </class>ctex>
367 <*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

```

368 \tl_clear_new:N \l__ctex_tmp_tl
369 \clist_map_inline:nn
370 {
371   10pt , 11pt , 12pt ,
372   8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
373 }
374 {
375   \tl_put_right:Nn \l__ctex_tmp_tl
376   {
377     #1 .code:n =
378     <*!class>
379     { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
380     </!class>
381     <*class>
382     {
383       \int_gset:Nn \g__ctex_font_size_int { 2 }
384       \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
385     } ,
386     </class>
387     #1 .value_forbidden:n = true ,
388   }
389 }
390 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
391 \tl_clear:N \l__ctex_tmp_tl

```

将未知选项传给标准文档类。

```

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

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

```

```

401 <*class|style>
402 \ProcessKeysOptions { ctex / option }
403 </class|style>
404 <*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。

```

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

```

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

```

410 <*article>
411 \tl_const:Nn \c__ctex_class_tl { article }
412 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
413 \LoadClass { article }
414 </article>
415 <*book>
416 \tl_const:Nn \c__ctex_class_tl { book }
417 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
418 \LoadClass { book }
419 </book>
420 <*report>
421 \tl_const:Nn \c__ctex_class_tl { report }
422 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
423 \LoadClass { report }
424 </report>
425 <*beamer>
426 \tl_const:Nn \c__ctex_class_tl { beamer }
427 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
428 \LoadClass { beamer }
429 </beamer>
430 </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。

```

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

```

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

```

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

```

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

```

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

```

```

444 {
445   \cs_if_free:cF { #1 + #2 }
446   {
447     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
448     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
449   }
450 }
451 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
452 \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。

```

453 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
454 {
455   \cs_if_free:NF \CJK@plane
456   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
457 }
458 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
459 {
460   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
461   #1
462 }
463 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
464 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
465 {
466   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
467   \tex_immediate:D \tex_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
468   \cs_new_protected_nopar:Npx #1
469   {
470     \exp_not:N \tex_pdffontattr:D \exp_not:N \tex_font:D
471     { /ToUnicode ~ \int_use:N \tex_pdflastobj:D \c_space_tl 0 ~ R }
472   }
473 }

```

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

```

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

```

首先检查选项,决定是否载入 `zhmCJK` 宏包。

```

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

```

不载入 `zhmCJK` 宏包时直接调用 `CJK` 及相关宏包。

```

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

```

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

```

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

```

```

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

```

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

```

504 \cs_new_protected_nopar:Npn \ctex_zhmap_input:n #1
505 {
506   \ctex_file_wrapper:nnn
507   {
508     \char_set_catcode_comment:n { 37 } % %
509     \char_set_catcode_escape:n { 92 } % \
510   }
511   { \file_input:n {#1} }
512   {
513     \char_set_catcode:nn { 37 } { \char_value_catcode:n { 37 } }
514     \char_set_catcode:nn { 92 } { \char_value_catcode:n { 92 } }
515   }
516 }

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

```

zhmCJK 判断结束。

```
538 \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。

```

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

```

```

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

```

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

```

555 \if_cs_exist:N \CJK@surr
556   \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
557   {
558     \tl_set:Nx \l__ctex_tmp_tl {#2}
559     \int_set:Nn \l__ctex_tmp_int
560     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
561     \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
562     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
563     {
564       \int_sub:Nn \l__ctex_tmp_int { 256 }
565       \tl_gset:Nx #1
566       {
567         \int_to_Hex:n
568         { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
569         \int_to_Hex:n
570         { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
571       }
572     }
573   }
574   \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
575 \fi:

```

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

```

576 \AtBeginDocument
577 {
578   \str_if_eq:eeF { \l__ctex_punct_tl } { quanjiao }
579   { \punctstyle { \l__ctex_punct_tl } }
580 }

```

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

```

581 \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。

```

582 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
583 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
584 {
585   \CJK@makeActive
586   \CJK@loadBinding { standard }
587 }

```

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%
}

```

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

```

\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。

```

588 \exp_args:Nx \ctex_at_end_preamble:n
589 {
590   \exp_not:N \CJK@envStart
591   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
592   \exp_not:N \CJKtilde
593 }

```

zhmCJK 判断结束。

```
594 \fi:
```

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

```

595 <@@=>
596 \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。

```
597 <@@=-ctex>
```

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

```

598 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
599 {
600   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
601   { \cs_set_eq:NN \CJK@ignorespaces #1 }
602   \ctex_set_ignorespaces:
603 }
604 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
605 { \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。

```
606 \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。

```

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

```

```

612     {
613         \cs_set_eq:cc
614         { CJKpunct@ ##1 @spaces }
615         { c__ctex_ #1 ##1 _punct_spaces_tl }
616     }
617 }
618 }
619 \clist_const:Nn \c__ctex_punct_family_clist
620 {
621     zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
622     zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
623 }

```

\ctex_punct_map_family:nn

CJK 族 #1 使用族 #2 的边界信息。

```

624 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
625 {
626     \cs_if_free:cF { CJKpunct@ #2 @spaces }
627     { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
628 }

```

\ctex_punct_map_bfseries:nn

CJK 族 #1 的 \bfseries 使用族 #2 的边界信息。

```

629 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
630 {
631     \clist_map_inline:nn {#1}
632     {
633         \ctex_punct_map_series:nnn { ##1 } { b } {#2}
634         \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
635     }
636 }
637 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
638 {
639     \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
640     \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
641     \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
642     \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
643     \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
644     \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
645 }

```

\ctex_punct_map_itshape:nn

CJK 族 #1 的 \itshape 使用族 #2 的边界信息。

```

646 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
647 {
648     \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
649     \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
650     \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
651     \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
652     \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
653     \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
654 }

```

\ctex_punct_space:nn

定义标点的边界信息。

\ctexspadef

```

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

```

载入边界信息文件。

```

658 \ctex_file_input:n { ctexspa.def }

```

```

659 </pdfTeX>

```

14.3.2 ctex-engine-xetex.def

```

660 <*xetex>

661 \RequirePackage { xeCJK }
662 \exp_args:Nx \xeCJKsetup
663 {
664     LoadFandol    = false ,
665     PunctStyle     = \l__ctex_punct_tl
666 }

```

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。

```

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

```

14.3.3 ctex-engine-luatex.def

```

670 <*luatex>

```

LuaTeX-jā 为了兼容 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。我们在这里禁用它。

```

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

```

14.3.3.1 LuaTeX-jā 的默认设置

```

685 \ExplSyntaxOff

```

以下设置抄录自 lltjdefs.sty, 略有改动。

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

```

686 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
687 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
688 \ltjdefcharrange{3}{%
689     "2000-"206F, "2070-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF,
690     "2B00-"2BFF}
691 \ltjdefcharrange{4}{%
692     "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
693     "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
694     "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
695 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
696 \ltjdefcharrange{6}{%
697     "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
698     "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
699 \ltjdefcharrange{7}{%
700     "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF,
701     "A830-"A83F, "AC00-"D7FF}
702 \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。

```

703 \ltjdefcharrange{9}{%
704   "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}

```

LuaTeX-j_a 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。

```

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

```

以下设置抄录自 ltj-latex.sty。

```

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

```

14.3.3.2 LuaTeX-j_a 的补丁

```

718 <@@=ctex_ltxj>

```

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

```

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

```

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

```

727 \cs_set_protected:Npn \fontspec_visible_space:
728 {
729   \tex_iffontchar:D \tex_font:D "2423 \exp_stop_f:
730   \ltjalchar "2423 \exp_stop_f:
731   \else:
732     \fontspec_visible_space_fallback:
733   \fi:
734 }

```

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

```

735 \ctex_at_end_package:nn { listings }
736 {
737   \use:x
738   {
739     \exp_not:N \RequirePackage { lltpj-listings }
740     \tl_set:Nn \exp_not:N \lstlistingname
741       { \exp_not:o { \lstlistingname } }
742     \tl_set:Nn \exp_not:N \lstlistlistingname
743       { \exp_not:o { \lstlistlistingname } }
744   }
745 }

```

14.3.3.3 字体切换方式

`\ctex_ljt_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。

```

746 \cs_new_protected_nopar:Npn \ctex_ljt_select_font:
747 {
748   \cs_if_exist_use:cF { \l__ctex_ljt_current_font_tl }
749   { \tl_if_empty:NF \CJK@family { \__ctex_ljt_select_font_aux: } }
750 }
751 \tl_new:N \CJK@family
752 \tl_new:N \l__ctex_ljt_current_font_tl
753 \tl_set:Nn \l__ctex_ljt_current_font_tl
754 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

`\__ctex_ljt_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。

```

755 \cs_new_protected_nopar:Npn \__ctex_ljt_select_font_aux:
756 {
757   \group_begin:
758   \tl_set_eq:NN \f@encoding \CJK@encoding
759   \tl_set_eq:NN \f@family \CJK@family
760   \__ctex_ljt_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
761   \ctex_ljt_pickup_font:
762   \group_end:
763   \font@name
764   \__ctex_ljt_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_ljt_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。

```

765   \cs_if_exist:cF { \l__ctex_ljt_current_font_tl }
766   { \__ctex_ljt_select_font_aux: }
767 }
768 \cs_new_protected_nopar:Npn \__ctex_ljt_push_fontname:n #1
769 {
770   \cs_gset_eq:NN \__ctex_ljt_save_fontname:w \font@name
771   \cs_gset_nopar:Npx \font@name {#1}
772 }
773 \cs_new_protected_nopar:Npn \__ctex_ljt_pop_fontname:
774 { \cs_gset_eq:NN \font@name \__ctex_ljt_save_fontname:w }

```

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

```

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

```

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

```

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

```
787 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
788 {
789   \get@external@font
790   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
791   {
792     \tl_set:Nx \external@font
793     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
794   }
795   \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。

```
796   \font@name
797   \lua_now:e { font.current(tex.getattribute('ltj@curjfont')) }
798   \use:c { \f@encoding + \f@family }
799   \use:c { \curr@fontshape }
800 }
```

\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。

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

```
820 }
```

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

```
821 \prg_new_conditional:Npnn \ctex_ltj_if_alter_nate_shape_exist:n #1 { T , F , TF }
822 {
823   \lua_now:e { luatexja.jfont.does_alt_set ('lua_escape:e {#1}') }
824   \prg_return_true: \else: \prg_return_false: \fi:
825 }
```

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。

```
826 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
827 { #1 \lua_now:e { luatexja.jfont.print_aftl_address() } ~ at }
```

在 `\selectfont` 中更新替代字体。

```
828 \cs_new_protected_nopar:Npn \ctex_ltj_select_alter_nate_font:
829 {
830   \ctex_ltj_if_alter_nate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
831   {
832     \lua_now:e
833     {
834       luatexja.jfont.output_alt_font_cmd
835       ('y', 'lua_escape:e { \l__ctex_ltj_current_shape_tl }')
836     }
837     \lua_now:e { luatexja.jfont.pickup_alt_font_a ('f@size') }
838   }
839 }
840 \tl_new:N \l__ctex_ltj_current_shape_tl
841 \tl_set:Nn \l__ctex_ltj_current_shape_tl
842 { \CJK@encoding / \CJK@family / \f@series / \f@shape }
```

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

```
843 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
844 {
845   \cs_if_exist:cF { #1/f@size }
846   {
847     \group_begin:
848     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
849     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
850     \ctex_ltj_pickup_font:
851     \group_end:
852     \__ctex_ltj_pop_fontname:
853   }
854 }
855 <@@=>
```

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。

```
856 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
857 {
858   \ltj@@getjfontnumber #1
859   \lua_now:e
860   {
```

```

861         luatexja.jfont.pickup_alt_font_b
862         ( \int_use:N \ltj@tempcntc, '\lua_escape:e {#2}' )
863     }
864 }

```

14.3.3.4 数学字体族

865 `<@@-ctex_ltj>`

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

`\ctex_ltj_if_jfont:nTF` 参数 #1 是一个 $\text{\LaTeX} 2_{\epsilon}$ 编码名称或者字体命令。 $\text{\LaTeX} 2_{\epsilon}$ 字体命令的一般形式是:

`\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle`

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

```

866 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
867 {
868     \lua_now:e
869     { luatexja.jfont.is_kenc( string.match('\lua_escape:e {#1}', '[^/]+') ) }
870     \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
871 }

```

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

```

872 \cs_new:Npn \ctex_ltj_if_jfont_math:nTF #1
873 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
874 \group_begin:
875     \char_set_catcode_other:N M
876     \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
877     { \ctex_ltj_if_jfont:nTF {#3} }
878 \group_end:
879 <@@=>

```

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

`\ctex_ltj_get_and_define_fonts:nN`

```

880 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
881 {
882     \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
883     { \ctex_ltj_get_and_define_fonts_ja:nN }
884     { \ctex_ltj_get_and_define_fonts_al:nN }
885     {#1} #2
886 }
887 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
888 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
889 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
890 {
891     \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
892     \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
893     \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
894     \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
895     \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
896     \ctex_ltj_pickup_font:
897     \tl_put_right:Nx \math@fonts
898     {
899         \ltj@setpar@global
900         \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
901         \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
902         \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
903     }
904 }

```

905 <@@=ctex_ltj>

\use@mathgroup
\ctex_ltj_use_math_group:Nn

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

```

906 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
907 {
908   \mode_if_math:T
909   {
910     \math@bgroup
911     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
912     \ctex_ltj_math_group_hook:
913     \ctex_ltj_if_jfont_math:NTF #1
914     { \jfam } { \mathgroup } #2 \scan_stop:
915     \math@egroup
916   }
917 }
918 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
919 \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。

```

920 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
921 {
922   \group_begin:
923   \cs_set_protected:Npn #1 ##1##2##3
924     { \ltjsetmathletter { ##1 } }
925   #2
926   \group_end:
927 }
928 \ctex_at_end_package:nn { unicode-math }
929 {
930   \cs_if_exist:NTF \um_input_math_symbol_table:
931   {
932     \ctex_ltj_set_math_letter:NN
933     \um_sym:nnn
934     \um_input_math_symbol_table:
935   }
936   {
937     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
938     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
939     { \__um_switchto_literal: }
940     \ctex_ltj_set_math_letter:NN
941     \__um_sym:nnn
942     \__um_input_math_symbol_table:
943   }
944 }

```

14.3.3.5 字体族的定义与使用

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

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

```

945 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
946 {
947   \str_if_eq:nnTF {#1} { plain }
948   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
949   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
950 }
951 \tl_new:N \l__ctex_ltj_jfm_tl
952 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
953 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

\CJK@encoding
__ctex_ltj_change_encoding:

在 $\text{\LaTeX}$ 下, LuaTeX-ja 依赖字体编码来实现特殊设置。例如上述的 `\ctex_ltj_if-jfont:nTF` 就是通过判断编码来实现的, 它在设置数学字体时会用到。所以不应该与

西文共用 EU2。定义字体族 `song` 为 `\CJK@encoding` 的默认替换字体。下划线 `_` 不在 `\nfss@catcodes` 里,可以放心使用。

```

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

```

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

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

```

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

```

979 \keys_define:nn { fontspec-preparse-external }
980 {
981   NoEmbed .code:n =
982   { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
983 }
984 \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。

```

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

```

```

1009 \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1010 }

```

分别保存 **fontspec** 设置的字体族名、字体名称和字体选项。

```

1011 \prop_new:N \g__ctex_ltj_family_name_prop
1012 \prop_new:N \g__ctex_ltj_family_font_name_prop
1013 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

删除重复的定义,清除替代字体的先前设置。

```

1014 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1015 {
1016   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1017   {
1018     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1019     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1020     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1021     {
1022       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1023       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1024       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1025       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1026     }
1027     \msg_warning:nxxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1028   }
1029 }
1030 \tl_new:N \l__ctex_ltj_tmp_tl
1031 \msg_new:nnn { ctex } { redefine-family }
1032 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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

```

1033 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1034 {
1035   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }
1036   {
1037     \group_begin:
1038     \__ctex_ltj_change_encoding:
1039     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1040     \exp_not:n { \cs_set_eq:NN \CTEX@fontfamily \use_none:n }
1041     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1042     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1043     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1044     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1045     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1046     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1047     \__ctex_ltj_set_alternate_family:n {#1}
1048     \group_end:
1049   }
1050 }
1051 \tl_new:N \l__ctex_ltj_base_family_tl
1052 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1053 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1054 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1055 {
1056   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1057   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1058   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1059   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1060 }
1061 \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

\CJKfamily 切换字体。

```

1062 \NewDocumentCommand \CJKfamily { m }

```

```

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

```

`\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。

```

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

```

```

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

```

```

1126 }

\ctex_ltj_fontspec:nn 1127 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1128 {
1129   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1130   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1131   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1132   {
1133     \int_gincr:N \g__ctex_ltj_family_int
1134     \__ctex_ltj_fontspec:xnn
1135     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1136     {#1} {#2}
1137   }
1138 }

1139 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1140 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1141 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1142 {
1143   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1144   {
1145     \cs_if_free:cF
1146     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1147     {
1148       \cs_gset_eq:cc
1149       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1150       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1151       \cs_gset_eq:cc
1152       { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1153       { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1154     }
1155     \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1156   }
1157   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1158   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1159   \ctex_ltj_switch_family:n {#1}
1160 }
1161 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1162 \prop_new:N \g__ctex_ltj_fontspec_prop

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

```

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。

```

\__ctex_ltj_pass_args:nnnn
1188 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1189 {
1190   \IfNoValueTF {#2}
1191   { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1192   {
1193     \use:x { #1 {#2} {#3} }
1194     #4
1195   }
1196 }
1197 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1198 {
1199   \use:x { #1 {#4} {#2} }
1200   #3
1201 }

\setCJKfamilyfont 1202 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 1203 {
  \CJKfontspec 1204   \__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 1205   { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1206   { }
1207 }
1208 \NewDocumentCommand \newCJKfontfamily { o m o m }
1209 {
1210   \tl_set:Nx \l__ctex_ltj_tmp_tl
1211   { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1212   \cs_new_protected_nopar:Npx #2
1213   { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1214   \__ctex_ltj_pass_args:nnnn
1215   { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1216   { }
1217 }
1218 \NewDocumentCommand \CJKfontspec { o m }
1219 {
1220   \__ctex_ltj_pass_args:nnnn
1221   { \ctex_ltj_fontspec:nn } {#1} {#2}
1222   { \tex_ignorespaces:D }
1223 }
1224 \NewDocumentCommand \addCJKfontfeatures { m }
1225 {
1226   \ctex_ltj_add_font_features:x {#1}
1227   \tex_ignorespaces:D
1228 }
1229 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\setCJKmainfont 1230 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1231 {
\setCJKmonofont 1232   \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1233   { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1234   { \normalfont }
1235 }
1236 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1237 \NewDocumentCommand \setCJKsansfont { o m }
1238 {
1239   \__ctex_ltj_pass_args:nnnn
1240   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1241   { \normalfont }
1242 }
1243 \NewDocumentCommand \setCJKmonofont { o m }
1244 {
1245   \__ctex_ltj_pass_args:nnnn
1246   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1247   { \normalfont }
1248 }
1249 \NewDocumentCommand \setCJKmathfont { o m }
1250 {

```

```

1251     \__ctex_ltj_pass_args:nnnn
1252     { \ctex_ltj_set_family:nnn { \__ctex_ltj_math_tl } } {#1} {#2}
1253     { }
1254 }
1255 \NewDocumentCommand \defaultCJKfontfeatures { m }
1256 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1257 \clist_new:N \g__ctex_ltj_default_features_clist
1258 \@onlypreamble \setCJKmainfont
1259 \@onlypreamble \setCJKsansfont
1260 \@onlypreamble \setCJKmonofont
1261 \@onlypreamble \setCJKmathfont
1262 \@onlypreamble \setCJKromanfont
1263 \@onlypreamble \defaultCJKfontfeatures

1264 \tl_if_exist:NF \CJKfamilydefault
1265 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1266 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1267 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1268 \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。

```

1269 \RenewDocumentCommand \fontfamily { m }
1270 {
1271     \tl_set:Nx \f@family {#1}
1272     \CTEX@fontfamily {#1}
1273 }
1274 \cs_new_protected_nopar:Npn \CTEX@fontfamily #1
1275 {
1276     \str_if_eq:nnTF {#1} { \familydefault }
1277     { \CJKfamily { \CJKfamilydefault } }
1278     { \__ctex_ltj_update_family_aux: }
1279 }
1280 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_aux:
1281 {
1282     \str_case_e:nn { \f@family }
1283     {
1284         { \rmdefault }      { \CJKfamily { \CJKrmdefault } }
1285         { \sfdefault }      { \CJKfamily { \CJKsfdefault } }
1286         { \ttdefault }      { \CJKfamily { \CJKttdefault } }
1287         { \familydefault } { \CJKfamily { \CJKfamilydefault } }
1288     }
1289 }

```

在导言区末尾更新 \CJKfamilydefault。

```

1290 \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。

```

1291 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1292 {
1293     \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1294     {
1295         \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1296         {
1297             \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
1298             { \use:n }
1299             {
1300                 \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1301                 { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1302                 { \use:n }
1303             }
1304         }
1305         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1306         {
1307             \prop_map_break:n

```

```

1308             { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1309         }
1310     }
1311 }
1312 \normalfont
1313 \ctex_ltj_update_mathfont:
1314 }
1315 }

```

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

```

1316 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1317 {
1318     \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1319     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1320     {
1321         \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1322         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1323     }
1324 }
1325 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1326 {
1327     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1328     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1329     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1330     \cs_if_free:cTF
1331     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1332     {
1333         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1334         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1335     }
1336     {
1337         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1338         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1339     }
1340 }

```

14.3.3.6 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1341 \keys_define:nn { ctex_ltj / fontspec }
1342 {
1343     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1344     AlternateFont .value_required:n = true ,
1345     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1346     CharRange .value_required:n = true
1347 }

```

\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。

```

1348 \group_begin:
1349 \char_set_catcode_other:N \l
1350 \cs_set:Npn \__ctex_ltj_tmp:w #1
1351 {
1352     \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1353     {
1354         \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1355         {
1356             \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1357             \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1358             \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1359             \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1360             { ! \tl_if_blank_p:n { #####1 } }

```

```

1361         \seq_concat:NNN \l__ctex_ltj_alternate_seq
1362         \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1363     }
1364 }
1365 }
1366 \char_set_catcode_active:N \l
1367 \__ctex_ltj_tmp:w { || }
1368 \group_end:
1369 \seq_new:N \l__ctex_ltj_tmp_seq
1370 \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。

```

1371 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1372 {
1373     \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1374     {
1375         \__ctex_ltj_check_family:n {#1}
1376         \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1377         { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1378         #3
1379     }
1380     { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1381 }

```

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

```

1382 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1383 {
1384     \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1385     { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1386 }
1387 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1388 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1389 { m m m +O{ } m u{ \q_stop } }
1390 {
1391     \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1392     \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1393     \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1394     \tl_if_blank:nTF {#5}
1395     { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1396     {
1397         \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1398         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1399     }
1400     \use:x
1401     {
1402         \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1403         { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1404         { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1405         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1406     }
1407 }
1408 \clist_new:N \l__ctex_ltj_alternate_options_clist

```

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

```

1409 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1410 {
1411     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1412     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1413     \ctex_ltj_set_alternate_family:coonn
1414     { \__ctex_ltj_alternate_cs:n {#1} }
1415     { \l__ctex_ltj_char_range_clist }
1416     { \l__ctex_ltj_font_options_clist } {#2} {#1}

```

```

1417 }
1418 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1419 {
1420   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1421   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1422   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1423 }
1424 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

\ctex_ltj_save_alternate_family:Nnnn

保存替代字体序列的定义,以备定义主字体时使用。

```

1425 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnnn #1#2#3#4
1426 {
1427   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1428   \cs_gset_protected_nopar:Npx #1
1429   { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1430 }

```

\ctex_ltj_set_alternate_family:nnn

实际定义替代字体族。

```

1431 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1432 {
1433   \group_begin:
1434   \l__ctex_ltj_change_encoding:
1435   \cs_set_eq:NN \CJKfamily \use_none:n
1436   \ctex_ltj_swap_cs:NN
1437   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1438   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1439   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1440   \group_end:
1441 }
1442 \tl_new:N \l__ctex_ltj_alternate_family_tl

```

\ctex_ltj_swap_cs:NN

交换两个控制序列的意义。

```

1443 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1444 {
1445   \cs_set_eq:NN \l__ctex_ltj_tmp:w #1
1446   \cs_set_eq:NN #1 #2
1447   \cs_set_eq:NN #2 \l__ctex_ltj_tmp:w
1448   \cs_undefine:N \l__ctex_ltj_tmp:w
1449 }

```

LTJFONTUID

\l__ctex_ltj_update_family_uid:N

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

```

1450 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1451 \cs_new_protected_nopar:Npn \l__ctex_ltj_update_family_uid:N #1
1452 {
1453   \int_gincr:N \g__ctex_ltj_family_int
1454   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1455 }
1456 \int_new:N \g__ctex_ltj_family_int

```

\ctex_ltj_declare_alternate_shape:nnnnnn

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

```

1457 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1458 {
1459   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1460   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1461   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1462   { \l_fontspec_family_tl } {#3} {#4}
1463 }

```

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

```

1464 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1465 {
1466   \clist_map_inline:Nn #1
1467   {
1468     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1469     {
1470       \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1471       \l__ctex_ltj_char_range_tl
1472     }
1473     { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1474   }
1475   \__ctex_ltj_save_alternate_shape:cnm
1476   { \__ctex_ltj_alternate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1477   { luatexja.jfont.clear_alt_font_latex }
1478   { '\lua_escape:e { \CJK@encoding/#2/#3/#4 }' }
1479 }

```

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。

```

1480 \NewDocumentCommand \ctex_ltj_set_alternate_shape:nnn
1481 { m m > { \SplitArgument { 1 } { -> } } m }
1482 { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
1483 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
1484 {
1485   \ctex_ltj_set_alternate_shape:n
1486   {
1487     \IfNoValueTF {#4}
1488     { \int_eval:n {#3} , \int_eval:n {#3} , }
1489     {
1490       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1491       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1492     }
1493     '\lua_escape:e { \CJK@encoding/#2 }' ,
1494     '\lua_escape:e { \CJK@encoding/#1 }'
1495   }
1496 }
1497 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1498 {
1499   \lua_now:e { luatexja.jfont.set_alt_font_latex ( #1 ) }
1500   \__ctex_ltj_save_alternate_shape:cnm
1501   { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1502   { luatexja.jfont.set_alt_font_latex } {#1}
1503 }

```

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

```

1504 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1505 {
1506   \tl_map_inline:Nn #3
1507   {
1508     \ctex_ltj_set_alternate_shape:n
1509     {
1510       ##1 ,
1511       '\lua_escape:e { \CJK@encoding/#2 }' ,
1512       '\lua_escape:e { \CJK@encoding/#1 }'
1513     }
1514   }
1515 }

```

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。

```

1516 \cs_new_protected_nopar:Npn \__ctex_ljt_save_alter_nate_shape:Nnn #1#2#3
1517 {
1518   \group_begin:
1519   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1520   \cs_set_eq:NN \l__ctex_ljt_base_family_tl \scan_stop:
1521   \cs_set_eq:NN \lua_escape:e \scan_stop:
1522   \cs_gset_protected_nopar:Npx #1
1523   { \exp_not:o {#1} \exp_not:N \lua_now:e { #2 ( #3 ) } }
1524   \group_end:
1525 }
1526 \cs_generate_variant:Nn \__ctex_ljt_save_alter_nate_shape:Nnn { c }

```

`clearalter_natefont`
`resetalter_natefont`

清除和重置操作总是全局的。

```

1527 \keys_define:nn { ctex }
1528 {
1529   clearalter_natefont .code:n =
1530   { \clist_map_function:xN {#1} \ctex_ljt_clear_alter_nate_font:n } ,
1531   resetalter_natefont .code:n =
1532   { \clist_map_function:xN {#1} \ctex_ljt_reset_alter_nate_font:n } ,
1533   clearalter_natefont .default:n = \l_ctex_ljt_family_tl ,
1534   resetalter_natefont .default:n = \l_ctex_ljt_family_tl
1535 }
1536 \cs_new_protected_nopar:Npn \ctex_ljt_clear_alter_nate_font:n #1
1537 {
1538   \group_begin:
1539   \ctex_ljt_family_if_exist:xNTF {#1} \l__ctex_ljt_base_family_tl
1540   {
1541     \cs_if_exist_use:cT { \__ctex_ljt_alter_nate_cs:n { clear / #1 } }
1542     {
1543       \prop_gput:Nno \g__ctex_ljt_reset_alter_nate_prop
1544       {#1} { \l__ctex_ljt_base_family_tl }
1545       \tl_set_eq:NN \CJK@family \l__ctex_ljt_base_family_tl
1546       \selectfont
1547     }
1548   }
1549   { \__ctex_ljt_family_unknown_warning:n {#1} }
1550   \group_end:
1551 }
1552 \cs_new_protected_nopar:Npn \ctex_ljt_reset_alter_nate_font:n #1
1553 {
1554   \group_begin:
1555   \prop_gpop:NnNT \g__ctex_ljt_reset_alter_nate_prop {#1} \CJK@family
1556   {
1557     \tl_set_eq:NN \l__ctex_ljt_base_family_tl \CJK@family
1558     \use:c { \__ctex_ljt_alter_nate_cs:n { reset / #1 } }
1559     \selectfont
1560   }
1561   \group_end:
1562 }
1563 \prop_new:N \g__ctex_ljt_reset_alter_nate_prop
1564 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

`declarecharrange`

预先声明字符范围。

```

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

```

```
1574 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }
```

#1 是名字, #2 是范围。

```
1575 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1576 {
1577   \tl_clear:N \l__ctex_ltj_char_range_tl
1578   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1579   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1580   \ctex_ltj_def_char_range_key:n {#1}
1581   \tl_clear:N \l__ctex_ltj_char_range_tl
1582 }
1583 \tl_new:N \l__ctex_ltj_char_range_tl
1584 \prop_new:N \g__ctex_ltj_char_range_prop
```

预先解释字符区间的意义。

```
1585 \NewDocumentCommand \ctex_ltj_save_char_range:n
1586 { > { \SplitArgument { 1 } { -> } } m }
1587 { \ctex_ltj_save_char_range:nn #1 }
1588 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1589 {
1590   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1591   { {
1592     \IfNoValueTF {#2}
1593     { \int_eval:n {#1} , \int_eval:n {#1} }
1594     {
1595       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1596       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1597     }
1598   } }
1599 }
```

在字体设置选项中定义字符范围键。

```
1600 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1601 {
1602   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1603   {
1604     \keys_define:nn { ctex_ltj / fontspec }
1605     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1606   }
1607 }
```

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。

```
1608 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1609 {
1610   \tl_if_blank:nTF {#2}
1611   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1612   {
1613     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1614     {
1615       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1616       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1617     }
1618   }
1619 }
```

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]。

```
1620 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1621 { +o o u { \q_stop } }
1622 {
```

```

1623 \exp_args:Nnf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1624 {
1625   \IfNoValueTF {#1} { {#3} }
1626   {
1627     \IfNoValueTF {#2}
1628     { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ {#1} ] {#3} } }
1629     { [ {#1} ] { [ #2 ] } }
1630   }
1631 }
1632 \seq_put_right:No \l__ctex_ltj_alterate_seq { \l__ctex_ltj_tmp_tl }
1633 }

```

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
}

1634 \AtBeginDocument
1635 {
1636   \ctex_appto_cmd:NnnTF \verbatim@font
1637   { \char_set_catcode_letter:n { 64 } }
1638   { \CTEX@verbatim@font@hook }
1639   { }
1640   { \ctex_patch_failure:N \verbatim@font }
1641 }
1642 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1643 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

```

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

```

1644 <@@=>
1645 \cs_set_eq:NN \@@italiccorr \/
1646 <@@=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 `\ltjsetpar@global`。

```

1647 \cs_new_protected_nopar:Npn \ctex_ltj_set_kanjiskip:N
1648 { \ltjsetpar@global \ltjsetkanjiskip }
1649 \cs_new_protected_nopar:Npn \ctex_ltj_set_xkanjiskip:N
1650 { \ltjsetpar@global \ltjsetxkanjiskip }

1651 </luatex>

```

14.3.4 ctex-engine-uptex.def

```

1652 <*uptex|aptex>

```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```

1653 \tl_set:Nn \CJKrmdefault { zhrm }

```

```

1654 \tl_set:Nn \CJKsfdefault { zhsf }
1655 \tl_set:Nn \CJKttdefault { zhTT }
1656 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1657 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }
1658 \RenewDocumentCommand \rmfamily { }
1659 {
1660   \not@math@alphabet \rmfamily \mathrm
1661   \romanfamily \rmdefault
1662   \kanjifamily \CJKrmdefault
1663   \selectfont
1664 }
1665 \RenewDocumentCommand \sffamily { }
1666 {
1667   \not@math@alphabet \sffamily \mathsf
1668   \romanfamily \sfdefault
1669   \kanjifamily \CJKsfdefault
1670   \selectfont
1671 }
1672 \RenewDocumentCommand \ttfamily { }
1673 {
1674   \not@math@alphabet \ttfamily \mathtt
1675   \romanfamily \ttdefault
1676   \kanjifamily \CJKttdefault
1677   \selectfont
1678 }
1679 \NewDocumentCommand \CJKfamily { m }
1680 {
1681   \kanjifamily {#1}
1682   \selectfont
1683 }

```

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

```

1684 \DeclareErrorKanjFont{JY2}{zhrm}{m}{n}{10}
1685 \DeclareKanjSubstitution{JY2}{zhrm}{m}{n}
1686 \DeclareKanjSubstitution{JT2}{zhrm}{m}{n}
1687 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
1688 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
1689 \jfam \symmincho

```

在导言区末尾更新 `\CJKfamilydefault`。

```

1690 \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。

```

1691 \normalfont

```

`\em` 取消 $\text{up}\text{\LaTeX}$ 对 `\em` 使用 `\mcfamily`、`\gtfamily` 命令的重定义, 恢复 $\text{\LaTeX}2_{\epsilon}$ 对 `\em` 的原始定义。如果用户已经重定义了 `\em`, 则新定义保持不变。 $\text{up}\text{\LaTeX}$ 2016/05/07u00 的定义有所变化, 这一行为可以由用户通过 `platexrelease` 包改变, 需要分支处理。

```

1692 \ctex_patch_cmd_once:NnnTF \em
1693 { \ExplSyntaxOff }
1694 { \eminnershapes \else \gtfamily \itshape }
1695 { \eminnershapes \else \itshape }
1696 { }
1697 {
1698   \ctex_patch_cmd:Nnn \em
1699   { \mcfamily \upshape \else \gtfamily \itshape }
1700   { \eminnershapes \else \itshape }
1701 }
1702 \cs_set_nopar:Npn \eminnershapes { \upshape }

```

```
\ctex_set_upfamily:nnn
```

将 NFSS 字体族 #1 设置为 JFM 字体名 #2, 粗体形式字体名 #3。其中字体名形如 `upzhserif`, 不包括表示方向的后缀 `-h` 与 `-v`。粗体字体名为空时不设置该字形。本命令不设置字体映射,

需要复用已有的字体映射或另行设置。

```

1703 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1704 {
1705   \DeclareKanjiFamily{JY2}{#1}{}
1706   \DeclareKanjiFamily{JT2}{#1}{}
1707   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1708   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1709   \tl_if_empty:nF { #3 }
1710   {
1711     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1712     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1713   }
1714 }
1715 \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。

```

1716 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1717 {
1718   \ctex_set_zhmap:n
1719   {
1720     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1721     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1722     \tl_if_empty:nF { #3 }
1723     {
1724       \special{ pdf:mapline~ #1b-h~  UniGB-UTF16-H~ #3 }
1725       \special{ pdf:mapline~ #1b-v~  UniGB-UTF16-V~ #3 }
1726     }
1727   }
1728 }
1729 \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。

```

1730 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1731 {
1732   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1733   \ctex_set_upmap:nnn { upserifit } { #3 } {}
1734   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1735   \ctex_set_upmap:nnn { upmono } { #6 } {}
1736 }
1737 \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。

```

1738 \RequirePackage { pxeveryysel }
1739 </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。

```

1740 <*pdf|luatex|uptex|aptex>
1741 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1742 {
1743   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1744   {
1745     \group_begin:
1746     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n

```

```

1747         \tl_gset:Nx \CJKfamilydefault
1748         {
1749             \str_case:onF { \familydefault }
1750             {
1751                 { \rmdefault } { \exp_not:N \CJKrmdefault }
1752                 { \sfdefault } { \exp_not:N \CJKsfdefault }
1753                 { \ttdefault } { \exp_not:N \CJKttdefault }
1754             }
1755             { \CJKfamilydefault }
1756         }
1757     \group_end:
1758 }

```

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

```

1759 <*luatex>
1760     \ctex_ltj_ensure_default_family:
1761 </luatex>
1762 }

```

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

```

1763 \tl_new:N \l__ctex_family_default_init_tl
1764 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1765 \tl_set:Nx \l__ctex_family_default_init_tl
1766 {
1767     \exp_not:N \__ctex_family_default_wrap:n
1768     { \exp_not:o { \CJKfamilydefault } }
1769 }
1770 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1771 <pdf|luatex|uptex|aptex>

```

14.3.6 操作系统的判断

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

```

1772 <*luatex>
1773 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1774 {
1775     \tl_gset:Nx \g__ctex_fontset_tl
1776     {
1777         \lua_now:e
1778         {
1779             if ~ os.name == 'windows' then ~
1780                 tex.sprint ( 'windows' )
1781             elseif ~ os.name == 'macosx' then ~
1782                 tex.sprint ( 'mac' )
1783             else ~
1784                 tex.sprint ( 'fandol' )
1785             end
1786         }
1787     }
1788 }
1789 </luatex>

```

Ap_{La}T_EX 可以使用 `\ngostype` 来判断。

```

1790 <*aptex>
1791 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1792 {
1793     \tl_gset:Nx \g__ctex_fontset_tl
1794     {
1795         \str_case:onF { \ngostype }
1796         {
1797             { Win32 } { windows }

```

```

1798         { Win64 } { windows }
1799         { Darwin } { mac }
1800     }
1801     { fandol }
1802 }
1803 }
1804 </aptex>

```

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

```

1805 <*xetex|pdfTeX|uptex>
1806 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1807 {
1808     \file_if_exist:nTF { /dev/null }
1809     {
1810         \file_if_exist:nTF { nul: }
1811         {
1812             \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1813             { \ctex_if_macosx:TF { mac } { windows } }
1814             { \ctex_if_macosx:TF { mac } { fandol } }
1815         }
1816         { \ctex_if_macosx:TF { mac } { fandol } }
1817     }
1818     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1819 }
1820 \str_const:Nx \c__ctex_upper_case_file_str
1821 { \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。

```

1822 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1823 {
1824     \file_if_exist:nTF { \c__ctex_macosx_file_str }
1825     { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1826     { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1827 }
1828 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1829 </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。

```

1830 <*class|ctex>
1831 \@ifpackageloaded { hyperref }
1832 {
1833     \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1834     { \hypersetup {#1} }
1835 }
1836 {
1837     \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1838     { \PassOptionsToPackage {#1} { hyperref } }
1839 }
1840 </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。

```

1841 <pdfTex>
1842 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1843 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1844 {
1845   \ctex_hypersetup:n { CJKbookmarks = true }
1846   \sys_if_output_pdf:TF
1847   { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1848   {
1849     \ctex_at_end_package:nn { hyperref }
1850     {
1851       \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1852       {
1853         \AtBeginShipoutFirst
1854         { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1855       }
1856       { \RequirePackage { xCJK2uni } }
1857     }
1858   }
1859 }
1860 { \ctex_hypersetup:n { pdfencoding = unicode } }
1861 </pdfTex>

```

在 $\text{Xe}\text{T}\text{E}\text{X}$ 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 `hxdetex.def` 中设置了 `\Hy@unicodetrue`, 从而书签总是会被 `\HyPsd@ConvertToUnicode` 转化成 UTF-16BE 编码的形式(抄录自 `\pdfstringdef` 的定义):

```

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

```

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

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

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

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>

\HyPsd@LoadUnicode 载入 puenc.def), 在 LuaTeX 下也启用这个选项。

```
1862 <*xetex|luatex>
1863 \ctex_hypersetup:n { pdfencoding = unicode }
1864 </xetex|luatex>
```

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

```
1865 <*uptex|aptex>
1866 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1867 \ctex_at_end_package:nn { hyperref }
1868 { \AtBeginShipoutFirst { \special { pdf:tounicode~ UTF8-UTF16 } } }
1869 </uptex|aptex>

1870 <*pdfTeX|xetex|luatex|uptex|aptex>
```

14.3.8 CJKfntef、xeCJKfntef 相关设置

CTEX 宏集对 pdfTeX 与 XeTeX 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。

```
1871 <*pdfTeX>
1872 \RequirePackage { CJKfntef }
1873 \normalem
1874 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1875 { \tl_clear:c { CJK#1color } }
1876 </pdfTeX>
1877 <*xetex>
1878 \RequirePackage { xeCJKfntef }
1879 \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
1880 {
1881   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1882   { \xeCJKsetup { #1 / format = { } } }
1883 }
1884 {
1885   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1886   { \tl_clear:c { CJK#1color } }
1887 }
1888 </xetex>
1889 <*luatex|uptex|aptex>
1890 \msg_new:nnn { ctex } { fntef-not-available }
<luatex> 1891 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
<uptex> 1892 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
<aptex> 1893 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1894 \msg_warning:nn { ctex } { fntef-not-available }
1895 </luatex|uptex|aptex>
1896 \clist_map_inline:nn
1897 { underdot , underline , underdblline , underwave , sout , xout }
1898 <*pdfTeX|xetex>
1899 {
1900   \__ctex_clear_fntef_color:n {#1}
1901   \cs_new_protected_nopar:cpx { CTEX#1 }
1902   {
1903     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1904     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1905     \exp_not:c { CJK#1 }
1906   }
1907 }
1908 \cs_new_protected_nopar:Npn { CTEXfilltwosides }
1909 {
1910   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1911   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
```

```

1912 \CJKfilltwosides
1913 }
1914 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1915 </pdfTeX|xetex>
1916 <*luatex|uptex|aptex>
1917 { \cs_new_eq:cN { CTEX#1 } \use:n }
1918 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1919 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1920 </luatex|uptex|aptex>
1921 <*pdfTeX>
1922 \clist_map_inline:nn
1923 {
1924     underdotbasesep , underdotsep , underlinebasesep ,
1925     underlinesep , underdbllinesep , underdbllinebasesep ,
1926     underwavebasesep , underwavesep , southeight ,
1927     underdotcolor , underwavecolor , underlinecolor ,
1928     underdbllinecolor , soutcolor , xoutcolor
1929 }
1930 {
1931     \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1932     \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1933 }
1934 </pdfTeX>

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1935 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1936 <*pdfTeX|xetex>
1937 {
1938     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1939     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1940 }
1941 </pdfTeX|xetex>
1942 <*luatex>
1943 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1944 </luatex>
1945 <*uptex|aptex>
1946 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
1947 </uptex|aptex>
1948 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1949 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1950 <*pdfTeX|xetex>
1951 {
1952     \cs_set_protected_nopar:Npn \CJKglue
1953     { \skip_horizontal:N \l__ctex_ccglue_skip }
1954 }
1955 </pdfTeX|xetex>
1956 <*luatex>
1957 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
1958 </luatex>
1959 <*uptex|aptex>
1960 { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
1961 </uptex|aptex>
1962 \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
1963 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1964 {
1965 <*pdfTeX|xetex>
1966     \if_meaning:w \CJKglue \__ctex_ccglue:
1967     \prg_return_false: \else: \prg_return_true: \fi:
1968 </pdfTeX|xetex>

```

```

1969 <*luatex>
1970   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1971   { \prg_return_false: } { \prg_return_true: }
1972 </luatex>
1973 <*uptex|aptex>
1974   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
1975   { \prg_return_false: } { \prg_return_true: }
1976 </uptex|aptex>
1977 }

```

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

```

1978 <*pdf|xetex>
1979 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1980 </pdf|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 `XYTeX` 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 `XYTeX` 可以用 `\fontcharwd` 来改进。而在 `pdfTeX` 下, 若使用 `zhmetrics` 技术, 所有的汉字共享同一个 `.tfm`, `\fontcharwd` 也就没有意义。在 `LuaTeX` 下, `LuaTeX-ja` 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 `\zw` 作为汉字宽度。`upTeX` 可以直接使用原生的长度单位 `zw`。

```

1981 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
<pdf|xetex> 1982 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1983 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1984 { \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-ja` 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。

```

1985 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1986 {
1987   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1988   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1989 }
1990 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1991 \if_cs_exist:N \@EverySelectfont@Init
1992   \group_begin:
1993   \cs_set:Npn \__ctex_tmp:N #1
1994   {
1995     \tl_set:Nn \l__ctex_tmp_tl {#1}
1996     \cs_new_eq:NN \CTEX@selectfont@save #1
1997     \cs_new_protected_nopar:Npn \__ctex_restore_selectfont:
1998     {
1999       \tl_put_left:Nn \@EverySelectfont@Init
2000       { \let #1 \CTEX@selectfont@save }
2001       \cs_undefine:N \__ctex_restore_selectfont:
2002     }
2003   }
2004   \ctex_parse_name:NN \__ctex_tmp:N \selectfont
2005   \exp_last_unbraced:NNo \group_end:
2006   \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
2007   { \ExplSyntaxOff }
2008   { \size@update }
2009   { \CTEX@selectfont@hook \size@update }

```

```

2010 { \__ctex_restore_selectfont: }
2011 { \ctex_patch_failure:N \selectfont }
2012 \fi:

```

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

```

2013 <pdfTeX>
2014 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
2015 </pdfTeX>
2016 <*xetex|luatex|uptex|aptex>
2017 \EverySelectfont { \CTEX@selectfont@hook }
2018 </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。

```

2019 <*luatex>
2020 \ctex_add_to_selectfont:n
2021 {
2022   \ctex_ltj_select_font:
2023   \ctex_ltj_select_alterate_font:
2024 }
2025 \tl_set:Nn \CJK@family { song } \selectfont
2026 \tl_clear:N \CJK@family
2027 </luatex>

```

```

\ctex_update_xkanjiskip:
\l__ctex_xkanjiskip_skip

```

upTeX 和 LuaTeX-ja 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。

```

2028 <*luatex|uptex|aptex>
2029 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
2030 {
2031   \skip_if_eq:nnT
2032   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
2033   { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2034   {
2035     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
2036     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
2037     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2038   }
2039 }
2040 \tl_new:N \l__ctex_xkanjiskip_tl
2041 \tl_set:Nn \l__ctex_xkanjiskip_tl
2042 { .25\zw plus 1pt minus 1pt }
2043 { .25zw plus 1pt minus 1pt }
2044 \skip_new:N \l__ctex_xkanjiskip_skip
2045 \skip_set:Nn \l__ctex_xkanjiskip_skip
2046 { \ltjgetparameter { xkanjiskip } }
2047 { \tex_xkanjiskip:D }

2048 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2049 </luatex|uptex|aptex>

```

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

```

\cdp
\c wd
2050 <*luatex>
2051 \dim_new:N \cht
2052 \dim_new:N \cdp
2053 \dim_new:N \c wd
2054 \newluafunction \g__ctex_kanjisize_func
2055 \group_begin:
2056 \char_set_catcode_space:n { 32 }
2057 \lua_now:e

```

```

2058 {
2059   local nulltable = { }
2060   local t = lua.get_functions_table()
2061   local fmt = luatexja.jfont.font_metric_table
2062   local getattribute = tex.getattribute
2063   local setdimen = tex.setdimen
2064   t[\int_use:N \g__ctex_kanjisize_func] = function ()
2065     local ft = fmt[getattribute('ltj@curjfont')] or nulltable
2066     local ft = ft and ft.char_type or nulltable
2067     local fk = ft and ft[0] or nulltable
2068     setdimen('cht', fk.height or 0)
2069     setdimen('cdp', fk.depth or 0)
2070     setdimen('cwd', fk.width or ft.zw or 0)
2071   end
2072 }
2073 \group_end:
2074 \cs_new_protected_nopar:Npn \ctex_update_kanjisize:
2075 { \tex_luafunction:D \g__ctex_kanjisize_func }
2076 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2077 </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。

```

2078 \keys_define:nn { ctex }
2079 {
2080   <*pdf|xetex>
2081     space .choice: ,
2082     space / true .code:n =
<pdfx> 2083       { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<xetex> 2084       { \xeCJKsetup { CJKspace = true } } ,
2085     space / auto .code:n =
<pdfx> 2086       { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
<xetex> 2087       { \xeCJKsetup { CJKspace = false } } ,
2088     space / false .code:n =
<pdfx> 2089       { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
<xetex> 2090       { \xeCJKsetup { CJKspace = false } } ,
2091     space .default:n = { true } ,
2092     space .initial:n = { auto }
2093   </pdf|xetex>
2094   <*luatex|uptex|aptex>
2095     space .code:n =
2096     { \msg_warning:nn { ctex } { invalid-option } }
2097   </luatex|uptex|aptex>
2098 }

```

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

```

2099 \keys_define:nn { ctex }
2100 {
2101   punct .code:n =
2102   {
2103     \tl_set:Nx \l__ctex_punct_tl { #1 }
<pdfx> 2104     \punctstyle { \l__ctex_punct_tl }
<xetex> 2105     \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
<luatex> 2106     \ctex_mono_jfm:o { \l__ctex_punct_tl }
<uptex|aptex> 2107     \msg_warning:nn { ctex } { invalid-option }
2108   } ,
2109   punct .default:n = { quanjiao } ,
2110 }

```

X_{La}TeX、Lua_{La}TeX 和 up_{La}TeX 总是使用 UTF8 编码。

```

2111 <*xetex|luatex|uptex|aptex>
2112 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }

```

```
2113 </xetex| luatex|uptex|aptex>
2114 </pdfxetex| xetex| luatex|uptex|aptex>
```

14.3.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```
<class|ctex> 2115 \ctex_file_input:n { \c__ctex_engine_file_str }
```

14.4 用户设置接口

```
\ctexset 2116 <*class|ctex|ctexheading>
2117 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
2118 </class|ctex|ctexheading>
```

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

```
2119 <*class|ctex>
2120 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
2121 {
2122   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
2123   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
2124   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
2125 }
2126 \NewDocumentCommand \CTEXoptions { +o }
2127 {
2128   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
2129   { \ctexset~ {~ #1~ }~ is~ set. }
2130   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
2131 }
```

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。

```
2132 \keys_define:nn { ctex }
2133 {
2134   autoindent .choice: ,
2135   autoindent .default:n = { true } ,
2136   autoindent / true .code:n =
2137   {
2138     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2139     \ctex_select_size:
2140   } ,
2141   autoindent / false .code:n =
2142   { \tl_clear:N \l__ctex_autoindent_tl } ,
2143   autoindent / unknown .code:n =
2144   {
2145     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2146     \ctex_select_size:
2147   }
2148 }
```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```
2149 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2150 \cs_new_protected_nopar:Npn \ctex_select_size:
2151 { \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。

```
2152 \cs_new_protected_nopar:Npn \ctex_update_size:
```

```

2153 {
2154   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2155   {
2156     \ctex_update_stretch:
2157     \ctex_update_parindent:
2158   }
2159   { \ctex_update_ziju: }
2160 }
2161 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2162 \tl_new:N \l__ctex_ziju_tl
2163 \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。

```

2164 \ctex_add_to_selectfont:n
2165 { \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`。

```

2166 \keys_define:nn { ctex }
2167 {
2168   linestretch .code:n =
2169   {
2170     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2171     \ctex_select_size:
2172   } ,
2173   linestretch .value_required:n = true
2174 }
2175 \tl_new:N \l__ctex_line_stretch_tl
2176 \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。

```

2177 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2178 {
2179   \ctex_update_em_unit:
2180   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2181   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2182   {
2183     \ctex_if_ccglue_touched:TF
2184     { \ctex_update_ccwd: }
2185     {
2186       \dim_set:Nn \l__ctex_tmp_dim
2187       { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
2188       \skip_set:Nn \l__ctex_ccglue_skip
2189       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2190       \ctex_update_ccglue:
2191     }
2192   }
2193   {
2194     \int_set:Nn \l__ctex_tmp_int
2195     { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2196     \tex_divide:D \l__ctex_tmp_int \ccwd
2197     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
2198     {

```

```

2199         \skip_set:Nn \l__ctex_ccglue_skip
2200         {
2201             \c_zero_dim plus \dim_eval:n
2202             {
2203                 ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2204                 \l__ctex_tmp_int
2205             }
2206         }
2207     }
2208     { \skip_zero:N \l__ctex_ccglue_skip }
2209     \ctex_update_ccglue:
2210 }
2211 }

```

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

```

2212 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2213 {
2214     \tl_if_empty:NF \l__ctex_autoindent_tl
2215     {
2216         \dim_compare:nNnF \parindent = \c_zero_dim
2217         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2218     }
2219 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2220 \NewDocumentCommand \ziju { m }
2221 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2222 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2223 {
2224     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2225     \ctex_select_size:
2226 }

```

`\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。

```

2227 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2228 {
2229     \ctex_update_em_unit:
2230     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2231     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2232     \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。

```

2233     {
2234         \dim_set:Nn \l__ctex_tmp_dim
2235         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2236         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2237         \tex_divide:D \l__ctex_tmp_int \ccwd
2238         \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。

```

2239         \dim_compare:nNnF \parindent = \c_zero_dim
2240         {
2241             \int_compare:nNnF \l__ctex_tmp_int < 3
2242             { \int_sub:Nn \l__ctex_tmp_int { 2 } }
2243         }
2244         \skip_set:Nn \l__ctex_ccglue_skip
2245         {
2246             \l__ctex_ziju_dim

```

```

2247         plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2248         minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2249         { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
2250     }
2251 }
2252 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2253 \ctex_update_ccglue:

```

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

```

2254 \ctex_update_parindent:
2255 }
2256 \dim_new:N \l__ctex_ziju_dim

```

$\backslash\text{CTEXindent}$
 $\backslash\text{CTEXnoindent}$

过时命令。

```

2257 \NewDocumentCommand \CTEXindent { }
2258 {
2259     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2260     { \parindent is~ set~ to~ 2\ccwd. }
2261     \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2262 }
2263 \NewDocumentCommand \CTEXnoindent { }
2264 {
2265     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2266     { \parindent is~ set~ to~ 0pt. }
2267     \dim_zero:N \parindent
2268 }

```

14.6 中文数字与日期

```

2269 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2270 \RequirePackage { zhnumber }

```

$\backslash\text{chinese}$

```

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

```

给 `enumitem` 宏包注册 $\backslash\text{chinese}$ 、 $\backslash\text{zhnum}$ 和 $\backslash\text{zhdig}$ 。

```

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

```

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

```

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

```

`today`

```

2288 \cs_set_eq:NN \CTEX@todayold \today
2289 \keys_define:nn { ctex }
2290 {
2291     today .choice: ,
2292     today / old .code:n =
2293     { \cs_set_eq:NN \today \CTEX@todayold } ,
2294     today / small .code:n =

```

```

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

```

14.7 其它中文标题定义

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

```

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

2315 \keys_define:nn { ctex }
2316 {
2317     contentsname .tl_set:N = \contentsname ,
2318     listfigurename .tl_set:N = \listfigurename ,
2319     listtablename .tl_set:N = \listtablename ,
2320     figurename .tl_set:N = \figurename ,
2321     tablename .tl_set:N = \tablename ,
2322     abstractname .tl_set:N = \abstractname ,
2323     indexname .tl_set:N = \indexname ,
2324     appendixname .tl_set:N = \appendixname ,
2325     proofname .tl_set:N = \proofname ,
2326     bibname .tl_set:N = \refname
2327     bibname .tl_set:N = \bibname
2328 }
2329 \algorithmname .tl_set:N = \algorithmname ,
2330 \bibname .tl_set:N = \bibname ,
2331 \refname .tl_set:N = \refname ,
2332 \continuation .tl_set:N = \insertcontinuationtext
2333 }
2334 }

2335 }
2336 \msg_new:nnn { ctex } { ctexbibname }
2337 {
2338     Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.\\
2339     The~key~`\bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2340 }
2341 \tl_if_exist:NTF \insertcontinuationtext
2342 {
2343     \keys_define:nn { ctex }
2344     {
2345         algorithmname .tl_set:N = \algorithmname ,
2346         bibname .tl_set:N = \bibname ,
2347         refname .tl_set:N = \refname ,
2348         continuation .tl_set:N = \insertcontinuationtext
2349     }
2350 }
2351 {
2352     \tl_if_exist:NTF \bibname
2353     { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2354     {

```

```

2355 \tl_if_exist:NTF \refname
2356 { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2357 {
2358 \msg_warning:nn { ctex } { ctexbibname }
2359 \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2360 }
2361 }
2362 }
2363 </ctex>
2364 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```
2365 <*class|heading>
```

14.8.1 定义标题格式选项

```

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

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

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

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

```

```

2406     }
2407     \group_end:
2408 }
2409 }

\__ctex_def_heading_keys:n 2410 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2411 {
2412     \tl_put_right:Nx \l__ctex_tmp_tl
2413     {
2414         #1 .meta:nn = { ctex / #1 } { #####1 } ,
2415         #1 / name .code:n =
2416         { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2417         #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2418         #1 / before_skip .tl_set:N = \exp_not:c { CTEX@#1@before_skip } ,
2419         #1 / after_skip .tl_set:N = \exp_not:c { CTEX@#1@after_skip } ,
2420         #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2421         #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2422         #1 / numbering .initial:n = true ,
2423         #1 / before_skip .initial:n = \c_zero_skip ,
2424         #1 / after_skip .initial:n = \c_zero_skip ,
2425         #1 / indent .initial:n = \c_zero_dim ,
2426         #1 / before_skip .value_required:n = true ,
2427         #1 / after_skip .value_required:n = true ,
2428         #1 / indent .value_required:n = true ,
2429 <article|book|report>
2430         #1 / after_indent .bool_set:N = \exp_not:c { CTEX@#1@after_indent } ,
2431         #1 / fix_skip .bool_set:N = \exp_not:c { CTEX@#1@fix_skip } ,
2432         #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2433         #1 / hang .initial:n = true ,
2434         #1 / run_in .bool_set:N = \exp_not:c { CTEX@#1@run_in } ,
2435         #1 / to_cline .code:n =
2436         {
2437             \cs_set:Npn \exp_not:c { CTEX@#1@to_cline}
2438             \exp_not:n { #####1#####2 } { #####1 }
2439         } ,
2440         \__ctex_plus_key_aux:nn {#1} { break } ,
2441 </article|book|report>
2442         \__ctex_plus_key_aux:nn {#1} { format } ,
2443         \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2444         \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2445         \__ctex_plus_key_aux:nn {#1} { titleformat } ,
2446         \__ctex_plus_key_aux:nn {#1} { aftername } ,
2447         \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2448     }
2449 }
2450 \cs_new_nopar:Npn \__ctex_plus_key_aux:nn #1#2
2451 {
2452     #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2453     #1 / #2 + .code:n =
2454     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2455     #1 / #2 ~ + .code:n =
2456     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2457 }

```

`\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 $\text{\LaTeX}$ 3 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。

```

2458 \NewDocumentCommand \ctex_assign_heading_name:nn
2459 { m > { \SplitArgument { 1 } { , } } +m }
2460 { \__ctex_assign_heading_name:nnn {#1} #2 }
2461 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2462 {
2463     \tl_set:cn { CTEX@pre#1 } {#2}

```

```

2464 \IfNoValueTF {#3}
2465 { \tl_clear:c { CTEX@post#1 } }
2466 { \tl_set:cn { CTEX@post#1 } {#3} }
2467 }

```

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

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2468 \group_begin:
2469 <*book|report>
2470 \tl_set:Nn \l__ctex_tmp_tl
2471 {
2472   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2473   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2474   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2475   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2476   chapter / lofskip .initial:n = \c_zero_skip ,
2477   chapter / lotskip .initial:n = \c_zero_skip ,
2478   chapter / lofskip .value_required:n = true ,
2479   chapter / lotskip .value_required:n = true ,
2480 }
2481 </book|report>
2482 <*article|beamer>
2483 \tl_clear:N \l__ctex_tmp_tl
2484 </article|beamer>

```

定义标题键值选项。

```

2485 \seq_map_inline:Nn \c__ctex_headings_seq
2486 {
2487   \__ctex_initial_heading:n {#1}
2488   \__ctex_def_heading_keys:n {#1}
2489 }
2490 \use:x
2491 {
2492   \group_end:
2493   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2494 }
2495 <*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。

```

2496 <*book|report>
2497 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2498 {
2499   \CTEX@fixheadingskip
2500   \dim_compare:nNf \tex_pagegoal:D < \c_max_dim
2501   { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2502 }
2503 </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`), $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 就不会加上 `\parskip`。这时候就需要用户把 `\parskip` 加到 `before` 或者 `after` 作为修正。

```

2504 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2505 {
2506   \par
2507   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2508   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2509 }
2510 \skip_new:N \l__ctex_heading_skip

```

```

2511 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2512 { \skip_set:Nn \l__ctex_heading_skip }
2513 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

\partmark 提供 \partmark。

```

2514 \ProvideDocumentCommand \partmark { m }
2515 { \markboth { } { } }

```

\CTEXifname 用于判断当前标题是否有编号。
\CTEX@ifnametrue
\CTEX@ifnamefalse

```

2516 \cs_new_eq:NN \CTEXifname \use_ii:nn
2517 \cs_new_protected_nopar:Npn \CTEX@ifnametrue
2518 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2519 \cs_new_protected_nopar:Npn \CTEX@ifnamefalse
2520 { \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。

```

2521 <*book|report>
2522 \cs_new_protected_nopar:Npn \CTEX@addloflotskip #1
2523 {
2524   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2525   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2526   {
2527     \addtocontents { lof }
2528     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2529   }
2530   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2531   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2532   {
2533     \addtocontents { lot }
2534     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2535   }
2536 }
2537 </book|report>

```

```

\CTEX@addtocline 2538 \cs_new_protected:Npn \CTEX@addtocline #1#2
2539 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

2540 <@@=>
\part 2541 <*article>
2542 \renewcommand\part{%
2543   \if@noskipsec \leavevmode \fi
2544   \par
2545   \CTEX@part@break
2546   % \addvspace{4ex}%
2547   \CTEX@setheadingskip \CTEX@part@beforeskip
2548   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2549   \addvspace \CTEX@headingskip
2550   \ifodd \CTEX@part@afterindent
2551     \@afterindenttrue
2552   \else
2553     \@afterindentfalse
2554   \fi
2555   \secdef\@part\@spart}
2556 </article>
2557 <*book|report>
2558 \renewcommand\part{%
2559 % \if@openright
2560 % \cleardoublepage

```

```

2561 % \else
2562 % \clearpage
2563 % \fi
2564 \CTEX@part@break
2565 % \thispagestyle{plain}%
2566 \thispagestyle{\CTEX@part@pagestyle}%
2567 \if@twocolumn
2568 \onecolumn
2569 \@tempwatrue
2570 \else
2571 \@tempwafalse
2572 \fi
2573 % \null\vfil
2574 \CTEX@setheadingskip \CTEX@part@beforeskip
2575 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2576 \vspace*{\CTEX@headingskip}%
2577 \secdef\@part\@spart}
2578 </book|report>

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

```

```

2627 %      \addcontentsline{toc}{part}{#1}%
2628 \fi
2629 \else
2630 \CTEX@ifnamefalse
2631 \CTEX@makeanchor{part*}%
2632 %      \addcontentsline{toc}{part}{#1}%
2633 \fi
2634 \CTEX@addtocline{part}{#1}%
2635 %      \markboth{}{}%
2636 \partmark{#1}%
2637 {\interlinepenalty \@M
2638 %      \normalfont \centering
2639 \normalfont \CTEX@part@format
2640 %      \ifnum \c@secnumdepth >-2\relax
2641 %      \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2642 %      \fi
2643 \CTEX@hangindent{part}%
2644 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2645 %      \Huge\bfseries #2\par}%
2646 \CTEX@part@titleformat{#2}%
2647 \CTEX@part@aftertitle}%
2648 \@endpart}
2649 </book|report>

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

\@endpart 2681 <*book|report>
2682 \def\@endpart{%
2683 %      \vfil
2684 \CTEX@setheadingskip \CTEX@part@afterskip
2685 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2686 \vskip \CTEX@headingskip
2687 \newpage
2688 \if@twoside
2689 \if@openright
2690 \null
2691 \thispagestyle{empty}%

```

```

2692             \newpage
2693             \fi
2694         \fi
2695         \if@tempwa
2696             \twocolumn
2697         \fi}
2698 </book|report>

```

14.8.2.2 chapter 的标题

```

2699 <*book|report>

\chapter 2700 \renewcommand\chapter{%
2701 %             \if@openright\cleardoublepage\else\clearpage\fi
2702 %             \thispagestyle{plain}%
2703             \CTEX@chapter@break
2704             \thispagestyle{\CTEX@chapter@pagestyle}%
2705             \global\@topnum\z@
2706 %             \@afterindentfalse
2707             \ifodd \CTEX@chapter@afterindent
2708                 \@afterindenttrue
2709             \else
2710                 \@afterindentfalse
2711             \fi
2712             \secdef\@chapter\@schapter}

\@chapter 2713 \def\@chapter[#1]#2{%
2714     \ifnum \c@secnumdepth >\m@ne
2715     <*book>
2716         \if@mainmatter
2717     </book>
2718         \ifodd \CTEX@chapter@numbering
2719             \CTEX@ifnametrue
2720             \refstepcounter{chapter}%
2721 %             \typeout{\@chapapp\space\thechapter.}%
2722             \typeout{\CTEXthechapter}%
2723 %             \addcontentsline{toc}{chapter}
2724 %             {\protect\numberline{\thechapter}#1}%
2725         \else
2726             \CTEX@ifnamefalse
2727             \CTEX@makeanchor{\Hy@chapapp*}%
2728 %             \addcontentsline{toc}{chapter}{#1}%
2729         \fi
2730     <*book>
2731         \else
2732             \CTEX@ifnamefalse
2733             \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2734 %             \addcontentsline{toc}{chapter}{#1}%
2735         \fi
2736     </book>
2737         \else
2738             \CTEX@ifnamefalse
2739             \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2740 %             \addcontentsline{toc}{chapter}{#1}%
2741         \fi
2742         \CTEX@addtocline{chapter}{#1}%
2743         \chaptermark{#1}%
2744 %         \addtocontents{lof}{\protect\addvspace{10\p@}}%
2745 %         \addtocontents{lot}{\protect\addvspace{10\p@}}%
2746         \CTEX@addloflotskip{chapter}%
2747         \if@twocolumn
2748             \@topnewpage[\@makechapterhead{#2}]%
2749         \else
2750             \@makechapterhead{#2}%
2751         \@afterheading
2752     \fi}

\@schapter 2753 \def\@schapter#1{%

```

```

2754 \CTEX@ifnamefalse
2755 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2756 \if@twocolumn
2757 \topnewpage[\@makeschapterhead{#1}]%
2758 \else
2759 \makeschapterhead{#1}%
2760 \@afterheading
2761 \fi}

\@makechapterhead 2762 \def\@makechapterhead#1{%
2763 % \vspace*{50\p@}%
2764 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2765 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2766 \vspace*{\CTEX@headingskip}%
2767 % {\normalfont \parindent \z@ \raggedright
2768 {\normalfont \CTEX@chapter@format
2769 \interlinepenalty\@M
2770 % \ifnum \c@secnumdepth >\m@ne
2771 % \if@mainmatter
2772 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2773 % \fi
2774 % \fi
2775 \CTEX@hangindent{chapter}%
2776 {\CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
2777 % \Huge \bfseries #1\par\nobreak
2778 \CTEX@chapter@titleformat{#1}%
2779 \CTEX@chapter@aftertitle
2780 \nobreak
2781 % \vskip 40\p@
2782 \CTEX@setheadingskip \CTEX@chapter@afterskip
2783 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2784 \vskip \CTEX@headingskip
2785 }}

\@makeschapterhead 2786 \def\@makeschapterhead#1{%
2787 % \vspace*{50\p@}%
2788 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2789 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2790 \vspace*{\CTEX@headingskip}%
2791 % {\normalfont \parindent \z@ \raggedright
2792 {\normalfont \CTEX@chapter@format
2793 \interlinepenalty\@M
2794 \CTEX@hangindent{chapter}{}%
2795 % \Huge \bfseries #1\par\nobreak
2796 \CTEX@chapter@titleformat{#1}%
2797 \CTEX@chapter@aftertitle
2798 \nobreak
2799 % \vskip 40\p@
2800 \CTEX@setheadingskip \CTEX@chapter@afterskip
2801 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2802 \vskip \CTEX@headingskip
2803 }}

2804 </book|report>

```

14.8.2.3 section 类的标题

```

\startsection 2805 \def\startsection#1#2#3#4#5#6{%
2806 \if@noskipsec \leavevmode \fi
2807 \par
2808 % \@tempskipa #4\relax
2809 % \@afterindenttrue
2810 % \ifdim \@tempskipa <\z@
2811 % \@tempskipa -\@tempskipa \@afterindentfalse
2812 % \fi
2813 \CTEX@update@sectionformat@n{#1}%
2814 \ifodd \CTEX@afterindent

```

```

2815     \@afterindenttrue
2816 \else
2817     \@afterindentfalse
2818 \fi
2819 \if@nobreak
2820     \everypar{}%
2821 \else
2822 %     \addpenalty\@secpenalty\addvspace\@tempskipa
2823     \csname CTEX@#1@break\endcsname
2824     \CTEX@setheadingskip{#4}%
2825     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2826     \addvspace \CTEX@headingskip
2827 \fi
2828 \@ifstar
2829     {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2830     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

\@secntformat 2831 \def\@secntformat#1{%
2832 % \csname the#1\endcsname\quad}%
2833     \csname CTEX@#1name\endcsname
2834     \csname CTEX@#1@aftername\endcsname}

\@sect 2835 \def\@sect#1#2#3#4#5#6[#7]#8{%
2836     \ifnum #2>\c@secnumdepth
2837         \CTEX@ifnamefalse
2838         \CTEX@makeanchor@sect{#1*}%
2839         \let\@svsec\@empty
2840     \else
2841         \ifodd \csname CTEX@#1@numbering\endcsname
2842             \CTEX@ifnametrue
2843             \refstepcounter{#1}%
2844             \protected@edef\@svsec{\@secntformat{#1}\relax}%
2845         \else
2846             \CTEX@ifnamefalse
2847             \CTEX@makeanchor{#1*}%
2848             \let\@svsec\@empty
2849         \fi
2850     \fi
2851 % \@tempskipa #5\relax
2852 % \ifdim \@tempskipa>\z@
2853 \unless \ifodd \CTEX@runin
2854     \begingroup
2855         #6{%
2856             \CTEX@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2857 % \interlinepenalty \@M #8\@par}%
2858             \interlinepenalty \@M
2859             \csname CTEX@#1@titleformat\endcsname{#8}%
2860             \csname CTEX@#1@aftertitle\endcsname}%
2861     \endgroup
2862     \csname #1mark\endcsname{#7}%
2863 % \addcontentsline{toc}{#1}{%
2864 %     \ifnum #2>\c@secnumdepth \else
2865 %         \protect\numberline{\csname the#1\endcsname}%
2866 %     \fi
2867 %     #7}%
2868     \CTEX@addtocline{#1}{#7}%
2869 \else
2870     \def\@svsechd{%
2871         #6{\hskip\glueexpr #3\relax
2872 % \@svsec #8}%
2873         \@svsec
2874         \csname CTEX@#1@titleformat\endcsname{#8}%
2875         \csname CTEX@#1@aftertitle\endcsname}%
2876     \csname #1mark\endcsname{#7}%
2877 % \addcontentsline{toc}{#1}{%
2878 %     \ifnum #2>\c@secnumdepth \else
2879 %         \protect\numberline{\csname the#1\endcsname}%

```

```

2880 %      \fi
2881 %      #7}%
2882      \CTEX@addtocline{#1}{#7}}%
2883      \fi
2884      \@xsect{#5}}

\@ssect 2885 \def\@ssect#1#2#3#4#5{%
2886      \CTEX@ifnamefalse
2887 % \@tempskipa #3\relax
2888 % \ifdim \@tempskipa>\z@
2889      \unless \ifodd \CTEX@runin
2890      \begingroup
2891      #4{%
2892          \CTEX@hangfrom{\hskip\glueexpr #1\relax}%
2893 %      \interlinepenalty \@M #5\@par}%
2894          \interlinepenalty \@M
2895          \CTEX@titleformat@n{#5}%
2896          \CTEX@aftertitle}%
2897      \endgroup
2898      \else
2899 %      \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2900      \def\@svsechd{#4{\hskip\glueexpr #1\relax
2901          \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2902      \fi
2903      \@xsect{#3}}

\@xsect 2904 \def\@xsect#1{%
2905 % \@tempskipa #1\relax
2906 % \ifdim \@tempskipa>\z@
2907      \unless \ifodd \CTEX@runin
2908      \par \nobreak
2909 %      \vskip \@tempskipa
2910      \CTEX@setheadingskip{#1}%
2911      \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2912      \vskip \CTEX@headingskip
2913      \@afterheading
2914      \else
2915      \@nobreakfalse
2916      \global\@noskipsectrue
2917      \everypar{%
2918          \if@noskipsec
2919          \global\@noskipsecfalse
2920          {\setbox\z@\lastbox}%
2921          \clubpenalty\@M
2922          \begingroup \@svsechd \endgroup
2923          \unskip
2924 %      \@tempskipa #1\relax
2925 %      \hskip -\@tempskipa
2926          \hskip\glueexpr #1\relax
2927      \else
2928          \clubpenalty \@clubpenalty
2929          \everypar{}%
2930      \fi}%
2931      \fi
2932      \ignorespaces}

2933 <@@=ctex>

```

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

```

2934 \cs_new_protected:Npn \CTEX@hangindent #1#2
2935 {
2936     \bool_if:cTF { \CTEX@#1@hang }
2937     { \@hangfrom }
2938     { \noindent \use:n }
2939     { \__ctex_indent_aux:n {#1} #2 }
2940 }
2941 \cs_new_protected:Npn \__ctex_indent_aux:n #1

```

```

2942 {
2943   \group_begin:
2944     \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@indent } }
2945     \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
2946       { \skip_horizontal:N \l__ctex_heading_skip }
2947   \group_end:
2948 }

```

\CTEX@hangfrom hang 选项控制是否采用悬挂缩进。

```

2949 \cs_new_protected_nopar:Npn \CTEX@hangfrom
2950 {
2951   \bool_if:NTF \CTEX@hang
2952     { \@hangfrom }
2953     { \noindent \use:n }
2954 }

```

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

```

2955 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2956 {
2957   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2958   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2959   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2960   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2961   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2962   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2963 }
2964 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2965 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2966 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2967 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2968 \cs_new_eq:NN \CTEX@hang \c_true_bool
2969 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

```

\CTEX@part@tocline 2970 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 2971 {
2972   \CTEXifname
2973     { \CTEXthepart \hspace { 1em } }
2974     { }
2975   #2
2976 }
2977 <*/book|report>
2978 \cs_new:Npn \CTEX@chapter@tocline #1#2
2979 {
2980   \CTEXifname
2981     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2982     { }
2983   #2
2984 }
2985 </book|report>

```

```

\CTEXnumberline 2986 \cs_new_nopar:Npn \CTEXnumberline #1
2987 {
2988   \CTEXifname
2989     { \protect \numberline { \use:c { CTEXthe #1 } } }
2990     { }
2991 }
2992 \int_zero:N \l__ctex_tmp_int
2993 \seq_map_inline:Nn \c__ctex_section_headings_seq
2994 {
2995   \int_incr:N \l__ctex_tmp_int
2996   \cs_gset_protected_nopar:cpx {#1}
2997   {
2998     \exp_not:N \@startsection {#1}
2999     { \int_use:N \l__ctex_tmp_int }

```

```

3000      { \exp_not:c { CTEX@#1@indent } }
3001      { \exp_not:c { CTEX@#1@beforeskip } }
3002      { \exp_not:c { CTEX@#1@afterskip } }
3003      { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
3004    }
3005    \cs_new:cpn { CTEX@#1@tocline } ##1##2
3006      { \CTEXnumberline { ##1 } ##2 }
3007  }

```

14.8.2.4 附录标题

```

appendix/name 3008 \keys_define:nn { ctex }
appendix/number 3009 {
appendix/numbering 3010   appendix .meta:nn = { ctex / appendix } {#1} ,
3011   appendix / name .code:n =
3012     { \ctex_assign_heading_name:nn { appendix } {#1} } ,
3013   appendix / number .tl_set:N = \CTEX@appendix@number ,
3014   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
3015   appendix / numbering .initial:n = true
3016 }
3017 \tl_new:N \CTEX@preappendix
3018 \tl_new:N \CTEX@postappendix

\appendix 3019 \cs_new_eq:NN \CTEX@save@appendix \appendix
3020 \cs_gset_protected_nopar:Npn \appendix
3021 {
3022   \CTEX@save@appendix
3023   <*article>
3024   \gdef \CTEX@presection { \CTEX@preappendix }
3025   \gdef \CTEX@thesection { \CTEX@appendix@number }
3026   \gdef \CTEX@postsection { \CTEX@postappendix }
3027   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
3028   </article>
3029   <*book|report>
3030   \gdef \CTEX@prechapter { \CTEX@preappendix }
3031   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3032   \gdef \CTEX@postchapter { \CTEX@postappendix }
3033   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3034   </book|report>
3035 }

```

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。

```

3036 \cs_new_protected_nopar:Npn \CTEX@makeanchor #1
3037 { }

```

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

```

3038 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
<article> 3039 { part , spart , sect , ssect }
<book|report> 3040 { part , spart , chapter , schapter , sect , ssect }
3041 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3042 {
3043   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3044   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3045 }

```

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

```

3046 \cs_new_protected_nopar:Npn \CTEX@hyperheadinghook

```

```

3047 {
3048   \group_begin:
3049   \ifHy@implicit
3050     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3051     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3052     {
3053       \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3054       {
3055         \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3056         \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3057       }
3058     }
3059   \else:
3060     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3061     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3062   \fi:
3063   \group_end:
3064 }

3065 \ctex_at_end_package:nn { hyperref }
3066 {
3067   \cs_gset_protected_nopar:Npn \CTEX@makeanchor #1
3068   {
3069     \Hy@MakeCurrentHrefAuto {#1}
3070     \Hy@raisedlink
3071     {
3072       \hyper@anchorstart { \@currentHref }
3073       \hyper@anchorend
3074     }
3075   }
3076   \CTEX@hyperheadinghook
3077 }

```

14.8.2.6 兼容 titlesec 宏包

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

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

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

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

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

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

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

```

3078 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
3079 {
3080   \ifpackagewith { titlesec } { explicit }
3081   {
3082     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3083     \__ctex_titlesec_format_explicit:Nn

```

```

3084     }
3085     { }
3086     \clist_map_inline:nn
3087     { indentafter , noindentafter , indentfirst , nonindentfirst }
3088     {
3089         \@ifpackagewith { titlesec } { ##1 }
3090         {
3091             \clist_map_break:n
3092             { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3093         }
3094     { }
3095     }
3096     \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3097 }
3098 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
3099 {
3100     \__ctex_titlesec_format:n {#1}
3101     \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3102 }
3103 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
3104 {
3105     \cs_if_free:cF { ttlf@#1 }
3106     { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3107 }
3108 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
3109 {
3110     \tl_set:Nx #1
3111     {
3112         \bool_if:cTF { CTEX@#2@runin }
3113         { \exp_not:N \ttlh@runin }
3114         { \exp_not:N \ttlh@hang }
3115         \tl_tail:N #1
3116     }
3117 }
3118 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3119 {
3120     \cs_set_nopar:Npx #1 ##1
3121     {
3122         \bool_if:cTF { CTEX@#2@runin }
3123         { \exp_not:N \ttlh@runin }
3124         { \exp_not:N \ttlh@hang }
3125         \exp_args:No \tl_tail:n { #1 { } }
3126     }
3127 }
3128 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
3129 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3130 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3131 {
3132     \exp_not:n { {#1} {#2} {#3} {#4} }
3133     { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3134 }
3135 \@ifpackageloaded { titlesec }
3136 { }
3137 {
3138     \ctex_at_end_package:nn { titlesec }
3139     {
3140         \@ifpackagewith { titlesec } { loadonly }
3141         { }
3142         { \ctex_titlesec_hook: }
3143     }
3144 }

```

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

```

3145 \ctex_at_end_package:nn { titlesec }
3146 {
3147     <*book|report>

```

```

3148 \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3149 </book|report>
3150 \cs_if_free:NF \ttl@tocpart
3151 {
3152   \cs_set_protected_nopar:Npn \ttl@tocpart
3153   { \tl_set:Nn \ttl@a { \CTEXthe part \hspace { 1em } } }
3154 }
3155 \seq_map_inline:Nn \c__ctex_headings_seq
3156 {
3157   \cs_if_exist:cF { ttl@toc #1 }
3158   {
3159     \cs_new_protected_nopar:cpx { ttl@toc #1 }
3160     {
3161       \tl_set:Nn \exp_not:N \ttl@a
3162       {
3163         \exp_not:N \protect
3164         \exp_not:N \numberline { \exp_not:c { CTExthe #1 } }
3165       }
3166     }
3167   }
3168 }
3169 }

```

14.8.2.7 兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题²⁰。

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

```

3170 \group_begin:
3171 \char_set_catcode_other:N \#
3172 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
3173 {
3174   \ctex_patch_cmd:Nnn \ttl@settopmark
3175   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3176   {
3177     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3178     \CTEX@titlepslabel@set {#1}
3179   }
3180   \ctex_patch_cmd:Nnn \ttl@setsubmark
3181   { \protect \@namedef { the#1 } { } }
3182   {
3183     \protect \@namedef { the#1 } { }
3184     \CTEX@titlepslabel@clear {#1}
3185   }
3186   \ctex_patch_cmd:Nnn \ttl@setsubmark
3187   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3188   {
3189     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3190     \CTEX@titlepslabel@set {#2}
3191   }
3192 }
3193 \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。

```

3194 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
3195 {
3196   \cs_if_free:cF { CTExthe#1 }

```

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

```

3197     { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3198   }
3199   \cs_new_nopar:Npn \CTEX@titleslabel@clear #1
3200   {
3201     \cs_if_free:cF { CTEXthe#1 }
3202     { \protect \@namedef { CTEXthe#1 } { } }
3203   }

```

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

```

3204 \ctex_at_end_package:nn { titlesec }
3205 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3206 \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。

```

3207 \ctex_at_end_package:nn { titleps }
3208 {
3209   \cs_set_protected_nopar:Npn \ttl@setifthe #1
3210   {
3211     \exp_args:Nco \cs_set_nopar:Npn { ifthe #1 }
3212     {
3213       \CTEXifname
3214       { \protect \@firstoftwo }
3215       { \protect \@secondoftwo }
3216     }
3217   }
3218   \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3219 }

```

14.8.3 目录标签的宽度

```

\numberline 3220 \cs_new_protected:Npn \CTEX@toc@width@n #1
3221 {
3222   \hbox_set:Nn \l__ctex_tmp_box {#1}
3223   \dim_set:Nn \@tempdima
3224   {
3225     \dim_max:nn { \@tempdima }
3226     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
3227   }
3228 }
3229 \group_begin:
3230 \char_set_catcode_other:N \#
3231 \use:n
3232 {
3233   \group_end:
3234   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3235   { \CTEX@toc@width@n {#1} }
3236   { }
3237   { \ctex_patch_failure:N \numberline }
3238   \@ifpackageloaded { tocloft }
3239   { }
3240   {
3241     \ctex_at_end_package:nn { tocloft }
3242     {
3243       \ctex_preto_cmd:NnnTF \numberline
3244       { \char_set_catcode_letter:n { 64 } }
3245       { \CTEX@toc@width@n {#1} }
3246       { }

```

```

3247         { \ctex_patch_failure:N \numberline }
3248     }
3249 }
3250 }

```

14.8.4 页眉信息的修改

```

\ps@headings 3251 <*article>
3252 \if@twoside
3253   \ctex_patch_cmd:Nnn \ps@headings
3254     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3255     { \CTEXifname { \CTEXthesection \quad } { } }
3256   \ctex_patch_cmd:Nnn \ps@headings
3257     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3258     { \CTEXifname { \CTEXthesubsection \quad } { } }
3259 \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。

```

3260   \ctex_patch_cmd:Nnn \ps@headings
3261     { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3262     { \CTEXifname { \CTEXthesection \quad } { } }
3263 \fi:
3264 </article>
3265 <*book|report>
3266 \ctex_patch_cmd:Nnn \ps@headings
3267   {
3268     <book> \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3269     <report> \ifnum \c@secnumdepth > \m@ne
3270       \@chapapp \thechapter . ~ \ %
3271     <report> \fi
3272     <book> \fi \fi
3273   }
3274   { \CTEXifname { \CTEXthechapter \quad } { } }
3275 \if@twoside
3276   \ctex_patch_cmd:Nnn \ps@headings
3277     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3278     { \CTEXifname { \CTEXthesection \quad } { } }
3279 \fi:
3280 </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。

```

3281 \ctex_at_end_package:nn { fancyhdr }
3282 {
3283 <*article>
3284   \ctex_patch_cmd:Nnn \ps@fancy
3285     { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3286     { \CTEXifname { \CTEXthesection \quad } { } }
3287   \ctex_patch_cmd:Nnn \ps@fancy
3288     { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3289     { \CTEXifname { \CTEXthesubsection \quad } { } }
3290 </article>
3291 <*book|report>
3292   \ctex_patch_cmd:Nnn \ps@fancy
3293     { \ifnum \c@secnumdepth > \m@ne \@chapapp \thechapter . ~ \ \fi }
3294     { \CTEXifname { \CTEXthechapter \quad } { } }
3295   \ctex_patch_cmd:Nnn \ps@fancy
3296     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3297     { \CTEXifname { \CTEXthesection \quad } { } }
3298 </book|report>
3299 }
3300 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

3301 `<*beamer>`

3302 `\ExplSyntaxOff`

对应 `\partpage`。

```
3303 \defbeamerteremplate*{part page}{CTEX}[1][]{%
3304   \begingroup
3305     \ctexset{autoindent=false}%
3306     %
3307     {\usebeamerfont{part name}%
3308      \usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3309     %
3310     \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3311     \CTEX@part@format
3312     \parindent \dimexpr \CTEX@part@indent \relax
3313     \ifodd \CTEX@part@numbering
3314       \CTEX@partname \CTEX@part@aftername
3315     \fi
3316     \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3317     %
3318       \usebeamerfont{part title}\insertpart\par
3319       \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3320     \end{beamercolorbox}%
3321     \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3322   \endgroup
}
```

对应 `\sectionpage`。

```
3323 \defbeamerteremplate*{section page}{CTEX}[1][]{%
3324   \begingroup
3325     \ctexset{autoindent=false}%
3326     %
3327     {\usebeamerfont{section name}%
3328      \usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}
3329     %
3330     \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3331     \CTEX@section@format
3332     \parindent \dimexpr \CTEX@section@indent \relax
3333     \ifodd \CTEX@section@numbering
3334       \CTEX@sectionname \CTEX@section@aftername
3335     \fi
3336     \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3337     %
3338       \usebeamerfont{section title}\insertsection\par
3339       \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3340     \end{beamercolorbox}%
3341     \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3342   \endgroup
}
```

对应 `\subsectionpage`。

```
3343 \defbeamerteremplate*{subsection page}{CTEX}[1][]{%
3344   \begingroup
3345     \ctexset{autoindent=false}%
3346     %
3347     {\usebeamerfont{subsection name}%
3348      \usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
3349     %
3350     \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3351     \CTEX@subsection@format
3352     \parindent \dimexpr \CTEX@subsection@indent \relax
3353     \ifodd \CTEX@subsection@numbering
3354       \CTEX@subsectionname \CTEX@subsection@aftername
3355     \fi
3356     \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3357     %
3358       \usebeamerfont{subsection title}\insertsubsection\par
3359       \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3360     \end{beamercolorbox}%
3361     \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3362   \endgroup
}
```

```

3359     \end{beamercolorbox}%
3360     \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3361     \endgroup
3362 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3363 \defbeamerplatformalias{part page}{default}{CTEX}
3364 \defbeamerplatformalias{section page}{default}{CTEX}
3365 \defbeamerplatformalias{subsection page}{default}{CTEX}
3366 \ExplSyntaxOn
3367 </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。

```

3368 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
3369 {
3370     \protected@edef \@currentlabel
3371     {
3372         \cs_if_exist:cTF { CTEX@the#1 }
3373         { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3374         { \exp_not:o { \@currentlabel } }
3375     }
3376 }

```

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

```

3377 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3378 {
3379     \seq_map_inline:Nn \c__ctex_headings_seq
3380     { \ctex_fix_varioref_label:n { ##1 } }
3381 }

```

`\__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。

```

3382 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3383 {
3384     \tl_if_empty:cT { p@#1 }
3385     { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3386 }

```

如果 `varioref` 已经被载入,则使用它来设置。

```

3387 \ifpackageloaded { varioref }
3388 { \ctex_varioref_hook: }
3389 {
3390     \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3391     \RenewDocumentCommand \refstepcounter { m }
3392     {
3393         \CTEX@save@refstepcounter {#1}
3394         \CTEX@setcurrentlabel@n {#1}
3395     }
3396     \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3397 }

```

14.8.7 载入 $\langle scheme \rangle$ 文件

```
3398 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3399 </class|heading>
```

14.8.8 标题格式的 scheme 定义

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

```
3400 <*scheme&(article|book|report|beamer)>

3401 \keys_set:nn { ctex / part }
3402 {
3403     aftertitle = \par ,
3404     <*article|book|report>
3405     hang       = false ,
3406     </article|book|report>
3407     <*plain>
3408     name       = \partname \space ,
3409     <*article|book|report>
3410     number     = \thepart ,
3411     </article|book|report>
3412     <*beamer>
3413     number     = \insertromanpartnumber ,
3414     </beamer>
3415     </plain>
3416     <*chinese>
3417     number     = \chinese { part } ,
3418     </chinese>
3419     <*article>
3420     beforeskip = 4ex ,
3421     afterskip  = 3ex ,
3422     <*plain>
3423     format     = \raggedright ,
3424     nameformat = \Large \bfseries ,
3425     aftername  = \par \nobreak ,
3426     titleformat = \huge \bfseries ,
3427     afterindent = false
3428     </plain>
3429     <*chinese>
3430     format     = \Large \bfseries \centering ,
3431     aftername  = \quad ,
3432     afterindent = true
3433     </chinese>
3434     </article>
3435     <*book|report>
3436     aftername  = \par \vskip 20 \p@ ,
3437     beforeskip = 0pt \@plus 1fil ,
3438     afterskip  = 0pt \@plus 1fil ,
3439     pagestyle  = plain ,
3440     break      = \if@openright \cleardoublepage \else \clearpage \fi ,
3441     <*plain>
3442     format     = \centering ,
3443     nameformat = \huge \bfseries ,
3444     titleformat = \Huge \bfseries
3445     </plain>
3446     <*chinese>
3447     format     = \huge \bfseries \centering
3448     </chinese>
3449     </book|report>
3450     <*beamer>
3451     format     = \centering ,
3452     nameformat = \usebeamerfont { part ~ name }
3453                 \usebeamercolor [fg] { part ~ name } ,
3454     aftername  = \vskip 1em \par ,
3455     titleformat = \usebeamerfont { part ~ title }
```

```

3456 </beamer>
3457 }

3458 <*book|report>
3459 \keys_set:nn { ctex / chapter }
3460 {
3461     pagestyle = plain ,
3462     aftertitle = \par ,
3463     hang = false ,
3464     beforekip = 50 \p@ ,
3465     afterkip = 40 \p@ ,
3466     lofskip = 10 \p@ ,
3467     lotskip = 10 \p@ ,
3468     break = \if@openright \cleardoublepage \else \clearpage \fi ,
3469 <*plain>
3470     name = \chaptername \space ,
3471     number = \thechapter ,
3472     format = \raggedright ,
3473     nameformat = \huge \bfseries ,
3474     aftername = \par \nobreak \vskip 20 \p@ ,
3475     titleformat = \Huge \bfseries ,
3476     afterindent = false ,
3477     tocline = \CTEXnumberline {#1} #2
3478 </plain>
3479 <*chinese>
3480     number = \chinese { chapter } ,
3481     format = \huge \bfseries \centering ,
3482     aftername = \quad ,
3483     afterindent = true
3484 </chinese>
3485 }
3486 </book|report>

3487 <@@= >

3488 \keys_set:nn { ctex / section }
3489 {
3490 <*article|book|report>
3491     number = \thesection ,
3492     aftername = \quad ,
3493     aftertitle = \@@par ,
3494     beforekip = 3.5ex \@plus 1ex \@minus .2ex ,
3495     afterkip = 2.3ex \@plus .2ex ,
3496     runin = false ,
3497     break = \addpenalty \@secpenalty ,
3498 <*plain>
3499     format = \Large \bfseries ,
3500     afterindent = false
3501 </plain>
3502 <*chinese>
3503     format = \Large \bfseries \centering ,
3504     afterindent = true
3505 </chinese>
3506 </article|book|report>
3507 <*beamer>
3508 <*plain>
3509     name = \sectionname \space ,
3510 </plain>
3511     format = \centering ,
3512     number = \insertsectionnumber ,
3513     nameformat = \usebeamerfont { section ~ name }
3514                 \usebeamerfont { section ~ name } ,
3515     aftername = \vskip 1em \par ,
3516     titleformat = \usebeamerfont { section ~ title } ,
3517     aftertitle = \par
3518 </beamer>
3519 }

3520 \keys_set:nn { ctex / subsection }

```

```

3521 {
3522 <*article|book|report>
3523     number      = \thesubsection ,
3524     format      = \large \bfseries ,
3525     aftername    = \quad ,
3526     aftertitle   = \@@par ,
3527     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3528     afterskip    = 1.5ex \@plus .2ex ,
3529     runin        = false ,
3530     break        = \addpenalty \@secpenalty ,
3531 <*plain>
3532     afterindent  = false
3533 </plain>
3534 <*chinese>
3535     afterindent  = true
3536 </chinese>
3537 </article|book|report>
3538 <*beamer>
3539 <*plain>
3540     name         = \subsectionname \space ,
3541     number       = \insertsubsectionnumber ,
3542 </plain>
3543 <*chinese>
3544     number       = \arabic { section } . \arabic { subsection } ,
3545 </chinese>
3546     format       = \centering ,
3547     nameformat    = \usebeamerfont { subsection ~ name }
3548                   \usebeamercolor [fg] { subsection ~ name } ,
3549     aftername     = \vskip 1em \par ,
3550     titleformat   = \usebeamerfont { subsection ~ title } ,
3551     aftertitle    = \par
3552 </beamer>
3553 }

3554 <*article|book|report>

3555 \keys_set:nn { ctex / subsubsection }
3556 {
3557     number      = \thesubsubsection ,
3558     format      = \normalsize \bfseries ,
3559     aftername    = \quad ,
3560     aftertitle   = \@@par ,
3561     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3562     afterskip    = 1.5ex \@plus .2ex ,
3563     runin        = false ,
3564     break        = \addpenalty \@secpenalty ,
3565 <*plain>
3566     afterindent  = false
3567 </plain>
3568 <*chinese>
3569     afterindent  = true
3570 </chinese>
3571 }

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

```

```

3586 \keys_set:nn { ctex / subparagraph }
3587 {
3588     number      = \thesubparagraph ,
3589     format      = \normalsize \bfseries ,
3590     aftername    = \quad ,
3591     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
3592     break       = \addpenalty \@secpenalty ,
3593 <*plain>
3594     afterindent = false
3595 </plain>
3596 <*chinese>
3597     afterindent = true
3598 </chinese>
3599 }

```

处理 sub3section 与 sub4section 的格式。

```

3600 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3601 {
3602     \keys_set:nn { ctex / paragraph }
3603     {
3604         aftertitle = \@@par ,
3605         afterskip  = 1ex \@plus .2ex ,
3606         runin      = false
3607     }
3608 }
3609 {
3610     \keys_set:nn { ctex / paragraph }
3611     {
3612         afterskip  = 1em ,
3613         runin      = true
3614     }
3615 }
3616 \int_compare:nNnTF \g__ctex_section_depth_int > 3
3617 {
3618     \keys_set:nn { ctex / subparagraph }
3619     {
3620         aftertitle = \@@par ,
3621         afterskip  = 1ex \@plus .2ex ,
3622         runin      = false
3623     }
3624 }
3625 {
3626     \keys_set:nn { ctex / subparagraph }
3627     {
3628         afterskip  = 1em ,
3629         runin      = true
3630     }
3631 }
3632 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3633 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3634 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3635 <@@=ctex>

```

处理附录的格式。

```

3636 \keys_set:nn { ctex / appendix }
3637 <*article>
3638 { number      = \@Alph \c@section }
3639 </article>
3640 <*book|report>
3641 {
3642     name       = \appendixname \space ,
3643     number     = \@Alph \c@chapter
3644 }
3645 </book|report>
3646 </article|book|report>

```

```
3647 </scheme&(article|book|report|beamer)>
```

14.8.9 ctex.sty 的 heading 选项

```
3648 <*ctex|ctexheading>
```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```
3649 \clist_map_inline:nn { article , book , report , beamer }
3650 {
3651   \@ifclassloaded {#1}
3652     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3653     { }
3654 }
```

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。

```
3655 \msg_new:nnn { ctex } { not-standard-class }
3656 {
3657   None~of~the~standard~document~classes~was~loaded.\\
3658   Heading~`#1'~is~selected.\\
3659   ctex~may~not~work~as~expected.
3660 }
<ctex> 3661 \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 3662 \use:n
3663 {
3664   \tl_if_exist:NTF \c__ctex_std_class_tl
3665     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3666     {
3667       \cs_if_exist:NTF \chapter
3668         {
3669           \cs_if_exist:NF \if@mainmatter
3670             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3671             \tl_const:Nn \c__ctex_class_tl { book }
3672         }
3673         { \tl_const:Nn \c__ctex_class_tl { article } }
3674       \msg_warning:nx { ctex } { not-standard-class } { \c__ctex_class_tl }
3675     }
3676   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3677 }
<ctex> 3678 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3679 </ctex|ctexheading>
```

14.8.10 标题配置文件

```
3680 <*name>
```

```
3681 \keys_set_known:nn { ctex }
3682 {
3683   contentsname = 目录 ,
3684   listfigurename = 插图 ,
3685   listtablename = 表格 ,
3686   figurename = 图 ,
3687   tablename = 表 ,
3688   abstractname = 摘要 ,
3689   indexname = 索引 ,
3690   bibname = 参考文献 ,
3691   appendixname = 附录 ,
3692   proofname = 证明 ,
3693   algorithmname = 算法 ,
3694   refname = 参考文献 ,
3695   continuation = (续) ,
3696   part / name = { 第 , 部分 } ,
3697   chapter / name = { 第 , 章 }
```

```
3698 }
3699 </name>
```

14.9 chinese 方案的其他设置

```
3700 <*scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
<article|book|report> 3701 \pagestyle { headings }
```

日期格式。

```
3702 \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。

```
3703 \ctex_if_autoindent_touched:F
3704 { \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。

```
3705 <*!generic>
3706 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3707 <*beamer>
3708 {
3709   \uselanguage { ChineseGBK }
3710   \languagealias { chinese } { ChineseGBK }
3711   \ctex_file_input:n { ctex-name-gbk.cfg }
3712 }
3713 {
3714   \uselanguage { ChineseUTF8 }
3715   \languagealias { chinese } { ChineseUTF8 }
3716   \ctex_file_input:n { ctex-name-utf8.cfg }
3717 }
```

让 translator 包优先查找中文翻译。

```
3718 \clist_put_left:Nn \trans@languagepath { chinese }
3719 </beamer>
3720 <*!beamer>
3721 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3722 { \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。

```
3723 \int_compare:nNf \g__ctex_font_size_int > { -1 }
3724 { \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。

```
3725 \fp_compare:nNf { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3726 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3727 </!beamer>
3728 </!generic>
```

不使用标题定义时的通用设置。

```
3729 <*generic>
3730 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3731 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3732 {
3733   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3734   {
3735     \uselanguage { ChineseGBK }
3736     \languagealias { chinese } { ChineseGBK }
3737     \ctex_file_input:n { ctex-name-gbk.cfg }
3738   }
3739 }
```

```

3740     \uselanguage { ChineseUTF8 }
3741     \languagealias { chinese } { ChineseUTF8 }
3742     \ctex_file_input:n { ctex-name-utf8.cfg }
3743   }
3744   \clist_put_left:Nn \trans@languagepath { chinese }
3745 }
3746 {
3747   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3748   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3749   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3750   \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3751   { \int_gset:Nn \g__ctex_font_size_int { 0 } }
3752   \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3753   { \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。

```

3754   \tl_if_exist:NT \c__ctex_std_class_tl
3755   { \RequirePackage { indentfirst } }
3756 }
3757 </generic>
3758 </scheme&chinese>

```

14.10 中文字号

```

3759 <*class|ctex|ctexsize>

```

```

\zihao 3760 \NewDocumentCommand \zihao { m }
3761 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

\ctex_zihao:n 3762 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3763 {
3764   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3765   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3766   { \msg_error:nnn { ctex } { fontsize } {#1} }
3767 }
3768 \msg_new:nnnn { ctex } { fontsize }
3769 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3770 {
3771   The~old~font~size~is~used~if~you~continue.\\
3772   The~available~font~sizes~are~listed~as~follow.\\
3773   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3774 }

```

14.10.1 定义中文字号

```

\c__ctex_font_size_prop
\__ctex_save_font_size:nn

```

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

```

3775 \prop_new:N \c__ctex_font_size_prop
3776 \seq_new:N \c__ctex_font_size_seq
3777 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3778 {
3779   \use:x
3780   {
3781     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3782     {
3783       { \dim_to_decimal:n {#2} }
3784       { \dim_to_decimal:n { (#2) * 6 / 5 } }
3785     }
3786   }
3787   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3788 }
3789 \clist_map_inline:nn

```

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

```

3790 {
3791   { 8 } { 5 bp } ,
3792   { 7 } { 5.5 bp } ,
3793   { -6 } { 6.5 bp } ,
3794   { 6 } { 7.5 bp } ,
3795   { -5 } { 9 bp } ,
3796   { 5 } { 10.5 bp } ,
3797   { -4 } { 12 bp } ,
3798   { 4 } { 14 bp } ,
3799   { -3 } { 15 bp } ,
3800   { 3 } { 16 bp } ,
3801   { -2 } { 18 bp } ,
3802   { 2 } { 22 bp } ,
3803   { -1 } { 24 bp } ,
3804   { 1 } { 26 bp } ,
3805   { -0 } { 36 bp } ,
3806   { 0 } { 42 bp }
3807 }
3808 { \__ctex_save_font_size:nn #1 }

\ctex_declare_math_sizes:nnnn 3809 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3810 {
3811   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3812   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3813 }

__ctex_get_font_sizes:Nn 3814 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3815 {
3816   \tl_clear:N #1
3817   \tl_map_inline:nn {#2}
3818   {
3819     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3820     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3821     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3822   }
3823 }

3824 \clist_map_inline:nn
3825 {
3826   { 8 } { 8 } { 5pt } { 5pt } ,
3827   { 7 } { 7 } { 5pt } { 5pt } ,
3828   { -6 } { -6 } { 5pt } { 5pt } ,
3829   { 6 } { 6 } { 5pt } { 5pt } ,
3830   { -5 } { -5 } { 6pt } { 5pt } ,
3831   { 5 } { 5 } { 7pt } { 5pt } ,
3832   { -4 } { -4 } { 8pt } { 6pt } ,
3833   { 4 } { 4 } { 5 } { 6 } ,
3834   { -3 } { -3 } { -4 } { -5 } ,
3835   { 3 } { 3 } { 4 } { 5 } ,
3836   { -2 } { -2 } { -3 } { -4 } ,
3837   { 2 } { 2 } { 3 } { 4 } ,
3838   { -1 } { -1 } { -2 } { -3 } ,
3839   { 1 } { 1 } { 2 } { 3 } ,
3840   { -0 } { -0 } { -1 } { -2 } ,
3841   { 0 } { 0 } { 1 } { 2 }
3842 }
3843 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3844 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3845 {
3846   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3847   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3848   { \msg_error:nnn { ctex } { fontsize } {#2} }
3849 }

```

```

3850 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3851 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

3852 \if_case:w \g__ctex_font_size_int
3853 \ctex_file_input:n { ctex-c5size.clo }
3854 \or:
3855 \ctex_file_input:n { ctex-cs4size.clo }
3856 \fi:

3857 </class|ctex|ctexsize>

3858 <*c5size>
3859 \ctex_set_font_size:Nnn \normalsize { 5 }
3860 {
3861   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3862   \abovedisplayshortskip \z@ \@plus3\p@
3863   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3864   \belowdisplayskip \abovedisplayskip
3865   \let\@listi\@listI
3866 }
3867 \ctex_set_font_size:Nnn \small { -5 }
3868 {
3869   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3870   \abovedisplayshortskip \z@ \@plus2\p@
3871   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3872   \def\@listi{\leftmargin\leftmargini
3873             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3874             \parsep 2\p@ \@plus\p@ \@minus\p@
3875             \itemsep \parsep}
3876   \belowdisplayskip \abovedisplayskip
3877 }
3878 \ctex_set_font_size:Nnn \footnotesize { 6 }
3879 {
3880   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3881   \abovedisplayshortskip \z@ \@plus\p@
3882   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3883   \def\@listi{\leftmargin\leftmargini
3884             \topsep 3\p@ \@plus\p@ \@minus\p@
3885             \parsep 2\p@ \@plus\p@ \@minus\p@
3886             \itemsep \parsep}
3887   \belowdisplayskip \abovedisplayskip
3888 }
3889 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3890 \ctex_set_font_size:Nnn \tiny { 7 } { }
3891 \ctex_set_font_size:Nnn \large { -4 } { }
3892 \ctex_set_font_size:Nnn \Large { -3 } { }
3893 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3894 \ctex_set_font_size:Nnn \huge { 2 } { }
3895 \ctex_set_font_size:Nnn \Huge { 1 } { }
3896 </c5size>
3897 <*cs4size>
3898 \ctex_set_font_size:Nnn \normalsize { -4 }
3899 {
3900   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3901   \abovedisplayshortskip \z@ \@plus3\p@
3902   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3903   \belowdisplayskip \abovedisplayskip
3904   \let\@listi\@listI
3905 }
3906 \ctex_set_font_size:Nnn \small { 5 }
3907 {
3908   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3909   \abovedisplayshortskip \z@ \@plus3\p@
3910   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3911   \def\@listi{\leftmargin\leftmargini
3912             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3913             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3914             \itemsep \parsep}
3915   \belowdisplayskip \abovedisplayskip

```

```

3916 }
3917 \ctex_set_font_size:Nnn \footnotesize { -5 }
3918 {
3919   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3920   \abovedisplayshortskip \z@ \@plus3\p@
3921   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3922   \def\@listi{\leftmargin\leftmargini
3923             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3924             \parsep 3\p@ \@plus2\p@ \@minus\p@
3925             \itemsep \parsep}
3926   \belowdisplayskip \abovedisplayskip
3927 }
3928 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3929 \ctex_set_font_size:Nnn \tiny { -6 } { }
3930 \ctex_set_font_size:Nnn \large { -3 } { }
3931 \ctex_set_font_size:Nnn \Large { -2 } { }
3932 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3933 \ctex_set_font_size:Nnn \huge { -1 } { }
3934 \ctex_set_font_size:Nnn \Huge { 1 } { }
3935 </cs4size>
<ctexsize> 3936 \normalsize
3937 <*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`。

```

3938 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3939 {
3940   \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。

```

3941   \group_begin: \footnotesize \exp_args:NNNo \group_end:
3942   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3943 }

```

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

```

3944 \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。

```

3945 \NewDocumentCommand \CTeX { }
3946 { C \TeX }
3947 \ctex_at_end_package:nn { hyperref }
3948 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

`captiondelimitter` 过时选项。

```

3949 \keys_define:nn { ctex }
3950 {
3951   captiondelimitter .code:n =

```

```

3952     {
3953       \msg_warning:nnn { ctex } { deprecated-option }
3954       { You~can~load~the~package~`caption'~to~get~its~functionality. }
3955     }
3956   }

3957 </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`

```

3958 <*scheme&chinese&(article|book|report)>
3959 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3960 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3961 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3962 </scheme&chinese&(article|book|report)>

3963 <*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。

```

3964 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3965 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3966 \msg_new:nnn { ctex } { fontset-unavailable }
3967 { 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。

```

3968 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3969 {
3970   \tl_if_empty:NTF \g__ctex_fontset_tl
3971   {
3972     \ctex_detected_platform:
3973     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3974   }
3975   {
3976     \file_if_exist:NTF { ctex-fontset- \g__ctex_fontset_tl .def }
3977     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3978     {
3979       \use:x
3980       {
3981         \ctex_detected_platform:
3982         \msg_error:nxxx { ctex } { fontset-not-found }
3983         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3984       }
3985       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3986     }
3987   }
3988 }
3989 \@onlypreamble \ctex_load_fontset:
3990 \msg_new:nnnn { ctex } { fontset-not-found }
3991 {
3992   CTeX~fontset~`#1'~could~not~be~found.\\
3993   Fontset~`#2'~will~be~used~instead.
3994 }
3995 { You~may~run~`mktexlsr'~firstly. }

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

3996 \keys_define:nn { ctex }
3997 {

```

```

3998 fontset .code:n =
3999 {
4000   \ctex_if_preamble:TF
4001   {
4002     \str_if_eq:eeTF {#1} { none }
4003     { \msg_warning:nnn { ctex } { invalid-value } {#1} }
4004     {
4005       \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
4006       {
4007         \tl_gset:Nx \g__ctex_fontset_tl {#1}
4008         \ctex_load_fontset:
4009       }
4010       {
4011         \msg_error:nnxx { ctex } { fontset-loaded }
4012         { \g__ctex_fontset_tl } {#1}
4013       }
4014     }
4015   }
4016   { \msg_error:nn { ctex } { fontset-only-preamble } }
4017 }
4018 }
4019 \msg_new:nnnn { ctex } { fontset-loaded }
4020 {
4021   CTeX~fontset~`#1'~has~been~loaded.
4022   \str_if_eq:nnF {#1} {#2} { \ \ Fontset~`#2'~will~be~ignored. }
4023 }
4024 { Only~one~fontset~can~be~loaded~in~the~preamble. }
4025 \msg_new:nnn { ctex } { fontset-only-preamble }
4026 {
4027   The~`fontset'~ option~ can~ be~ used~ only~ in~ preamble.
4028 }

```

载入中文字库。

```

4029 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
4030 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

4031 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
4032 </class|ctex>
4033 <*config>
4034 %%
4035 </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。

```

4036 <*ctexopts>
4037 %%
4038 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
4039 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

4040 <*c19|c70>
4041 %%
4042 %% Chinese characters

```

```

4043 %%
<rm>4044 %% character set: GBK (extension of GB 2312)
<c70>4045 %% character set: Unicode
4046 %% font encoding: Unicode
4047 %%
4048 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm><c19>4049 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm><c70>4050 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf><c19>4051 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf><c70>4052 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt><c19>4053 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt><c70>4054 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

up $\text{\LaTeX}$ 使用的字体族。up $\text{\LaTeX}$ 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>4055 \DeclareKanjiFamily{JY2}{zhrm}{\}
<rm><jt2>4056 \DeclareKanjiFamily{JT2}{zhrm}{\}
<sf><jy2>4057 \DeclareKanjiFamily{JY2}{zhshf}{\}
<sf><jt2>4058 \DeclareKanjiFamily{JT2}{zhshf}{\}
<tt><jy2>4059 \DeclareKanjiFamily{JY2}{zhht}{\}
<tt><jt2>4060 \DeclareKanjiFamily{JT2}{zhht}{\}

4061 <*rm>
4062 <*c19>
4063 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
4064 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
4065 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
4066 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
4067 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4068 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4069 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
4070 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
4071 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
4072 </c19>
4073 <*c70>
4074 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
4075 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
4076 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
4077 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
4078 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4079 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4080 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
4081 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
4082 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
4083 </c70>
4084 <*jy2>
4085 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{\}
4086 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{\}
4087 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{\}
4088 </jy2>
4089 <*jt2>
4090 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{\}
4091 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{\}
4092 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{\}
4093 </jt2>
4094 </rm>

4095 <*sf>
4096 <*c19>
4097 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4098 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4099 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4100 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
4101 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4102 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}

```

```

4103 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4104 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4105 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4106 </c19>
4107 <*c70>
4108 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyoud}{\CJKnormal}
4109 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyoud}{\CJKbold}
4110 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyoud}{\CJKbold}
4111 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyoudsl}{\CJKnormal}
4112 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyoudsl}{\CJKbold}
4113 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyoudsl}{\CJKbold}
4114 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyoud}{\CJKnormal}
4115 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyoud}{\CJKbold}
4116 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyoud}{\CJKbold}
4117 </c70>
4118 <*jy2>
4119 \DeclareFontShape{JY2}{zhfs}{m}{n}{<-> upzhfs-h}{ }
4120 \DeclareFontShape{JY2}{zhfs}{bx}{n}{<-> upzhfsb-h}{ }
4121 </jy2>
4122 <*jt2>
4123 \DeclareFontShape{JT2}{zhfs}{m}{n}{<-> upzhfs-v}{ }
4124 \DeclareFontShape{JT2}{zhfs}{bx}{n}{<-> upzhfsb-v}{ }
4125 </jt2>
4126 </sf>
4127 <*tt>
4128 <*c19>
4129 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbksf}{\CJKnormal}
4130 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbksf}{\CJKbold}
4131 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbksf}{\CJKbold}
4132 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbksfsl}{\CJKnormal}
4133 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbksfsl}{\CJKbold}
4134 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbksfsl}{\CJKbold}
4135 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbksf}{\CJKnormal}
4136 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbksf}{\CJKbold}
4137 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbksf}{\CJKbold}
4138 </c19>
4139 <*c70>
4140 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4141 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4142 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4143 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4144 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4145 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4146 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4147 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4148 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4149 </c70>
4150 <*jy2>
4151 \DeclareFontShape{JY2}{zhstt}{m}{n}{<-> upzhmono-h}{ }
4152 </jy2>
4153 <*jt2>
4154 \DeclareFontShape{JT2}{zhstt}{m}{n}{<-> upzhmono-v}{ }
4155 </jt2>
4156 </tt>
4157 <*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。)

```

4158 <*windows>
4159 \file_if_exist:nTF { C:/bootfont.bin }
4160 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
4161 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }

```

4162 </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。

```

4163 <*windowsnew>
4164 \tl_new:N \l__ctex_msyh_suffix_tl
4165 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
4166 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
4167 {
4168   \file_if_exist:nF { msyh.ttc }
4169   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
4170 }
4171 </windowsnew>
4172 <*windowsnew|windowsold>
4173 \sys_if_engine_pdftex:TF
4174 {
4175   \ctex_zhmap_case:nnn
4176   {
4177     \ctex_punct_set:n { windows }
4178     \setCJKmainfont
4179     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
4180 <*windowsold>
4181     \setCJKsansfont { simhei.ttf }
4182     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4183 </windowsold>
4184 <*windowsnew>
4185     \setCJKsansfont
4186     [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4187     \setCJKfamilyfont { zhyahei }
4188     [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4189     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4190     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
4191 </windowsnew>
4192     \setCJKmonofont { simfang.ttf }
4193     \setCJKfamilyfont { zhkai } { simkai.ttf }
4194     \setCJKfamilyfont { zhfs } { simfang.ttf }
4195     \setCJKfamilyfont { zhsong } { simsun.ttc }
4196     \setCJKfamilyfont { zhhei } { simhei.ttf }
4197     \setCJKfamilyfont { zhli } { simli.ttf }
4198     \setCJKfamilyfont { zhyou } { simyou.ttf }
4199     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4200     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4201     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4202     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4203   }
4204   {
4205     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
4206     \ctex_punct_set:n { windows }
4207     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4208     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4209     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4210   }
4211   {
4212     \tl_set:Nn \CJKrmdefault { rm }
4213     \tl_set:Nn \CJKsfdefault { sf }
4214     \tl_set:Nn \CJKttdefault { tt }
4215   }
4216 }
4217 {
4218   \sys_if_engine_uptex:TF
4219   {
4220     \ctex_set_upfonts:nnnnnn
4221     {simsun.ttc} {simhei.ttf} {simkai.ttf}
4222     {msyh\l__ctex_msyh_suffix_tl} {msyhbd\l__ctex_msyh_suffix_tl}
4223     {simhei.ttf} {simhei.ttf}

```

```

4224         {simfang.ttf}
4225         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4226         \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4227         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4228         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
<windowsnew> 4229         \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4230         \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4231         \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4232         \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4233         \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4234     }
4235     {
4236 <*windowsold>
4237         \setCJKmainfont
4238         [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
4239         \setCJKsansfont { SimHei }
4240         \setCJKmonofont { FangSong_GB2312 }
4241         \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
4242         \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
4243 </windowsold>
4244 <*windowsnew>
4245         \setCJKmainfont
4246         [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
4247         \setCJKsansfont
4248         [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4249         \setCJKmonofont { FangSong }
4250         \setCJKfamilyfont { zhkai } { KaiTi }
4251         \setCJKfamilyfont { zhfs } { FangSong }
4252 </windowsnew>
4253         \setCJKfamilyfont { zhsong } { SimSun }
4254         \setCJKfamilyfont { zhhei } { SimHei }
4255         \setCJKfamilyfont { zhli } { LiSu }
4256         \setCJKfamilyfont { zhyou } { YouYuan }
4257 <*windowsnew>
4258         \setCJKfamilyfont { zhyahei }
4259         [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4260 </windowsnew>
4261     }
4262 }
4263 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```

4264 <*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。

```

4265 \sys_if_engine_pdftex:TF
4266 {
4267     \sys_if_output_pdf:TF
4268     { \ctex_fontset_error:n { adobe } }
4269     {
4270         \ctex_zhmap_case:nnn
4271         {
4272             \setCJKmainfont
4273             [
4274                 cmap = UniGB-UTF16-H ,
4275                 BoldFont = AdobeHeitiStd-Regular.otf ,
4276                 ItalicFont = AdobeKaitiStd-Regular.otf
4277             ] { AdobeSongStd-Light.otf }
4278             \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4279             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4280             \setCJKfamilyfont { zhsong }
4281             [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
4282             \setCJKfamilyfont { zhhei }

```

```

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

```

14.15.4 ctex-fontset-fandol.def

```

4333 \sys_if_engine_pdftex:TF
4334 {
4335     \sys_if_output_pdf:TF
4336     { \ctex_fontset_error:n { fandol } }
4337     {
4338         \ctex_zhmap_case:nnn
4339         {
4340             \setCJKmainfont
4341             [
4342                 cmap = UniGB-UTF16-H ,
4343                 BoldFont = FandolSong-Bold.otf ,
4344                 ItalicFont = FandolKai-Regular.otf
4345             ] { FandolSong-Regular.otf }
4346             \setCJKsansfont

```

```

4347         [
4348             cmap = UniGB-UTF16-H ,
4349             BoldFont = FandolHei-Bold.otf
4350         ] { FandolHei-Regular.otf }
4351     \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4352     \setCJKfamilyfont { zhsong }
4353         [
4354             cmap = UniGB-UTF16-H ,
4355             BoldFont = FandolSong-Bold.otf
4356         ] { FandolSong-Regular.otf }
4357     \setCJKfamilyfont { zhhei }
4358         [
4359             cmap = UniGB-UTF16-H ,
4360             BoldFont = FandolHei-Bold.otf
4361         ] { FandolHei-Regular.otf }
4362     \setCJKfamilyfont { zhfs }
4363         [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4364     \setCJKfamilyfont { zhkai }
4365         [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4366     \ctex_punct_set:n { fandol }
4367     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4368     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4369     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4370     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4371     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4372     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4373 }
4374 {
4375     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4376     \ctex_punct_set:n { fandol }
4377     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4378     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4379     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4380 }
4381 { \ctex_fontset_error:n { fandol } }
4382 }
4383 }
4384 {
4385     \sys_if_engine_uptex:TF
4386     {
4387         \ctex_set_upfonts:nnnnnn
4388         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4389         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4390         {FandolFang-Regular.otf}
4391         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4392         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4393         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4394         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4395     }
4396     {
4397         \setCJKmainfont
4398         [
4399             Extension = .otf ,
4400             BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular
4401         ]
4402         { FandolSong-Regular }
4403     \setCJKsansfont
4404         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4405     \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4406     \setCJKfamilyfont { zhsong }
4407         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4408     \setCJKfamilyfont { zhhei }
4409         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4410     \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4411     \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4412 }
4413 }

```

```
4414 </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。

```
4415 <*mac>
4416 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
4417 { \ctex_file_input:n { ctex-fontset-macnew.def } }
4418 { \ctex_file_input:n { ctex-fontset-macold.def } }
4419 </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 的字族设置也有同样的情况。

```
4420 <*macold|macnew>
4421 \sys_if_engine_pdftex:TF
4422 { \ctex_fontset_error:n { mac } }
4423 {
4424   \sys_if_engine_uptex:TF
4425   { \ctex_fontset_error:n { mac } }
4426   {
4427     <*macold>
4428     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4429     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4430   </macold>
4431   <*macnew>
4432   \setCJKmainfont
4433   [
4434     UprightFont = *~Light ,
4435     BoldFont = *~Bold ,
4436     ItalicFont = Kaiti~SC ,
4437     BoldItalicFont = Kaiti~SC~Bold
```

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

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

```

4438         ] { Songti~SC }
4439         \setCJKsansfont { PingFang~SC }
4440 </macnew>
4441         \setCJKmonofont { STFangsong }
4442 <*macold>
4443         \setCJKfamilyfont { zhsong } { STSong }
4444         \setCJKfamilyfont { zhhei } { STHeiti }
4445 </macold>
4446 <*macnew>
4447         \setCJKfamilyfont { zhsong }
4448         [
4449             UprightFont = *~Light ,
4450             BoldFont = *~Bold ,
4451         ] { Songti~SC }
4452         \setCJKfamilyfont { zhhei } { Heiti~SC }
4453         \setCJKfamilyfont { zhpfs } { PingFang~SC }
4454 </macnew>
4455         \setCJKfamilyfont { zhfs } { STFangsong }
4456 <macold> \setCJKfamilyfont { zhkai } { STKaiti }
4457 <macnew> \setCJKfamilyfont { zhkai } { Kaiti~SC }
4458     }
4459 }
4460 </macold|macnew>

```

14.15.6 ctex-fontset-founder.def

```

4461 <*founder>
4462 \sys_if_engine_pdftex:TF
4463 {
4464     \ctex_zhmap_case:nnn
4465     {
4466         \setCJKmainfont
4467         [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4468         \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4469         \setCJKmonofont { FZFSK.TTF }
4470         \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4471         \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4472         \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4473         \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4474         \setCJKfamilyfont { zhli } { FZLSK.TTF }
4475         \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4476         \ctex_punct_set:n { founder }
4477         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4478         \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4479         \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4480         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4481         \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4482         \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4483         \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4484     }
4485     {
4486         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4487         \ctex_punct_set:n { founder }
4488         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4489         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4490         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4491     }
4492     { \ctex_fontset_error:n { founder } }
4493 }
4494 {
4495     \sys_if_engine_uptex:TF
4496     {
4497         \ctex_set_upfonts:nnnnnn
4498         { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4499         { FZXH1K.TTF } { FZHTK.TTF }
4500         { FZFSK.TTF }

```

```

4501 \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4502 \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4503 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4504 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4505 \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4506 \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4507 \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4508 \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4509 }
4510 {
4511 \setCJKmainfont
4512 [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }
4513 \setCJKsansfont [ BoldFont = FZHei-B01 ] { FZXiHeiI-Z08 }
4514 \setCJKmonofont { FZFangSong-Z02 }
4515 \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4516 \setCJKfamilyfont { zhhei } { FZHei-B01 }
4517 \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4518 \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4519 \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4520 \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4521 }
4522 }
4523 </founder>
4524 <*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。

```

4525 \sys_if_engine_pdftex:TF
4526 {
4527 \ctex_zhmap_case:nnn
4528 {
4529 \setCJKmainfont
4530 [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4531 \setCJKsansfont { wqy-zenhei.ttc }
4532 \setCJKmonofont { uming.ttc }
4533 \setCJKfamilyfont { zhsong } { uming.ttc }
4534 \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4535 \setCJKfamilyfont { zhkai } { ukai.ttc }
4536 \ctex_punct_set:n { ubuntu }
4537 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4538 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4539 \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4540 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4541 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4542 }
4543 {
4544 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4545 \ctex_punct_set:n { ubuntu }
4546 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4547 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4548 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4549 }
4550 { \ctex_fontset_error:n { ubuntu } }
4551 }
4552 {
4553 \sys_if_engine_uptex:TF
4554 {

```

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

```

4555         \ctex_set_upfonts:nnnnnn
4556         { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4557         { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4558         { uming.ttc }
4559         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4560         \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4561         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4562     }
4563     {
4564         \setCJKmainfont
4565         [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4566         \setCJKsansfont { WenQuanYi~Zen~Hei }
4567         \setCJKmonofont { AR~PL~UMing~CN }
4568         \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4569         \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4570         \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4571     }
4572 }
4573 </ubuntu>

```

14.15.8 中文字体命令

```

4574 <!*windows&!mac>
4575 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4576 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
</ubuntu> 4577 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4578 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4579 <*windowsnew|windowsold|founder>
4580 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4581 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4582 </windowsnew|windowsold|founder>
<windowsnew> 4583 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4584 <*macnew>
4585 \NewDocumentCommand \yahei { } { \CJKfamily { zhpf } }
4586 \NewDocumentCommand \pingfang { } { \CJKfamily { zhpf } }
4587 </macnew>
4588 </!windows&!mac>
4589 </fontset>
4590 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4591 \begingroup\catcode61\catcode48\catcode32=10\relax%
4592 \catcode 35=6 % #
4593 \catcode123=1 % {
4594 \catcode125=2 % }
4595 \toks0{\endlinechar=\the\endlinechar\relax}%
4596 \toks2{\endlinechar=13 }%
4597 \def\x#1 #2 {%
4598     \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4599     \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4600 \x 13 5 % carriage return
4601 \x 32 10 % space
4602 \x 35 6 % #
4603 \x 40 12 % (
4604 \x 41 12 % )
4605 \x 45 12 % -
4606 \x 46 12 % .
4607 \x 47 12 % /
4608 \x 58 12 % :
4609 \x 60 12 % <
4610 \x 61 12 % =
4611 \x 64 11 % @

```

```

4612 \x 91 12 % [
4613 \x 93 12 % ]
4614 \x 123 1 % {
4615 \x 125 2 % }
4616 \edef\x#1{\endgroup%
4617 \edef\noexpand#1{%
4618 \the\toks0 %
4619 \let\noexpand\noexpand\noexpand#1%
4620 \noexpand\noexpand\noexpand\noexpand\undefined%
4621 \noexpand\noexpand\noexpand\endinput}%
4622 \the\toks2}%
4623 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4624 \begingroup\expandafter\endgroup
4625 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4626 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4627 \expandafter\ifx\csname pdfoutput\endcsname\relax
4628 \iffalse\else\ifnum\pdfoutput < 1 \iffalse\else \iftrue\fi\fi
4629 \else ifctexpdf\fi
4630 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4631 \begingroup
4632 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4633 \long\def\x#1\ProvidesFile#2[#3]{%
4634 #1%
4635 \immediate\write-1{File: #2 #3}%
4636 \expandafter\xdef\csname ver@#2\endcsname{#3}}
4637 \expandafter\x%
4638 \fi
4639 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4640 <*windows>
4641 \ProvidesFile{zhwindowsfonts.tex}%
4642 [2019/04/07 v2.4.15 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4643
4644 \ifzhmappdf
4645 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4646 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4647 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4648 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4649 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4650 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4651 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4652
4653 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4654 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4655 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4656 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4657 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4658 \pdfmapline{=unili@Unicode@ <simli.ttf}
4659 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4660
4661 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4662 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4663 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4664 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4665 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4666 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4667
4668 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4669 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4670 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4671 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}

```

```

4672 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4673 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4674
4675 \else
4676 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4677 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4678 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4679 \special{pdf:mapline gbkkhei@UGBK@ unicode simhei.ttf -v 150}
4680 \special{pdf:mapline gbkkfs@UGBK@ unicode simfang.ttf -v 50}
4681 \special{pdf:mapline gbkkli@UGBK@ unicode simli.ttf -v 150}
4682 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4683
4684 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4685 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4686 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4687 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4688 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4689 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4690 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4691
4692 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4693 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4694 \special{pdf:mapline gbkkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4695 \special{pdf:mapline gbkkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4696 \special{pdf:mapline gbkkli@UGBK@ unicode simli.ttf -s .167 -v 150}
4697 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4698
4699 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4700 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4701 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4702 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4703 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4704 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -s .167 -v 60}
4705
4706 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4707 <*adobe>
4708 \ProvidesFile{zhadobefonts.tex}%
4709 [2019/04/07 v2.4.15 Adobe font map loader for DVIPDFMx (CTEX)]
4710
4711 \ifzhmappdf
4712 %% pdfTeX does not support OTF fonts
4713
4714 \else
4715 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4716 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4717 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4718 \special{pdf:mapline gbkkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4719 \special{pdf:mapline gbkkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4720
4721 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4722 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4723 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4724 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4725 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4726
4727 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4728 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4729 \special{pdf:mapline gbkkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4730 \special{pdf:mapline gbkkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4731
4732 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4733 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4734 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4735 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4736

```

4737 </adobe>

14.15.9.3 zhfandolfonts.tex

```

4738 <*fandol>
4739 \ProvidesFile{zhfandolfonts.tex}%
4740 [2019/04/07 v2.4.15 Fandol font map loader for DVIPDFMx (CTEX)]
4741
4742 \ifzhmappdf
4743 %% pdfTeX does not support OTF fonts
4744
4745 \else
4746 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4747 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4748 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4749 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4750 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4751
4752 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4753 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4754 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4755 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4756 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4757
4758 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4759 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4760 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4761 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4762
4763 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4764 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4765 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4766 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4767
4768 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4769 <*founder>
4770 \ProvidesFile{zhfounderfonts.tex}%
4771 [2019/04/07 v2.4.15 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4772
4773 \ifzhmappdf
4774 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4775 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4776 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4777 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4778 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4779 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4780 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4781
4782 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4783 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4784 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4785 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4786 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4787 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4788 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4789
4790 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4791 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4792 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4793 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4794 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4795 \pdfmapline{=gbkyouisl@UGBK@ <FZY1K.TTF}
4796
4797 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4798 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}

```

```

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

```

14.15.9.5 zhubuntufonts.tex

```

4836 <*ubuntu>
4837 \ProvidesFile{zhubuntufonts.tex}%
4838 [2019/04/07 v2.4.15 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4839
4840 \ifzhmappdf
4841 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4842 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4843 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4844 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4845 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4846 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4847
4848 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4849 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4850 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4851 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4852 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4853 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4854
4855 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4856 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4857 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4858 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4859 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4860
4861 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4862 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4863 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}

```

```

4864 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4865 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4866
4867 \else
4868 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4869 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4870 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4871 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4872 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4873
4874 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4875 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4876 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4877 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4878 \special{pdf:mapline unifss@Unicode@ unicode :0:uming.ttc}
4879
4880 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4881 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4882 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4883 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4884
4885 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4886 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4887 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4888 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4889
4890 </ubuntu>
4891 \fi
4892
4893 \ctex@zhmap@endinput
4894 </zhmap>

```

14.15.10 制作 spa 文件

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

```

4895 <*spa>
4896 <*macro>
4897 \input expl3-generic %
4898 \ExplSyntaxOn
4899 \sys_if_engine_xetex:F
4900 {
4901   \msg_new:nn { ctex } { xetex }
4902   { XeTeX~is~required~to~compile~this~document! }
4903   \msg_fatal:nn { ctex } { xetex }
4904 }

```

CJKpunct 定义的标点符号是:

“「『（（〔{〈《【

—…、。：：！？%）〕}〉》】’”| |

注意顺序不能改变。

```

4905 \seq_const_from_clist:Nn \c__ctex_punct_seq
4906 {
4907     "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4908     "3008 , "300A , "3016 , "3010 ,
4909     "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4910     "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4911     "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4912 }

```

`\ctex_make_spa:nn` #1 是 spa 文件名, #2 是由 CIK 族名与字体构成的逗号列表。

4913 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2

```

4914 {
4915   \iow_open:Nn \g__ctex_spa_iow {#1}
4916   \clist_map_inline:nn {#2}
4917     { \__ctex_write_family:nn ##1 }
4918   \iow_close:N \g__ctex_spa_iow
4919 }
4920 \iow_new:N \g__ctex_spa_iow
4921 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

4922 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4923 {
4924   \group_begin:
4925     \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4926     \l__ctex_punct_font
4927     \clist_clear:N \l__ctex_punct_bounds_clist
4928     \seq_map_inline:Nn \c__ctex_punct_seq
4929       { \exp_args:No \__ctex_save_bounds:n { \int_use:N \tex_XeTeXcharglyph:D ##1 } }
4930     \iow_now:Nx \g__ctex_spa_iow
4931     {
4932       \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4933       { \l__ctex_punct_bounds_clist , , , }
4934     }
4935   \group_end:
4936 }
4937 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4938 {
4939   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4940     {
4941       \__ctex_calc_bounds:nn { 1 } {#1} ,
4942       \__ctex_calc_bounds:nn { 3 } {#1}
4943     }
4944 }
4945 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1em 的比值的一百倍。

```

4946 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4947 {
4948   \fp_eval:n
4949     {
4950       round
4951       (
4952         \dim_to_decimal_in_unit:nn
4953           { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
4954           { 1 em }
4955       )
4956     }
4957 }
4958 \ExplSyntaxOff
4959 </macro>

```

下面是 CTeX 定义的一些字体。

```

4960 <*make>
4961 \input ctexspamacro %
4962
4963 \MAKESPA {ctexpunct.spa}
4964 {
4965   {adobezhsong}      {AdobeSongStd-Light} ,
4966   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4967   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4968   {adobezhfs}        {AdobeFangsongStd-Regular} ,
4969   {fandolzhsong}     {FandolSong} ,
4970   {fandolzhsongb}    {FandolSong-Bold} ,
4971   {fandolzhhei}      {FandolHei} ,
4972   {fandolzhheib}     {FandolHei-Bold} ,

```

```

4973 {fandolzhkai} {FandolKai} ,
4974 {fandolzhfs} {FandolFang} ,
4975 {founderzhsong} {FZShuSong-Z01} ,
4976 {founderzhsongb} {FZXiaoBiaoSong-B05} ,
4977 {founderzhhei} {FZHei-B01} ,
4978 {founderzhheil} {FZXiHeiI-Z08} ,
4979 {founderzhkai} {FZKai-Z03} ,
4980 {founderzhfs} {FZFangSong-Z02} ,
4981 {founderzhli} {FZLiShu-S01} ,
4982 {founderzhyou} {FZXiYuan-M01} ,
4983 {founderzhyoub} {FZZhunYuan-M02} ,
4984 {ubuntuzhsong} {AR PL Uming CN} ,
4985 {ubuntuzhhei} {WenQuanYi Zen Hei} ,
4986 {ubuntuzhkai} {AR PL UKai CN} ,
4987 {windowszhsong} {SimSun} ,
4988 {windowszhhei} {SimHei} ,
4989 {windowszhkai} {KaiTi} ,
4990 {windowszhfs} {FangSong} ,
4991 {windowszhli} {LiSu} ,
4992 {windowszhyou} {YouYuan} ,
4993 {windowszhyawei} {Microsoft YaHei} ,
4994 {windowszhyawei} {Microsoft YaHei Bold}
4995 }
4996
4997 \primitive\end
4998 </make>
4999 </spa>

```

14.16 translator 宏包的中文字典

```
5000 <*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。

```

5001 <*theorem>
5002 \providetranslation{Comments}{评论}
5003 \providetranslation{comments}{评论}
5004 \providetranslation{Comment}{评论}
5005 \providetranslation{comment}{评论}
5006 \providetranslation{Corollaries}{推论}
5007 \providetranslation{corollaries}{推论}
5008 \providetranslation{Corollary}{推论}
5009 \providetranslation{corollary}{推论}
5010 \providetranslation{Definitions}{定义}
5011 \providetranslation{definitions}{定义}
5012 \providetranslation{Definition}{定义}
5013 \providetranslation{definition}{定义}
5014 \providetranslation{Examples}{例}
5015 \providetranslation{examples}{例}
5016 \providetranslation{Example}{例}
5017 \providetranslation{example}{例}
5018 \providetranslation{Exercises}{练习}
5019 \providetranslation{exercises}{练习}
5020 \providetranslation{Exercise}{练习}
5021 \providetranslation{exercise}{练习}
5022 \providetranslation{Facts}{事实}
5023 \providetranslation{facts}{事实}
5024 \providetranslation{Fact}{事实}
5025 \providetranslation{fact}{事实}
5026 \providetranslation{Key Lemmas}{关键引理}
5027 \providetranslation{key lemmas}{关键引理}
5028 \providetranslation{Key Lemma}{关键引理}
5029 \providetranslation{key lemma}{关键引理}
5030 \providetranslation{Key Observations}{关键观察}
5031 \providetranslation{key observations}{关键观察}

```

```

5032 \providetranslation{Key Observation}{关键观察}
5033 \providetranslation{key observation}{关键观察}
5034 \providetranslation{Lemmas}{引理}
5035 \providetranslation{lemmas}{引理}
5036 \providetranslation{Lemma}{引理}
5037 \providetranslation{lemma}{引理}
5038 \providetranslation{Main Theorems}{主要定理}
5039 \providetranslation{main theorems}{主要定理}
5040 \providetranslation{Main Theorem}{主要定理}
5041 \providetranslation{main theorem}{主要定理}
5042 \providetranslation{Observations}{观察}
5043 \providetranslation{observations}{观察}
5044 \providetranslation{Observation}{观察}
5045 \providetranslation{observation}{观察}
5046 \providetranslation{Problems}{问题}
5047 \providetranslation{problems}{问题}
5048 \providetranslation{Problem}{问题}
5049 \providetranslation{problem}{问题}
5050 \providetranslation{Proofs}{证明}
5051 \providetranslation{proofs}{证明}
5052 \providetranslation{Proof}{证明}
5053 \providetranslation{proof}{证明}
5054 \providetranslation{Proof Sketch}{证明提要}
5055 \providetranslation{Proof sketch}{证明提要}
5056 \providetranslation{proof sketch}{证明提要}
5057 \providetranslation{Proof Sketches}{证明提要}
5058 \providetranslation{Proof sketches}{证明提要}
5059 \providetranslation{proof sketches}{证明提要}
5060 \providetranslation{Sketch of Proof}{证明提要}
5061 \providetranslation{Sketch of Proofs}{证明提要}
5062 \providetranslation{Sketch of proof}{证明提要}
5063 \providetranslation{Sketch of proofs}{证明提要}
5064 \providetranslation{sketch of proof}{证明提要}
5065 \providetranslation{sketch of proofs}{证明提要}
5066 \providetranslation{Propositions}{命题}
5067 \providetranslation{propositions}{命题}
5068 \providetranslation{Proposition}{命题}
5069 \providetranslation{proposition}{命题}
5070 \providetranslation{Remarks}{注}
5071 \providetranslation{remarks}{注}
5072 \providetranslation{Remark}{注}
5073 \providetranslation{remark}{注}
5074 \providetranslation{Solutions}{解}
5075 \providetranslation{solutions}{解}
5076 \providetranslation{Solution}{解}
5077 \providetranslation{solution}{解}
5078 \providetranslation{Theorems}{定理}
5079 \providetranslation{theorems}{定理}
5080 \providetranslation{Theorem}{定理}
5081 \providetranslation{theorem}{定理}
5082 </theorem>
5083 </dict>

```

14.17 ctexcap 宏包

```
5084 <*ctexcap>
```

ctexcap 是过时宏包。

```

5085 \clist_new:N \l__ctex_ctexcap_options_clist
5086 \clist_set:Nx \l__ctex_ctexcap_options_clist
5087 { \exp_not:v { opt@ \@currname . \@current } , heading }
5088 \msg_new:nnn { ctexcap } { deprecated }
5089 {
5090   Package~`ctexcap'~is~deprecated.\\
5091   Please~use~package~`ctex'~with~option~`#1'~instead: \\\
5092   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5093 }

```

```

5094 \msg_warning:nnx { ctexcap } { deprecated }
5095 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

5096 \PassOptionsToPackage { heading = true } { ctexcap }
5097 \RequirePackageWithOptions { ctex }
5098 </ctexcap>

```

14.18 ctexhook 宏包

```

5099 <*ctexhook>

```

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
5100 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
5101 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
5102 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
5103 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
5104 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
5105 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5106 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
5107 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5108 \cs_set_nopar:Npx \document
5109 {
5110   \CTEX@document@left@hook
5111   \exp_not:o { \document }
5112   \CTEX@document@right@hook
5113 }
5114 \tl_new:N \g__ctex_end_preamble_hook_tl
5115 \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。

```

5116 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
5117 {
5118   \@ifpackageloaded {#1}
5119   {#2}
5120   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
5121 }

```

给钩子附加内容。

```

\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn
5122 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
5123 {
5124   \tl_if_exist:NF #1 { \tl_new:N #1 }
5125   \tl_gput_right:Nn #1 {#2}
5126 }
5127 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

宏包末尾钩子, 只执行一次, 用后清除。

```

\ctex_package_end_hook:n
\ctex_package_end_hook:o
5128 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
5129 {
5130   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
5131   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
5132 }
5133 \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。

```

5134 \tl_put_left:Nn \@popfilename
5135 {
5136   \cs_if_eq:NNT \@currentx \pkgextension
5137   { \ctex_package_end_hook:o { \@currname } }
5138 }

```

5139 `</ctexhook>`

14.19 ctexpatch 宏包

5140 `<*ctexpatch>`

5141 `\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。

```
5142 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
5143 {
5144   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnw #1
5145   { once } {#2} { \use_i:nn } { \use_ii:nn }
5146 }
```

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

```
5147 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5148 {
5149   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnw #1
5150   { all } {#2} { \use_i:nn } { \use_ii:nn }
5151 }
```

`\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。

```
5152 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5153 {
5154   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnw #1
5155   { once }
5156   {
5157     \ExplSyntaxOff
5158     \char_set_catcode_letter:n { 64 }
5159   }
5160   { }
5161   { \ctex_patch_failure:N #1 }
5162 }
5163 \cs_new_protected:Npn \ctex_patch_failure:N #1
5164 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5165 \msg_new:nnn { ctex } { patch-failure }
5166 { Oops!~Command~`#1'~is~NOT~patchable.\\ }
```

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

```
5167 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5168 {
5169   \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
5170   { left } {#2} { \use_i:nn } { \use_ii:nn }
5171 }
```

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

```
5172 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5173 {
5174   \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
5175   { right } {#2} { \use_i:nn } { \use_ii:nn }
5176 }
```

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

```
5177 \cs_new_protected:Npn \ctex_patch_boot:NnnnTF #1#2#3#4#5#6
5178 {
5179   \tl_set:Nn \__ctex_patch_true:w {#5}
5180   \tl_set:Nn \__ctex_patch_false:w {#6}
```

```

5181 \group_begin:
5182 \char_set_catcode_other:n { 35 }
5183 \ctex_parse_name:NN #1 #2 {#3} {#4}
5184 }
5185 \tl_new:N \__ctex_patch_true:w
5186 \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。

```

5187 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5188 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5189 \group_begin:
5190 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5191 {
5192   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5193   {
5194     \bool_lazy_or:nnTF
5195     { \cs_if_exist_p:c { ##3 ~ } }
5196     { \cs_if_exist_p:c { #1##3 } }
5197     {
5198       \group_begin:
5199       \use:x
5200       {
5201         \group_end:
5202         \__ctex_parse_name:nNNNnN
5203         { \cs_replacement_spec:N ##2 }
5204         \exp_not:N ##2
5205         \exp_not:c { ##3 ~ }
5206         \exp_not:c { #1##3 }
5207       } {##3} ##1
5208     }
5209     { ##1##2 }
5210   }
5211   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5212   {
5213     \exp_args:Nc ##6
5214     {
5215       \str_case:nnTF {##1}
5216       {
5217         { \protect ##3 } { }
5218         { \x@protect ##2 \protect ##3 } { }
5219       }
5220       {
5221         \str_if_eq:eeTF
5222         { \exp_not:n { #1@protected@ ##3 #1##3 } }
5223         {
5224           \exp_last_unbraced:Nf \__ctex_parse_name:w
5225           \cs_replacement_spec:N ##3 #3 ~ #2 \q_stop
5226         }
5227         { #1##5 ~ } { ##5 ~ }
5228       }
5229     }
5230   }

```

```

5229         {
5230             \str_case:onTF { \_ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5231             {
5232                 { #1@protected@ ##2 ##4 } { }
5233                 { #1@ ##4 } { }
5234             }
5235             { #1##5 } {##5}
5236         }
5237     }
5238 }
5239 \cs_new:Npn \_ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5240 }
5241 \use:x
5242 {
5243     \_ctex_tmp:w
5244     { \c_backslash_str }
5245     { \c_left_brace_str }
5246     { \tl_to_str:n { testopt } }
5247 }
5248 \group_end:
5249 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

```

\_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。

```

5250 \str_new:N \_l\_ctex_prefix_str
5251 \str_new:N \_l\_ctex_parameter_str
5252 \str_new:N \_l\_ctex_replacement_str

```

```

\ctex_get_macro_meaning:NTF
\_ctex_get_macro_meaning:w

```

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

```

5253 \group_begin:
5254 \cs_set_protected:Npn \_ctex_tmp:w #1
5255 {
5256     \prg_new_protected_conditional:Npnn
5257     \ctex_get_macro_meaning:N ##1 { TF }
5258     {
5259         \exp_after:wN \_ctex_get_macro_meaning:w
5260         \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5261     }
5262     \cs_new_protected:Npn \_ctex_get_macro_meaning:w
5263     ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5264     {
5265         \tl_if_empty:nTF { ##4 }
5266         { \prg_return_false: }
5267         {
5268             \str_set:Nn \_l\_ctex_prefix_str { ##1 }
5269             \str_set:Nn \_l\_ctex_parameter_str { ##2 }
5270             \str_set:Nn \_l\_ctex_replacement_str { ##3 }
5271             \prg_return_true:
5272         }
5273     }
5274 }
5275 \exp_args:No \_ctex_tmp:w { \tl_to_str:n { macro: } }
5276 \group_end:

```

```

\ctex_if_rescanable:NnTF

```

检查宏是否可以重建。

```

5277 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5278 {
5279     \ctex_get_macro_meaning:NTF #1
5280     {
5281         \_ctex_patch_rebuild:Nn \_ctex_rebuild_cmd:w {#2}
5282         \cs_if_eq:NNTF #1 \_ctex_rebuild_cmd:w {#3} {#4}
5283     }
5284     {#4}
5285 }
5286 \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。

```

5287 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5288 {
5289   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl      \l__ctex_prefix_str      {#2}
5290   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl   \l__ctex_parameter_str   {#2}
5291   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5292   \use:x
5293   {
5294     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5295     \exp_not:o { \l__ctex_parameter_tl }
5296     { \exp_not:o { \l__ctex_replacement_tl } }
5297   }
5298 }
5299 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5300 {
5301   \str_if_empty:NTF #2
5302   { \tl_clear:N #1 }
5303   { \tl_set_rescan:Nno #1 {#3} {#2} }
5304 }
5305 \tl_new:N \l__ctex_prefix_tl
5306 \tl_new:N \l__ctex_parameter_tl
5307 \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。

```

5308 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5309 {
5310   \group_end:
5311   \ctex_if_rescanable:NnTF #1 {#3}
5312   {
5313     \use:x
5314     {
5315       \__ctex_patch_replace:nnnTF {#2}
5316       { \tl_to_str:n {#4} }
5317       { \tl_to_str:n {#5} }
5318     }
5319     {
5320       \__ctex_patch_rebuild:Nn #1 {#3}
5321       \__ctex_patch_true:w
5322     }
5323     { \__ctex_patch_false:w }
5324   }
5325   { \__ctex_patch_false:w }
5326 }

```

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

```

5327 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5328 {
5329   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5330   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5331 }

```

`\__ctex_hookto_cmd:Nnnnw` 在宏的前/后附加钩子。

```

5332 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5333 {
5334   \group_end:
5335   \ctex_get_macro_meaning:NTF #1
5336   {
5337     \str_if_empty:NTF \l__ctex_parameter_str
5338     { \__ctex_hookto_cmd_parameterless:Nnnnw }
5339     { \__ctex_hookto_cmd_parameter:Nnnnw }
5340     #1 {#2} {#3} {#4}
5341   }
5342   { \__ctex_patch_false:w }
5343 }

```

_ctex_hookto_cmd_parameterless:Nnnnw

如果宏没有参数,可以直接进行附加操作。注意保持宏的前缀。

```

5344 \cs_new_protected:Npn \_ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5345 {
5346   \str_if_empty:NF \l__ctex_prefix_str
5347   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5348   \tex_edef:D #1
5349   {
5350     \use:c { __ctex_ #2 _hook_aux:nn }
5351     { \exp_not:o {#1} }
5352     { \exp_not:n {#4} }
5353   }
5354   \_ctex_patch_true:w
5355 }
5356 \cs_generate_variant:Nn \tl_rescan:nn { no }
5357 \cs_new:Npn \_ctex_left_hook_aux:nn #1#2 { #2#1 }
5358 \cs_new_eq:NN \_ctex_right_hook_aux:nn \use:nn

```

_ctex_hookto_cmd_parameter:Nnnnw

如果宏有参数,需要在字符串中进行附加,然后再重建。

```

5359 \cs_new_protected:Npn \_ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5360 {
5361   \_ctex_patch_rebuild:Nn \_ctex_rebuild_cmd:w {#3}
5362   \cs_if_eq:NNTF #1 \_ctex_rebuild_cmd:w
5363   {
5364     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5365     \_ctex_patch_rebuild:Nn #1 {#3}
5366     \_ctex_patch_true:w
5367   }
5368   { \_ctex_patch_false:w }
5369 }
5370 </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。	112
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。	41
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。	82
\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。	86
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。	83
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。	39
新增 linestretch 选项。	80
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。	79
应用 [®] TeX3 重新整理代码。	1
中文字号不再采用近似值。	109
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。	46

v2.0.1 (2015/05/15)

General: 修复 10pt、11pt 等选项无效的问题。	42
---	----

v2.0.2 (2015/05/16)

General: 修复加载 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。	85
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。	85
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。	85
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。	85
新的标题格式选项 numbering。	85
新的标题格式选项 runin。	85
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 [®] 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。	116
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。	86
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)	v2.4.9	(2017/02/27)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	50	General: 调整 <code>unicode-math</code> 补丁的代码顺序。	55
<code>\em</code> : 兼容 <code>upL^AT_EX 2016/05/07</code> 的定义。	69	v2.4.10	(2017/07/19 – 2017/07/23)
v2.4.3	(2016/06/03 – 2016/08/26)	General: 常数 <code>\c_minus_one</code> 已过时。	35
General: 更新 <code>unicode-math</code> 补丁。	55	定义 <code>\cht</code> , <code>\cdp</code> 和 <code>\c wd</code> 。	77
简化 <code>fontspec</code> 补丁。	50	使用 <code>lazy</code> 函数对 <code>Boolean</code> 表达式进行最小化运算 (<code>L^AT_EX 3 2017/07/19</code>)。	35
确保 <code>\proofname</code> 非空。	83	v2.4.11	(2017/08/17 – 2017/11/21)
删除选项 <code>part/fixbeforeskip</code> 和 <code>chapter/fixbeforeskip</code> 。	86	General: 不把 <code>Enclosed Alphanumerics</code> 设置为 <code>JAchar</code> 。	49
新的标题格式选项 <code>fixskip</code> 。	85	不把希腊和西里尔字母设置为 <code>JAchar</code> 。	50
v2.4.4	(2016/09/09 – 2016/09/19)	因上游 <code>l3keys</code> 变化, 重新定义 <code>format_L+</code> 等带空格加号 的选项。	85
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	49	<code>\ps@headings</code> : 补充页眉空格。	100
改进 <code>hyperref</code> 宏包的标题锚点设置。	95	v2.4.12	(2017/12/05 – 2018/01/27)
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	45	General: 同步 <code>L^AT_EX 3 2017/12/16</code> 。	1
使用 <code>titlesec</code> 时, 章节目录也使用 <code>C_TE_X</code> 的编号。	97	修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥 东)。	79
提供 <code>\CTEXifname</code> 。	87	正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	68
提供 <code>\partmark</code> 。	87	<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	112
新的标题格式选项 <code>break</code> 。	85	v2.4.13	(2018/03/23)
v2.4.5	(2016/10/01 – 2016/10/25)	General: 修正导言区 <code>\selectfont</code> 钩子位置。	76
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	86	v2.4.14	(2018/05/01)
新的标题格式选项 <code>hang</code> 。	85	General: 配置 <code>macnew</code> 的默认字体设置。	121
新的标题格式选项 <code>tocline</code> 。	85	区分 <code>macold</code> 及 <code>macnew</code> 。	121
<code>\ps@headings</code> : 修复补丁失败。	100	为 <code>macnew</code> 配置字体命令。	124
v2.4.6	(2016/10/31 – 2016/11/20)	v2.4.15	(2019/01/29 – 2019/04/05)
General: <code>\CTEXifname</code> 初始为假。	87	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。	87
支持字体属性可选项在后的新语法。	60	将 <code>JY2</code> 和 <code>JT2</code> 编码的字体定义提取到单独的文件中。	114
重新初始化 <code>\ifthechapter</code> 等。	99	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
v2.4.7	(2016/12/23 – 2016/12/27)	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
General: 依赖 <code>pxeveryse</code> 宏包。	70	同步 <code>L^AT_EX 3 2019/03/05</code> 。	1
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错 误。	100	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。	122
v2.4.8	(2017/02/23)	显式补丁 <code>upL^AT_EX</code> 的 <code>\rmfamily</code> 等字体命令。	68
<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	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。	87

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	
\	509, 3270, 3277, 3293, 3296
\#	475, 3171, 3230
.../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
\/	1645
\\	15, 33, 34, 103, 146, 148, 150, 207, 217, 673, 1103, 1185, 2338, 3657, 3658, 3771, 3772, 3992, 4022, 5090, 5091, 5092, 5166
\{	1114, 1115, 5092
\}	1114, 1115, 5092
10pt	9, 368
11pt	9, 368
12pt	9, 368
A	
\abovedisplayshortskip	3862, 3870, 3881, 3901, 3909, 3920
\abovedisplayskip	3861, 3864, 3869, 3876, 3880, 3887, 3900, 3903, 3908, 3915, 3919, 3926
\abstractname	2322
abstractname	12
\addCJKfontfeature	1185, 1229
\addCJKfontfeatures	1202
\addcontentsline	2539, 2585, 2589, 2594, 2623, 2627, 2632, 2723, 2728, 2734, 2740, 2863, 2877
\AddEnumerateCounter	2277, 2279, 2280, 2281
\addpenalty	2822, 3497, 3530, 3564, 3578, 3592
\addtocontents	2527, 2533, 2744, 2745
\addvspace	2528, 2534, 2546, 2549, 2744, 2745, 2822, 2826, 3310, 3320, 3330, 3340, 3350, 3360
adobefonts	30
\algorithmname	2329, 2345
algorithmname	12
AlternateFont	27, 1341
\appendix	3019
appendix/name	24, 3008
appendix/number	24, 3008
appendix/numbering	23, 3008
\appendixname	2324, 3642
appendixname	12
\arabic	3544
\AtBeginDocument	576, 1634
\AtBeginDvi	434
\AtBeginShipoutFirst	436, 1853, 1868
\AtBeginUTFCommand	720
\AtEndOfClass	136
\AtEndOfPackage	137
\AtEndUTFCommand	726
autoindent	10, 186, 2132
B	
\baselinestretch	2187
\begin	3316, 3336, 3356
\begingroup	2854, 2890, 2922, 3304, 3324, 3344, 4591, 4624, 4631
\belowdisplayshortskip	3863, 3871, 3882, 3902, 3910, 3921
\belowdisplayskip	3864, 3876, 3887, 3903, 3915, 3926
\bfdefault	967, 973, 1331, 1338
\bfseries	2601, 2605, 2641, 2645, 2658, 2676, 2772, 2777, 2795, 3424, 3426, 3430, 3443, 3444, 3447, 3473, 3475, 3481, 3499, 3503, 3524, 3558, 3575, 3589
\bibname	2327, 2330, 2338, 2346, 2352, 2353
bibname	12
bool commands:	
\bool_gset_false:N	267, 272
\bool_gset_true:N	262
\bool_if:NTF	70, 1143, 2936, 2951, 3112, 3122, 3133, 3661
\bool_lazy_or:nnTF	5194
\bool_new:N	132, 1180
\bool_set_false:N	1155
\bool_set_true:N	1173
\c_false_bool	2967, 2969
\c_true_bool	2966, 2968
box commands:	
\box_ht:N	3942
\box_new:N	60
\box_wd:N	1939, 3226
C	
c5size	30
cap	30
captiondelimiter	31, 3949

- \catcode 4591, 4592, 4593, 4594, 4598, 4599
- CCT 30
- CCTfont 30
- \ccwd 21, 26, 116, 190, 1935, 1982, 1983, 1984, 2138, 2176, 2195, 2196, 2203, 2230, 2231, 2232, 2235, 2237, 2238, 2249, 2260, 2261, 3959, 3960, 3961
- \cdp 2050
- \centering 2638, 2673, 3306, 3326, 3346, 3430, 3442, 3447, 3451, 3481, 3503, 3511, 3546
- \chapter 2700, 3667
- chapter/beforeskip 32
- chapter/fixbeforeskip 32
- chapter/hang 22
- chapter/lofskip 21, 2468
- chapter/lotskip 21, 2468
- chapter/numbering 15
- chapter/pagestyle 22, 2468
- \chaptermark 2743
- \chaptername 3470
- char commands:
 - \char_set_catcode:nn 73, 513, 514, 549, 550
 - \char_set_catcode_active:N 1366
 - \char_set_catcode_comment:n 508
 - \char_set_catcode_escape:n 509
 - \char_set_catcode_letter:n 67, 1637, 3244, 5158
 - \char_set_catcode_math_superscript:n 544
 - \char_set_catcode_other:N 475, 875, 1349, 3171, 3230
 - \char_set_catcode_other:n 543, 5182
 - \char_set_catcode_space:n 2056
 - \char_value_catcode:n 73, 513, 514, 549, 550
- CharRange 27, 1341
- \Chinese 31, 2273
- \chinese 26, 2271, 2281, 3417, 3480
- \cht 2050
- \CJK 1904
- \CJKbold 4070, 4071, 4081, 4082, 4098, 4099, 4101, 4102, 4104, 4105, 4109, 4110, 4112, 4113, 4115, 4116, 4130, 4131, 4133, 4134, 4136, 4137, 4141, 4142, 4144, 4145, 4147, 4148
- \CJKfamily 523, 527, 531, 535, 1039, 1062, 1277, 1284, 1285, 1286, 1287, 1435, 1679, 4575, 4576, 4577, 4578, 4580, 4581, 4583, 4585, 4586
- \CJKfamilydefault 517, 518, 535, 591, 1264, 1265, 1277, 1287, 1295, 1297, 1301, 1308, 1321, 1656, 1657, 1743, 1747, 1755, 1768, 1770
- \CJKfilltwosides 1912
- \CJKfontspec 1202
- \CJKglue 1938, 1952, 1966, 1979
- \CJKhook 606
- \CJKnormal 4063, 4064, 4065, 4066, 4067, 4068, 4069, 4074, 4075, 4076, 4077, 4078, 4079, 4080, 4097, 4100, 4103, 4108, 4111, 4114, 4129, 4132, 4135, 4140, 4143, 4146
- \CJKpunctmapfamily 639, 640, 641, 642, 643, 644, 648, 649, 650, 651, 652, 653
- \CJKrmdefault 498, 518, 519, 523, 1110, 1121, 1233, 1265, 1266, 1284, 1297, 1300, 1301, 1653, 1656, 1662, 1751, 4199, 4201, 4202, 4207, 4208, 4209, 4212, 4289, 4292, 4293, 4298, 4299, 4300, 4367, 4370, 4371, 4377, 4378, 4379, 4477, 4480, 4481, 4488, 4489, 4490, 4537, 4540, 4541, 4546, 4547, 4548
- \CJKsfdefault 499, 520, 527, 1111, 1122, 1240, 1267, 1285, 1654, 1669, 1752, 4182, 4189, 4190, 4213, 4290, 4368, 4372, 4478, 4482, 4538
- \CJKtilde 592
- \CJKttdefault 500, 521, 531, 1112, 1123, 1246, 1268, 1286, 1655, 1676, 1753, 4200, 4214, 4291, 4369, 4479, 4539
- clearalternatefont 28, 1527
- \cleardoublepage 2560, 2701, 3440, 3468
- \clearpage 2562, 2701, 3440, 3468
- clist commands:
 - \clist_clear:N 988, 4927
 - \clist_concat:NNN 1008
 - \clist_const:Nn 619
 - \clist_gput_right:Nn 384, 396, 406, 408
 - \clist_gset:Nn 1256
 - \clist_if_empty:NTF 1354, 1373, 1613
 - \clist_map_break:n 3091, 3652
 - \clist_map_function:nN 1530, 1532, 1564, 1578
 - \clist_map_inline:Nn 609, 1466
 - \clist_map_inline:nn 369, 631, 1571, 1896, 1922, 3086, 3649, 3789, 3824, 4916
 - \clist_new:N 139, 1005, 1257, 1408, 4945, 5085
 - \clist_put_left:Nn 1009, 3718, 3744
 - \clist_put_right:Nn 1172, 1454, 4939
 - \clist_set:Nn 992, 1391, 1392, 5086
 - \clist_use:Nn 5095
- \clubpenalty 2921, 2928
- \contentsname 2317
- contentsname 12
- continuation 12
- cs commands:
 - \cs:w 3385
 - \cs_end: 3385
 - \cs_generate_variant:Nn 93, 451, 463, 952, 1074, 1161, 1181, 1182, 1387, 1424, 1526, 1564, 1572, 5127, 5133, 5249, 5356
 - \cs_gset_eq:NN 263, 268, 273, 483, 770, 774, 1148, 1151, 3055, 3056, 3061
 - \cs_gset_nopar:Npx 771
 - \cs_gset_protected_nopar:Npn 3020, 3067
 - \cs_gset_protected_nopar:Npx 809, 1035, 1428, 1522, 2996
 - \cs_if_eq:NNTF 911, 3053, 5136, 5282, 5362
 - \cs_if_exist:NTF 44, 360, 460, 765, 777, 807, 845, 930, 1427, 1519, 2014, 3157, 3372, 3667, 3669, 5141
 - \cs_if_exist_p:N 5195, 5196
 - \cs_if_exist_use:N 1058, 1059
 - \cs_if_exist_use:NTF 4, 748, 1080, 1541, 5130
 - \cs_if_free:NTF 445, 455, 611, 626, 1145, 1330, 2151, 2165, 2277, 3105, 3150, 3196, 3201, 3205
 - \cs_new:Npn 111, 118, 125, 866, 872, 876, 984, 1061, 2970, 2978, 3005, 3130, 5239, 5357

- \cs_new_eq:NN ... 6, 7, 8, 9, 20, 23, 107, 108, 109, 134, 135, 361, 452, 596, 657, 786, 811, 887, 918, 1229, 1236, 1764, 1917, 1918, 1919, 1931, 1979, 1990, 1996, 2272, 2273, 2274, 2513, 2516, 2964, 2965, 2966, 2967, 2968, 2969, 3019, 3043, 3044, 3390, 3665, 3670, 4921, 5286, 5358
- \cs_new_nopar:Npn 826, 1053, 1106, 1117, 2271, 2450, 2986, 3194, 3199, 4946
- \cs_new_protected:Npn 62, 920, 985, 1188, 1352, 1443, 1457, 1608, 1985, 2461, 2538, 2934, 2941, 3220, 3844, 3850, 4922, 5100, 5102, 5116, 5122, 5142, 5147, 5152, 5163, 5167, 5172, 5177, 5187, 5192, 5211, 5262, 5277, 5287, 5299, 5308, 5327, 5332, 5344, 5359
- \cs_new_protected_nopar:Npn 77, 79, 136, 137, 432, 443, 453, 458, 464, 496, 504, 539, 556, 598, 604, 607, 624, 629, 637, 646, 655, 746, 755, 768, 773, 775, 787, 801, 828, 843, 856, 880, 889, 906, 945, 962, 1006, 1014, 1033, 1054, 1064, 1089, 1127, 1139, 1141, 1163, 1165, 1274, 1280, 1291, 1316, 1325, 1371, 1382, 1409, 1418, 1425, 1431, 1451, 1464, 1483, 1497, 1504, 1516, 1536, 1552, 1570, 1573, 1575, 1588, 1600, 1642, 1647, 1649, 1703, 1716, 1730, 1741, 1773, 1791, 1806, 1822, 1833, 1837, 1874, 1881, 1885, 1908, 1914, 1935, 1949, 1981, 1997, 2029, 2074, 2150, 2152, 2177, 2212, 2222, 2227, 2380, 2410, 2497, 2504, 2511, 2517, 2519, 2522, 2949, 2955, 3036, 3046, 3078, 3098, 3103, 3108, 3118, 3128, 3172, 3368, 3377, 3382, 3762, 3777, 3809, 3814, 3964, 3968, 4913, 4937, 5104, 5106, 5128
- \cs_new_protected_nopar:Npx ... 468, 1212, 1901, 3159
- \cs_replacement_spec:N 5203, 5225
- \cs_set:Npn 1350, 1993, 2437
- \cs_set_eq:NN 110, 191, 196, 201, 536, 554, 574, 601, 605, 613, 627, 780, 781, 888, 919, 937, 982, 1039, 1040, 1427, 1435, 1445, 1446, 1447, 1519, 1520, 1521, 1645, 1746, 2288, 2293, 2296, 2301, 2518, 2520, 2957, 2958, 2959, 2960, 2961, 2962, 3050, 3082, 3092
- \cs_set_nopar:Npn 1702, 3211
- \cs_set_nopar:Npx 1932, 3120, 5108
- \cs_set_protected:Npn 727, 923, 5190, 5254
- \cs_set_protected_nopar:Npn 600, 938, 1952, 3152, 3209, 3851
- \cs_set_protected_nopar:Npx 1987
- \cs_to_str:N 54, 1211, 5188
- \cs_undefine:N . 1018, 1019, 1023, 1024, 1448, 2001, 5131
- cs4size 30
- \csname 2823, 2832, 2833, 2834, 2841, 2859, 2860, 2862, 2865, 2874, 2875, 2876, 2879, 4623, 4625, 4626, 4627, 4632, 4636
- \CTeX 26, 3945
- ctex commands:
 - \ctex_add_cmap:n 448, 453
 - \ctex_add_cmap:Nn 456, 458, 463
 - \ctex_add_to_selectfont:n 1985, 2020, 2048, 2076, 2164
 - \ctex_after_end_preamble:n 37, 110, 5100
 - \ctex_appto_cmd:NnnTF 479, 1636, 5172
 - \ctex_assign_heading_name:nn 2416, 2458, 3012
 - \ctex_at_end:n 136, 295, 358, 1979, 4031
 - \ctex_at_end_package:nn 133, 435, 482, 735, 928, 1847, 1849, 1867, 2275, 3065, 3138, 3145, 3204, 3206, 3207, 3241, 3281, 3396, 3947, 5116
 - \ctex_at_end_preamble:n 363, 581, 588, 1290, 1690, 5100
 - \ctex_auto_ignorespaces: 595, 605, 2086
 - \ctex_CJK_input:n 539
 - \ctex_declare_math_sizes:nnnn 3809, 3843
 - \ctex_default_pt:n 114, 118
 - \ctex_detected_platform: 1772, 3972, 3981
 - \ctex_family_cmap:nn 443, 452
 - \ctex_file_input:n 77, 399, 658, 2115, 3676, 3711, 3716, 3721, 3722, 3737, 3742, 3748, 3749, 3853, 3855, 3973, 3977, 3985, 4031, 4160, 4161, 4417, 4418
 - \ctex_file_wrapper:nnn 62, 78, 81, 506, 541
 - \ctex_fix_varioref_label:n 3380, 3382
 - \ctex_fontset_error:n 3964, 4268, 4302, 4336, 4381, 4422, 4425, 4492, 4550
 - \ctex_gadd_hook:Nn 606, 5120, 5122
 - \ctex_get_macro_meaning:N 5257
 - \ctex_get_macro_meaning:NFT 5253, 5279, 5335
 - \ctex_hypersetup:n 72, 361, 1830, 1842, 1845, 1860, 1863, 1866
 - \ctex_if_autoindent_touched:TF 134, 191, 196, 201, 3703
 - \ctex_if_ccglue_touched:TF 1963, 2183
 - \ctex_if_ccglue_touched_p: 1963
 - \ctex_if_macosx:TF 1813, 1814, 1816, 1822
 - \ctex_if_preamble:TF 109, 4000
 - \ctex_if_rescanable:NnTF 5277, 5311
 - \ctex_ignorespaces_case:N 598, 2083, 2086, 2089
 - \ctex_load_fontset: 3968, 4008, 4030
 - \ctex_load_zhmap:nnnn 496, 4205, 4296, 4375, 4486, 4544
 - \ctex_ljt_add_font_features:n 1163, 1226
 - \ctex_ljt_add_font_features:nn 1163, 1164
 - \ctex_ljt_char_range_key:nn 1605, 1608
 - \ctex_ljt_clear_alternate_font:n 1530, 1536
 - \ctex_ljt_declare_alternate_shape:nnnnnn 1437, 1457
 - \ctex_ljt_declare_char_range:n 1567, 1570, 1572
 - \ctex_ljt_declare_char_range:nn 1574, 1575
 - \ctex_ljt_def_char_range_key:n 1580, 1600
 - \ctex_ljt_ensure_default_family: 1291, 1760
 - \ctex_ljt_extract_font: 780, 787
 - \ctex_ljt_family_if_exist:nNTF 1066, 1075, 1295, 1300, 1318, 1321, 1539
 - \l_ctex_ljt_family_tl 1068, 1073, 1130, 1131, 1146, 1150, 1153, 1164, 1533, 1534
 - \ctex_ljt_fontspec:nn 1127, 1139, 1174, 1221
 - \ctex_ljt_get_and_define_fonts:nN 880
 - \ctex_ljt_get_and_define_fonts_al:nN 884, 887
 - \ctex_ljt_get_and_define_fonts_ja:nN 883, 889
 - \ctex_ljt_if_alternate_shape_exist:nTF 790, 803, 821, 830
 - \ctex_ljt_if_jfont:nTF 55, 866, 877, 882
 - \ctex_ljt_if_jfont_math:NFT 872, 913
 - \ctex_ljt_math_group_hook: 55, 55, 912, 918, 938
 - \ctex_ljt_pickup_font: ... 761, 775, 850, 892, 894, 896
 - \ctex_ljt_reset_alternate_font:n 1532, 1552

- `\ctex_ltj_save_alternate_family:Nnnn` [1402](#), [1422](#), [1425](#)
`\ctex_ltj_save_alternate_seq:Nn` [1377](#), [1382](#)
`\ctex_ltj_save_alternate_seq:Nnnwn` [1382](#)
`\ctex_ltj_save_alternate_seq:Nnnwnw` [1385](#), [1388](#)
`\ctex_ltj_save_char_range:n` [1578](#), [1585](#)
`\ctex_ltj_save_char_range:nn` [1587](#), [1588](#)
`\ctex_ltj_select_alternate_font:` [828](#), [2023](#)
`\ctex_ltj_select_font:` [746](#), [2022](#)
`\ctex_ltj_set_alternate_family:nn` [1380](#), [1409](#)
`\ctex_ltj_set_alternate_family:nnn` . [1421](#), [1429](#), [1431](#)
`\ctex_ltj_set_alternate_family:Nnnnn` [1413](#), [1418](#), [1424](#)
`\ctex_ltj_set_alternate_family:nnTF` [993](#), [1371](#)
`\ctex_ltj_set_alternate_seq:n` [1343](#), [1348](#)
`\ctex_ltj_set_alternate_shape:n` ... [1485](#), [1497](#), [1508](#)
`\ctex_ltj_set_alternate_shape:nnN` [1470](#), [1504](#)
`\ctex_ltj_set_alternate_shape:nnn` [1473](#), [1480](#)
`\ctex_ltj_set_alternate_shape:nnnn` [1482](#), [1483](#)
`\ctex_ltj_set_alternate_shape:Nnnnnnn` . [1460](#), [1464](#)
`\ctex_ltj_set_family:nnn`
 [985](#), [1158](#), [1205](#), [1215](#), [1233](#), [1240](#), [1246](#), [1252](#)
`\ctex_ltj_set_kanjiskip:N` [1647](#), [1957](#)
`\ctex_ltj_set_math_letter:NN` [920](#), [932](#), [940](#)
`\ctex_ltj_set_xkanjiskip:N` [1647](#), [2036](#)
`\ctex_ltj_subst_font:` [781](#), [801](#)
`\ctex_ltj_swap_cs:NN` [1436](#), [1443](#)
`\ctex_ltj_switch_family:n`
 [1063](#), [1064](#), [1074](#), [1131](#), [1159](#), [1213](#)
`\ctex_ltj_update_mathfont:` [1313](#), [1316](#)
`\ctex_ltj_update_mathfont:n` [1319](#), [1322](#), [1325](#)
`\ctex_ltj_use_math_group:Nn` [906](#), [937](#)
`\ctex_make_spa:nn` [4913](#)
`\ctex_mono_jfm:n` [945](#), [953](#), [2106](#)
`\ctex_package_end_hook:n` [5128](#), [5137](#)
`\ctex_parse_name:NN` [2004](#), [5183](#), [5187](#)
`\ctex_parse_name:NNn` [5188](#), [5192](#), [5249](#)
`\ctex_patch_boot:NnnnTF`
 [5144](#), [5149](#), [5154](#), [5169](#), [5174](#), [5177](#)
`\ctex_patch_cmd:Nnn`
 [1698](#), [3174](#), [3180](#), [3186](#), [3253](#), [3256](#), [3260](#),
 [3266](#), [3276](#), [3284](#), [3287](#), [3292](#), [3295](#), [3959](#), [3960](#), [3961](#), [5152](#)
`\ctex_patch_cmd_all:NnnnTF` [5147](#)
`\ctex_patch_cmd_once:NnnnTF` [1692](#), [2006](#), [5142](#)
`\ctex_patch_failure:N` [485](#),
 [525](#), [529](#), [533](#), [537](#), [1640](#), [2011](#), [3237](#), [3247](#), [5161](#), [5163](#)
`\ctex_plane_to_utfxvibe:Nn` [555](#)
`\ctex_preto_cmd:NnnTF` [522](#), [526](#), [530](#), [534](#), [3234](#), [3243](#), [5167](#)
`\ctex_punct_map_bfseries:nn`
 [629](#), [4190](#), [4202](#), [4208](#), [4293](#),
 [4299](#), [4371](#), [4372](#), [4378](#), [4481](#), [4482](#), [4483](#), [4489](#), [4541](#), [4547](#)
`\ctex_punct_map_family:nn` [624](#), [4182](#), [4189](#),
 [4199](#), [4200](#), [4207](#), [4289](#), [4290](#), [4291](#), [4298](#), [4367](#), [4368](#),
 [4369](#), [4377](#), [4477](#), [4478](#), [4479](#), [4488](#), [4537](#), [4538](#), [4539](#), [4546](#)
`\ctex_punct_map_itshape:nn` [646](#),
 [4201](#), [4209](#), [4292](#), [4300](#), [4370](#), [4379](#), [4480](#), [4490](#), [4540](#), [4548](#)
`\ctex_punct_map_series:nnn` [633](#), [634](#), [637](#)
`\ctex_punct_set:n` [607](#),
 [4177](#), [4206](#), [4288](#), [4297](#), [4366](#), [4376](#), [4476](#), [4487](#), [4536](#), [4545](#)
`\ctex_punct_space:nn` [655](#)
`\ctex_scheme_input:n` [79](#), [3398](#), [3678](#)
`\ctex_select_size:` .. [2139](#), [2146](#), [2149](#), [2150](#), [2171](#), [2225](#)
`\ctex_set_default_ccwd:Nn` [111](#), [200](#), [2145](#), [2170](#)
`\ctex_set_font_size:Nnn` [3844](#), [3859](#),
 [3867](#), [3878](#), [3889](#), [3890](#), [3891](#), [3892](#), [3893](#), [3894](#), [3895](#),
 [3898](#), [3906](#), [3917](#), [3928](#), [3929](#), [3930](#), [3931](#), [3932](#), [3933](#), [3934](#)
`\ctex_set_ignorespaces:` [598](#), [606](#)
`\ctex_set_upfamily:nnn`
 [1703](#), [4225](#), [4226](#), [4227](#), [4228](#), [4229](#),
 [4230](#), [4232](#), [4312](#), [4313](#), [4314](#), [4315](#), [4391](#), [4392](#), [4393](#),
 [4394](#), [4501](#), [4502](#), [4503](#), [4504](#), [4505](#), [4507](#), [4559](#), [4560](#), [4561](#)
`\ctex_set_upfonts:nnnnnn`
 [1730](#), [4220](#), [4308](#), [4387](#), [4497](#), [4555](#)
`\ctex_set_upmap:nnn`
 ... [1716](#), [1732](#), [1733](#), [1734](#), [1735](#), [4231](#), [4233](#), [4506](#), [4508](#)
`\ctex_set_zhmap:n` [431](#), [501](#), [1718](#)
`\ctex_titleps_hook:` [3170](#), [3205](#), [3206](#)
`\ctex_titlesec_hook:` [3078](#), [3142](#)
`\ctex_update_ccglue:` [1949](#), [2190](#), [2209](#), [2253](#)
`\ctex_update_ccwd:` [1935](#), [2184](#), [2261](#)
`\ctex_update_default_family:` [47](#), [581](#), [1290](#), [1690](#), [1740](#)
`\ctex_update_em_unit:` [1981](#), [2179](#), [2229](#)
`\ctex_update_kanjisize:` [2074](#), [2076](#)
`\ctex_update_parindent:` [2157](#), [2212](#), [2254](#)
`\ctex_update_size:` [2151](#), [2152](#)
`\ctex_update_stretch:` [2156](#), [2177](#)
`\ctex_update_xkanjiskip:` [2028](#), [2048](#)
`\ctex_update_ziju:` [2159](#), [2227](#)
`\ctex_varioref_hook:` [3377](#), [3388](#), [3396](#)
`\ctex_zhmap_case:nnn`
 [135](#), [263](#), [268](#), [273](#), [4175](#), [4270](#), [4338](#), [4464](#), [4527](#)
`\ctex_zhmap_input:n` [501](#), [504](#)
`\ctex_zihao:n` [3761](#), [3762](#)
`\ctex_ziju:n` [2221](#), [2222](#)
ctex internal commands:
`\g_ctex_after_end_preamble_hook_tl` [5103](#), [5107](#), [5115](#)
`\_ctex_assign_heading_name:nnn` [2458](#)
`\l_ctex_autoindent_tl`
 [133](#), [190](#), [195](#), [200](#), [2138](#), [2142](#), [2145](#), [2214](#), [2217](#)
`\_ctex_calc_bounds:nn` [4941](#), [4942](#), [4946](#)
`\_ctex_ccglue:` [1966](#), [1979](#)
`\l_ctex_ccglue_skip` [1953](#),
 [1957](#), [1960](#), [1962](#), [1970](#), [1974](#), [2188](#), [2199](#), [2208](#), [2244](#), [2252](#)
`\c_ctex_class_tl`
 .. [84](#), [86](#), [411](#), [416](#), [421](#), [426](#), [3665](#), [3671](#), [3673](#), [3674](#), [3676](#)
`\_ctex_clear_fntef_color:n` ... [1874](#), [1881](#), [1885](#), [1900](#)
`\c_ctex_cmap_encoding_seq` [441](#), [447](#)
`\l_ctex_ctexcap_options_clist` [5085](#), [5086](#), [5095](#)
`\_ctex_def_heading_keys:n` [2410](#), [2488](#)
`\_ctex_default_pt:w` [120](#), [125](#)
`\l_ctex_encoding_tl` [129](#), [223](#),
 [224](#), [489](#), [492](#), [583](#), [591](#), [1843](#), [2112](#), [2269](#), [3706](#), [3733](#), [3747](#)
`\g_ctex_end_preamble_hook_tl` [5101](#), [5105](#), [5114](#)

`\c_ctex_engine_file_str` [43](#), [2115](#)
`\c_ctex_engine_str` [43](#)
`\l_ctex_family_default_init_tl` [1743](#), [1763](#)
`\_ctex_family_default_wrap:n` [1746](#), [1764](#), [1767](#)
`\_ctex_fix_varioref_label:n` [3382](#)
`\g_ctex_font_size_int` [152](#),
[163](#), [164](#), [165](#), [379](#), [383](#), [405](#), [3723](#), [3724](#), [3750](#), [3751](#), [3852](#)
`\c_ctex_font_size_prop` [3764](#), [3775](#), [3819](#), [3846](#)
`\c_ctex_font_size_seq` [3773](#), [3776](#), [3787](#)
`\l_ctex_font_size_tl` [3764](#), [3765](#), [3811](#), [3812](#), [3846](#), [3847](#)
`\g_ctex_fontset_tl` [227](#), [1775](#), [1793](#), [1818](#), [1825](#), [1826](#),
[3970](#), [3973](#), [3976](#), [3977](#), [3983](#), [3985](#), [4005](#), [4007](#), [4012](#), [4029](#)
`\_ctex_get_font_sizes:Nn` [3811](#), [3814](#)
`\_ctex_get_macro_meaning:w` [5253](#)
`\l_ctex_heading_bool` [306](#), [3661](#)
`\l_ctex_heading_skip` [2501](#), [2508](#), [2510](#), [2512](#),
[2513](#), [2524](#), [2525](#), [2528](#), [2530](#), [2531](#), [2534](#), [2944](#), [2945](#), [2946](#)
`\c_ctex_headings_cs_seq` [3038](#), [3051](#), [3060](#)
`\c_ctex_headings_seq` ... [2370](#), [2485](#), [3155](#), [3218](#), [3379](#)
`\_ctex_hookto_cmd:Nnnnw` [5169](#), [5174](#), [5332](#)
`\_ctex_hookto_cmd_parameter:Nnnnw` [5339](#), [5359](#)
`\_ctex_hookto_cmd_parameterless:Nnnnw` . [5338](#), [5344](#)
`\_ctex_indent_aux:n` [2939](#), [2941](#)
`\_ctex_initial_heading:n` [2380](#), [2487](#)
`\g_ctex_kanjisize_func` [2054](#), [2064](#), [2075](#)
`\_ctex_left_hook_aux:nn` [5357](#)
`\l_ctex_line_spread_fp`
..... [112](#), [183](#), [3725](#), [3726](#), [3752](#), [3753](#), [3938](#), [3940](#)
`\l_ctex_line_stretch_tl` [2170](#), [2175](#), [2176](#), [2180](#)
`\l_ctex_ltj_add_alternate_bool` [1143](#), [1155](#), [1173](#), [1180](#)
`\_ctex_ltj_alternate_cs:n`
..... [1019](#), [1022](#), [1023](#), [1024](#), [1058](#), [1059](#), [1061](#), [1146](#),
[1149](#), [1150](#), [1152](#), [1153](#), [1377](#), [1414](#), [1476](#), [1501](#), [1541](#), [1558](#)
`\l_ctex_ltj_alternate_family_tl` [1439](#), [1442](#)
`\l_ctex_ltj_alternate_options_clist`
..... [1392](#), [1393](#), [1404](#), [1408](#)
`\l_ctex_ltj_alternate_seq`
..... [989](#), [1361](#), [1362](#), [1370](#), [1376](#), [1384](#), [1632](#)
`\l_ctex_ltj_base_CJKfamily_tl`
..... [990](#), [1004](#), [1056](#), [1476](#), [1501](#)
`\l_ctex_ltj_base_family_tl` [65](#),
[1020](#), [1051](#), [1057](#), [1420](#), [1461](#), [1520](#), [1539](#), [1544](#), [1545](#), [1557](#)
`\_ctex_ltj_change_encoding:` [954](#), [1038](#), [1434](#)
`\l_ctex_ltj_char_range_clist` [988](#),
[1345](#), [1354](#), [1373](#), [1391](#), [1403](#), [1415](#), [1438](#), [1460](#), [1611](#), [1613](#)
`\_ctex_ltj_char_range_parse_feature:w` . [1616](#), [1620](#)
`\g_ctex_ltj_char_range_prop` [1468](#), [1575](#)
`\l_ctex_ltj_char_range_tl`
..... [1468](#), [1471](#), [1577](#), [1579](#), [1581](#), [1583](#), [1590](#)
`\_ctex_ltj_check_family:n` [1014](#), [1375](#)
`\l_ctex_ltj_current_font_tl`
..... [51](#), [748](#), [752](#), [753](#), [765](#), [807](#), [812](#)
`\l_ctex_ltj_current_shape_tl` [830](#), [835](#), [840](#), [841](#)
`\_ctex_ltj_declare_char_range:nn` [1571](#), [1573](#)
`\g_ctex_ltj_default_features_clist` [1008](#), [1256](#), [1257](#)
`\_ctex_ltj_family_csname:n` ... [1018](#), [1035](#), [1053](#), [1080](#)
`\g_ctex_ltj_family_font_name_prop`
..... [995](#), [1011](#), [1016](#), [1091](#), [1167](#), [1293](#), [1305](#)
`\g_ctex_ltj_family_font_options_prop` [996](#), [1011](#), [1170](#)
`\g_ctex_ltj_family_int` . [1133](#), [1135](#), [1453](#), [1454](#), [1456](#)
`\g_ctex_ltj_family_name_prop`
..... [1011](#), [1020](#), [1043](#), [1077](#), [1420](#)
`\_ctex_ltj_family_unknown_warning:n` [1071](#), [1089](#), [1549](#)
`\l_ctex_ltj_font_options_clist` [992](#), [997](#),
[998](#), [999](#), [1005](#), [1042](#), [1171](#), [1172](#), [1175](#), [1411](#), [1412](#), [1416](#)
`\_ctex_ltj_fontspec:nnn` [1134](#), [1141](#), [1161](#)
`\g_ctex_ltj_fontspec_family_tl`
..... [1041](#), [1044](#), [1045](#), [1046](#), [1052](#), [1057](#), [1082](#)
`\g_ctex_ltj_fontspec_prop` [1129](#), [1157](#), [1162](#)
`\_ctex_ltj_gset_family_cs:nn` [1000](#), [1033](#)
`\_ctex_ltj_if_jfont_math:w` [873](#), [876](#)
`\l_ctex_ltj_jfm_tl` [945](#), [966](#), [968](#), [1009](#)
`\c_ctex_ltj_math_fam_int` [974](#), [975](#)
`\c_ctex_ltj_math_family_tl`
..... [1327](#), [1329](#), [1331](#), [1334](#), [1338](#)
`\c_ctex_ltj_math_tl`
..... [969](#), [970](#), [972](#), [974](#), [1252](#), [1318](#), [1328](#), [1333](#), [1337](#)
`\_ctex_ltj_msg_def_family_map:n` [1104](#), [1106](#)
`\_ctex_ltj_msg_family_map:n` [1032](#), [1103](#), [1117](#)
`\_ctex_ltj_noembed_wrap:n` [982](#), [984](#)
`\_ctex_ltj_pass_args:nnnn`
..... [1188](#), [1204](#), [1214](#), [1220](#), [1232](#), [1239](#), [1245](#), [1251](#)
`\_ctex_ltj_patch_external_font:w` [793](#), [826](#)
`\_ctex_ltj_pop_fontname:` [764](#), [773](#), [852](#)
`\_ctex_ltj_post_arg:w` [1191](#), [1197](#)
`\_ctex_ltj_push_fontname:n` [760](#), [768](#), [849](#)
`\g_ctex_ltj_reset_alternate_prop`
..... [1025](#), [1543](#), [1555](#), [1563](#)
`\_ctex_ltj_save_alternate_shape:Nnn` [1475](#), [1500](#), [1516](#)
`\_ctex_ltj_save_fontname:w` [770](#), [774](#)
`\_ctex_ltj_select_font_aux:` [749](#), [755](#)
`\_ctex_ltj_set_alternate_family:n` [1047](#), [1054](#)
`\_ctex_ltj_tmp:w` [1350](#), [1367](#), [1445](#), [1447](#), [1448](#)
`\l_ctex_ltj_tmp_seq` [1358](#), [1359](#), [1362](#), [1369](#)
`\l_ctex_ltj_tmp_tl`
..... [991](#), [992](#), [1016](#), [1027](#), [1030](#), [1168](#), [1176](#),
[1210](#), [1213](#), [1215](#), [1295](#), [1300](#), [1318](#), [1319](#), [1321](#), [1322](#),
[1356](#), [1357](#), [1358](#), [1395](#), [1397](#), [1398](#), [1405](#), [1615](#), [1623](#), [1632](#)
`\g_ctex_ltj_unknown_family_seq` ... [1093](#), [1095](#), [1100](#)
`\_ctex_ltj_update_family_aux:` [1278](#), [1280](#)
`\_ctex_ltj_update_family_uid:N` [998](#), [1411](#), [1450](#)
`\_ctex_ltj_use_global_options:N` [999](#), [1006](#), [1393](#), [1412](#)
`\c_ctex_macosx_file_str` [1824](#), [1828](#)
`\l_ctex_msyh_suffix_tl`
..... [4164](#), [4165](#), [4169](#), [4186](#), [4188](#), [4222](#)
`\l_ctex_parameter_str` [5250](#), [5269](#), [5290](#), [5337](#)
`\l_ctex_parameter_tl` [5290](#), [5295](#), [5306](#)
`\_ctex_parse_name:nNNNnN` [5202](#), [5211](#)
`\_ctex_parse_name:w` [5224](#), [5230](#), [5239](#)
`\_ctex_patch_cmd:Nnnnw` [5144](#), [5149](#), [5154](#), [5308](#)
`\_ctex_patch_false:w` [5180](#), [5186](#), [5323](#), [5325](#), [5342](#), [5368](#)
`\_ctex_patch_rebuild:Nn` [5281](#), [5287](#), [5320](#), [5361](#), [5365](#)

- _ctex_patch_replace:nnnTF 5315, 5327
 - _ctex_patch_rescan:NNn 5289, 5290, 5291, 5299
 - _ctex_patch_true:w 5179, 5185, 5321, 5354, 5366
 - _ctex_plus_key_aux:nn 2440, 2442, 2443, 2444, 2445, 2446, 2447, 2450
 - \l_ctex_prefix_str 5250, 5268, 5289, 5346, 5347
 - \l_ctex_prefix_tl 5289, 5294, 5305
 - \l_ctex_punct_bounds_clist ... 4927, 4933, 4939, 4945
 - \c_ctex_punct_family_clist 609, 619
 - \l_ctex_punct_font 4925, 4926
 - \c_ctex_punct_seq 4905, 4928
 - \l_ctex_punct_tl 282, 578, 579, 665, 953, 2103, 2104, 2105, 2106
 - _ctex_rebuild_cmd:w ... 5281, 5282, 5286, 5361, 5362
 - \l_ctex_replacement_str 5250, 5270, 5291, 5329, 5330, 5364
 - \l_ctex_replacement_tl 5291, 5296, 5307
 - _ctex_restore_selectfont: 1997, 2001, 2010
 - _ctex_right_hook_aux:nn 5358
 - _ctex_save_bounds:n 4929, 4937
 - _ctex_save_cmap:Nn 460, 464
 - _ctex_save_font_size:nn 3775
 - \l_ctex_scheme_tl 317, 3398, 3678
 - \g_ctex_section_depth_int 94, 311, 313, 3600, 3616, 3632
 - \c_ctex_section_headings_seq . 2366, 2372, 2993, 3096
 - _ctex_set_font_size:nnNn 3847, 3850
 - \g_ctex_spa_iow 4915, 4918, 4920, 4930
 - \c_ctex_std_class_tl ... 3649, 3664, 3665, 3731, 3754
 - \g_ctex_std_options_clist 138, 384, 396, 406, 408, 412, 417, 422, 427
 - _ctex_titlesec_format:n 3092, 3100, 3103
 - _ctex_titlesec_format:Nn 3082, 3106, 3108
 - _ctex_titlesec_format_explicit:Nn ... 3083, 3118
 - _ctex_titlesec_hook:n 3092, 3096, 3098
 - _ctex_titlesec_spacing:Nn 3101, 3128
 - _ctex_titlesec_spacing:nnnnn 3129, 3130
 - _ctex_tmp:N 1993, 2004
 - _ctex_tmp:w 5190, 5243, 5254, 5275
 - \l_ctex_tmp_box 58, 1938, 1939, 3222, 3226
 - \l_ctex_tmp_dim 58, 2180, 2181, 2186, 2189, 2195, 2234, 2236, 2238, 2247, 2249
 - \l_ctex_tmp_int 58, 559, 561, 562, 564, 568, 570, 2194, 2196, 2197, 2203, 2204, 2236, 2237, 2238, 2241, 2242, 2247, 2249, 2992, 2995, 2999
 - \l_ctex_tmp_tl 58, 368, 375, 390, 391, 466, 467, 558, 560, 1995, 2006, 2412, 2470, 2483, 2493, 3730, 3731, 3819, 3820
 - \c_ctex_upper_case_file_str 1812, 1820
 - \c_ctex_version_tl 3, 6, 7, 8, 9, 20, 23
 - _ctex_write_family:nn 4917, 4922
 - \l_ctex_xkanjiskip_skip 2028
 - \l_ctex_xkanjiskip_tl 2035, 2040, 2041
 - \c_ctex_zero_tl 2154, 2161, 2163
 - \g_ctex_zhmCJK_bool 132, 262, 267, 272, 488, 582
 - \l_ctex_ziju_dim 2230, 2231, 2235, 2246, 2248, 2252, 2256
 - \l_ctex_ziju_tl 2154, 2162, 2163, 2224, 2230
 - \ctexbibname 2339, 2359
 - \CTEXcounter 2274
 - \CTEXdigits 26, 2284
 - \CTEXfilltwosides 31, 1908, 1918
 - \CTEXifname 16, 2516, 2604, 2644, 2776, 2972, 2980, 2988, 3213, 3255, 3258, 3262, 3274, 3278, 3286, 3289, 3294, 3297
 - \CTEXindent 31, 2257
 - \CTEXnoindent 31, 2257
 - \CTEXnumber 26, 2284
 - \CTEXnumberline 2986, 3006, 3477
 - \CTEXoptions 31, 2119
 - \ctexset 5, 2116, 2123, 2129, 3305, 3325, 3345
 - \CTEXsetfont 31, 2149
 - \CTEXsetup 31, 2119
 - \CTEXsout 31
 - \ctexspadef 655, 4932
 - \CTEXthechapter 16, 2722, 2981, 3148, 3274, 3294
 - \CTEXtheparagraph 16
 - \CTEXthepart 16, 2973, 3153
 - \CTEXthesection 16, 3255, 3262, 3278, 3286, 3297
 - \CTEXthesubparagraph 16
 - \CTEXthesubsection 16, 3258, 3289
 - \CTEXthesubsubsection 16
 - \CTEXunderdblline 31
 - \CTEXunderdot 31
 - \CTEXunderline 31
 - \CTEXunderwave 31
 - \CTEXxout 31
 - \CurrentOption 396
 - \cwd 2050
- D**
- declarecharrange 27, 1565
 - \DeclareErrorKanjiFont 1684
 - \DeclareFontEncoding 955
 - \DeclareFontFamily 474, 964, 4049, 4050, 4051, 4052, 4053, 4054
 - \DeclareFontShape 965, 967, 1707, 1708, 1711, 1712, 4063, 4064, 4065, 4066, 4067, 4068, 4069, 4070, 4071, 4074, 4075, 4076, 4077, 4078, 4079, 4080, 4081, 4082, 4085, 4086, 4087, 4090, 4091, 4092, 4097, 4098, 4099, 4100, 4101, 4102, 4103, 4104, 4105, 4108, 4109, 4110, 4111, 4112, 4113, 4114, 4115, 4116, 4119, 4120, 4123, 4124, 4129, 4130, 4131, 4132, 4133, 4134, 4135, 4136, 4137, 4140, 4141, 4142, 4143, 4144, 4145, 4146, 4147, 4148, 4151, 4154
 - \DeclareFontSubstitution 958
 - \DeclareKanjiFamily 1705, 1706, 4055, 4056, 4057, 4058, 4059, 4060
 - \DeclareKanjiSubstitution 1685, 1686
 - \DeclareMathSizes 3812
 - \DeclareSymbolFont 970, 1328, 1687
 - \def 2580, 2618, 2651, 2669, 2682, 2713, 2753, 2762, 2786, 2805, 2831, 2835, 2870, 2885, 2899, 2900, 2904, 3872, 3883, 3911, 3922, 4597, 4633

`\defaultCJKfontfeatures` [1230](#)
`\defbeamertemplate` [3303](#), [3323](#), [3343](#)
`\defbeamertemplatealias` [3363](#), [3364](#), [3365](#)
dim commands:
`\dim_abs:n` [2248](#)
`\dim_add:Nn` [2231](#)
`\dim_compare:nNnTF` [113](#), [2181](#), [2216](#), [2232](#), [2239](#), [2500](#), [2945](#)
`\dim_eval:n` [2201](#), [2247](#)
`\dim_max:nn` [3225](#)
`\dim_min:nn` [2248](#)
`\dim_new:N` [61](#), [1948](#), [2051](#), [2052](#), [2053](#), [2256](#)
`\dim_set:Nn` [1939](#), [1982](#), [1983](#),
[1984](#), [2180](#), [2186](#), [2217](#), [2230](#), [2234](#), [2261](#), [2507](#), [3223](#), [3942](#)
`\dim_sub:Nn` [2238](#)
`\dim_to_decimal:n` [37](#), [3783](#), [3784](#), [3821](#)
`\dim_to_decimal_in_unit:nn` [4952](#)
`\dim_use:N` [121](#), [3942](#)
`\dim_zero:N` [2267](#)
`\c_max_dim` [2181](#), [2500](#)
`\c_zero_dim` [2189](#), [2201](#), [2216](#), [2232](#), [2239](#), [2425](#), [2945](#), [3633](#)
`\dimexpr` [3312](#), [3332](#), [3352](#)
`\directlua` [706](#), [707](#)
`\document` [5108](#), [5111](#)

E

`\edef` [4616](#), [4617](#)
`\else` . [1694](#), [1695](#), [1699](#), [1700](#), [2552](#), [2561](#), [2570](#), [2586](#), [2591](#),
[2624](#), [2629](#), [2701](#), [2709](#), [2725](#), [2731](#), [2737](#), [2749](#), [2758](#),
[2816](#), [2821](#), [2840](#), [2845](#), [2864](#), [2869](#), [2878](#), [2898](#), [2914](#),
[2927](#), [3440](#), [3468](#), [4628](#), [4629](#), [4675](#), [4714](#), [4745](#), [4804](#), [4867](#)
else commands:
`\else:` [491](#), [731](#), [824](#), [870](#), [1967](#), [3059](#), [3259](#)
`\em` [1692](#)
`\eminnershape` [1694](#), [1695](#), [1700](#), [1702](#)
`\end` [3319](#), [3339](#), [3359](#), [4997](#)
`\endCJKfilltwosides` [1914](#)
`\endcsname` [2823](#),
[2832](#), [2833](#), [2834](#), [2841](#), [2859](#), [2860](#), [2862](#), [2865](#), [2874](#),
[2875](#), [2876](#), [2879](#), [4623](#), [4625](#), [4626](#), [4627](#), [4630](#), [4632](#), [4636](#)
`\endCTEXfilltwosides` [1914](#), [1919](#)
`\endgroup` [2861](#), [2897](#), [2922](#), [3321](#), [3341](#), [3361](#), [4616](#), [4624](#), [4639](#)
`\endinput` [4621](#)
`\endlinechar` [4595](#), [4596](#)
`\everypar` [2820](#), [2917](#), [2929](#)
`\EverySelectfont` [2014](#), [2017](#)
exp commands:
`\exp_after:wN`
[120](#), [777](#), [793](#), [795](#), [870](#), [873](#), [3129](#), [3765](#), [3812](#), [3847](#), [5259](#)
`\exp_args:cc` [3373](#)
`\exp_args:Nc` [3101](#), [3106](#), [5213](#)
`\exp_args:Nco` [3211](#)
`\exp_args:Nnc` [102](#)
`\exp_args:NNf` [1623](#)
`\exp_args:NNNo` [3941](#)
`\exp_args:Nno` [3385](#)
`\exp_args:No` [560](#), [1821](#), [3125](#), [4929](#), [5275](#)

`\exp_args:Nx` [295](#), [588](#), [662](#), [2221](#), [3761](#), [3940](#)
`\exp_last_unbraced:Nn` [5224](#)
`\exp_last_unbraced:NNn` [2005](#)
`\exp_not:N` [125](#), [126](#),
[470](#), [590](#), [591](#), [592](#), [739](#), [740](#), [742](#), [812](#), [848](#), [958](#), [1043](#),
[1044](#), [1045](#), [1046](#), [1402](#), [1523](#), [1751](#), [1752](#), [1753](#), [1767](#),
[1903](#), [1904](#), [1905](#), [1932](#), [2386](#), [2387](#), [2388](#), [2393](#), [2395](#),
[2396](#), [2397](#), [2398](#), [2401](#), [2402](#), [2405](#), [2417](#), [2418](#), [2419](#),
[2420](#), [2421](#), [2430](#), [2431](#), [2432](#), [2434](#), [2437](#), [2452](#), [2454](#),
[2456](#), [2998](#), [3000](#), [3001](#), [3002](#), [3003](#), [3113](#), [3114](#), [3123](#),
[3124](#), [3161](#), [3163](#), [3164](#), [3781](#), [3983](#), [5204](#), [5205](#), [5206](#), [5294](#)
`\exp_not:n` [69](#),
[390](#), [741](#), [743](#), [1039](#), [1040](#), [1041](#), [1042](#), [1175](#), [1176](#), [1403](#),
[1404](#), [1405](#), [1429](#), [1523](#), [1746](#), [1768](#), [1988](#), [2438](#), [2493](#),
[3132](#), [3374](#), [5087](#), [5111](#), [5222](#), [5294](#), [5295](#), [5296](#), [5351](#), [5352](#)
`\exp_stop_f:` [729](#), [730](#)
`\expandafter`
[4598](#), [4599](#), [4623](#), [4624](#), [4625](#), [4626](#), [4627](#), [4632](#), [4636](#), [4637](#)
`\ExplSyntaxOff` [66](#), [72](#), [479](#),
[522](#), [526](#), [530](#), [534](#), [685](#), [1693](#), [2007](#), [3234](#), [3302](#), [4958](#), [5157](#)
`\ExplSyntaxOn` [71](#), [717](#), [3366](#), [4898](#)

F

`\familydefault` [1276](#), [1287](#), [1749](#)
`fancyhdr` [30](#), [343](#)
`\fangsong` [4577](#)
`\fi` .. [2543](#), [2548](#), [2554](#), [2563](#), [2572](#), [2575](#), [2590](#), [2595](#), [2602](#),
[2613](#), [2628](#), [2633](#), [2642](#), [2664](#), [2685](#), [2693](#), [2694](#), [2697](#),
[2701](#), [2711](#), [2729](#), [2735](#), [2741](#), [2752](#), [2761](#), [2765](#), [2773](#),
[2774](#), [2783](#), [2789](#), [2801](#), [2806](#), [2812](#), [2818](#), [2825](#), [2827](#),
[2849](#), [2850](#), [2866](#), [2880](#), [2883](#), [2902](#), [2911](#), [2930](#), [2931](#),
[3254](#), [3257](#), [3261](#), [3271](#), [3272](#), [3277](#), [3285](#), [3288](#), [3293](#),
[3296](#), [3315](#), [3335](#), [3355](#), [3440](#), [3468](#), [4628](#), [4629](#), [4638](#), [4891](#)
fi commands:
`\fi:` [409](#), [538](#),
[575](#), [594](#), [733](#), [824](#), [870](#), [1967](#), [2012](#), [3062](#), [3263](#), [3279](#), [3856](#)
`\figurename` [2320](#)
`figurename` [12](#)
file commands:
`\g_file_curr_name_str` [677](#), [1821](#)
`\file_if_exist:nTF`
[48](#), [1808](#), [1810](#), [1812](#), [1824](#), [3976](#), [4159](#), [4166](#), [4168](#), [4416](#)
`\file_if_exist_input:nTF` [86](#)
`\file_input:n` [78](#), [87](#), [89](#), [511](#), [547](#)
`fntef` [31](#), [343](#)
`\font` [4049](#), [4050](#), [4051](#), [4052](#), [4053](#), [4054](#)
`\fontfamily` [1269](#)
`fontset` [7](#), [227](#), [3996](#)
`\fontsize` [3765](#)
fontspec commands:
`\g_fontspec_encoding_tl` [963](#)
`\l_fontspec_family_tl` [1462](#)
`\fontspec_set_family:Nnn` [1041](#), [1439](#)
`\fontspec_visible_space:` [727](#)
`\fontspec_visible_space_fallback:` [732](#)

fontspec internal commands:

_fontspec_fontname_wrap:n	56, 982
\footnotesep	3941
\footnotesize	3878, 3917, 3941
fp commands:	
\fp_compare:nNnTF	3725, 3752, 3938
\fp_eval:n	2224, 4948
\fp_set:Nn	3726, 3753
\fp_use:N	2161, 3940
\c_nan_fp	184
\c_zero_fp	2161, 3725, 3752, 3938

G

GBK	6, 223
\gdef	3024, 3025, 3026, 3027, 3030, 3031, 3032, 3033
\global	2705, 2916, 2919
\globaljfont	795
\glueexpr	2856, 2871, 2892, 2899, 2900, 2926, 3310, 3320, 3330, 3340, 3350, 3360
group commands:	
\group_begin:	474, 722, 757, 779, 805, 847, 874, 922, 987, 1037, 1348, 1433, 1518, 1538, 1554, 1745, 1992, 2055, 2392, 2400, 2468, 2943, 3048, 3170, 3229, 3941, 4924, 5105, 5181, 5189, 5198, 5253
\group_end:	478, 487, 726, 762, 783, 818, 851, 878, 926, 1002, 1048, 1368, 1440, 1524, 1550, 1561, 1757, 2005, 2073, 2403, 2407, 2492, 2947, 3063, 3193, 3233, 3941, 4935, 5105, 5201, 5248, 5276, 5310, 5334
\group_insert_after:N	815, 816
\gtfamily	1694, 1699

H

hbox commands:

\hbox_set:Nn	1938, 3222
heading	9, 306
\heiti	4576
\hskip	2856, 2871, 2892, 2899, 2900, 2925, 2926, 3285, 3288
\hspace	2585, 2623, 2973, 2981, 3153
\Huge	2645, 2676, 2777, 2795, 3444, 3475, 3895, 3934
\huge	2605, 2641, 2658, 2772, 3426, 3443, 3447, 3473, 3481, 3894, 3933
hyperref	31, 343
\hypersetup	360, 361, 1834
\hyphenchar	4049, 4050, 4051, 4052, 4053, 4054

I

if commands:

\if_bool:N	488, 582
\if_case:w	405, 3852
\if_cs_exist:N	555, 1991
\if_false:	108
\if_meaning:w	1966
\if_true:	107
\ifctexpdf	106
\ifdim	2810, 2852, 2888, 2906
\IfNoValueF	2124, 2130
\IfNoValueTF	1190, 1211, 1487, 1592, 1625, 1627, 2464

\ifnum	2581, 2600, 2619, 2640, 2714, 2770, 2836, 2864, 2878, 3254, 3257, 3261, 3268, 3269, 3277, 3285, 3288, 3293, 3296, 4628
\ifodd	2548, 2550, 2575, 2582, 2613, 2620, 2664, 2685, 2707, 2718, 2765, 2783, 2789, 2801, 2814, 2825, 2841, 2853, 2889, 2907, 2911, 3313, 3333, 3353
\ifx	4626, 4627, 4632
\ifzhmappdf	4644, 4711, 4742, 4773, 4840
\ignorespaces	2932
\immediate	4635
indent	30, 203
\indexname	2323
indexname	12
\input	4897, 4961
\insertcontinuationtext	2332, 2341, 2348
\insertpart	3317, 3318
\insertromanpartnumber	3308, 3413
\insertsection	3337, 3338
\insertsectionnumber	3328, 3512
\insertsubsection	3357, 3358
\insertsubsectionnumber	3348, 3541

int commands:

\int_compare:nNnTF	561, 2197, 2241, 3600, 3616, 3632, 3723, 3750
\int_const:Nn	974
\int_div_truncate:nn	80, 568
\int_eval:n	1488, 1490, 1491, 1593, 1595, 1596
\int_from_hex:n	560
\int_gincr:N	1133, 1453
\int_gset:Nn	96, 163, 164, 165, 311, 313, 379, 383, 3724, 3751
\int_incr:N	2995
\int_mod:nn	570
\int_new:N	59, 95, 153, 1456
\int_set:Nn	154, 545, 551, 559, 2194, 2236
\int_sub:Nn	564, 2242
\int_to_Hex:n	562, 567, 569
\int_use:N	471, 551, 862, 1135, 1454, 2064, 2999, 4929
\int_zero:N	2992
\c_zero_int	2197
\interlinepenalty	2597, 2637, 2654, 2672, 2769, 2793, 2857, 2858, 2893, 2894

iow commands:

\iow_close:N	4918
\iow_indent:n	5092
\iow_new:N	4920
\iow_now:Nn	4930
\iow_open:Nn	4915
\itemsep	3875, 3886, 3914, 3925
\itshape	1694, 1695, 1699, 1700

J

\jfam	914, 975, 1689
-------	----------------

K

\kaishu	4578
\kanjifamily	1662, 1669, 1676, 1681
\kanjifamilydefault	1657

kernel internal commands:

\l_kernel_expl_bool 70

keys commands:

\l_keys_choice_tl 296
 \keys_define:nn 157, 390, 393, 979,
 1341, 1450, 1527, 1565, 1604, 2078, 2099, 2132, 2166,
 2289, 2315, 2343, 2353, 2356, 2359, 2493, 3008, 3949, 3996
 \keys_if_exist:nnTF 1602
 \l_keys_key_tl 142, 144, 146
 \keys_set:nn 29, 173,
 179, 235, 244, 253, 279, 289, 296, 303, 333, 339, 2117,
 2124, 2130, 3401, 3459, 3488, 3520, 3555, 3572, 3586,
 3602, 3610, 3618, 3626, 3633, 3634, 3636, 3702, 3704, 4038
 \keys_set_known:nn 3681
 \keys_set_known:nnN 991

L

\labelformat 3385
 \languagealias 3710, 3715, 3736, 3741
 \LARGE 3893, 3932
 \Large 2601, 3424, 3430, 3499, 3503, 3892, 3931
 \large 3524, 3891, 3930
 \lastbox 2920
 \leavevmode 2543, 2806

left commands:

\c_left_brace_str 5245
 \leftmargin 3872, 3883, 3911, 3922
 \leftmargini 3872, 3883, 3911, 3922
 \let 2000, 2839, 2848, 3865, 3904, 4619, 4625
 \linespread 3940
 linespread 10, 183
 linestretch 11, 2166
 \linewidth 2195, 2203, 2235
 \lishu 4580
 \listfigurename 2318
 listfigurename 12
 \listtablename 2319
 listtablename 12
 \LoadClass 413, 418, 423, 428
 \long 4633
 \lstlistingname 740, 741
 \lstlistlistingname 742, 743
 \ltjalchar 730
 \ltjdefcharrange 686, 687, 688, 691, 695, 696, 699, 702, 703
 LTJFONTUID 1450
 \ltjgetparameter 1943, 1970, 2032, 2046

ltjitaliccorr internal commands:

_ctex_ltjitaliccorr 1644
 \ltjsetkanjiskip 1648
 \ltjsetmathletter 924
 \ltjsetparameter 705, 711, 1643
 \ltjsetxkanjiskip 1650

lua commands:

\lua_escape:e
 .. 823, 835, 862, 869, 1478, 1493, 1494, 1511, 1512, 1521

\lua_now:e 723, 797,
 823, 827, 832, 837, 859, 868, 961, 1499, 1523, 1777, 2057

M

\MAKESPA 4921, 4963
 \markboth 2515, 2607, 2635
 \mathgroup 914
 \mathrm 1660
 \mathsf 1667
 \mathtt 1674
 \mcfamily 1699
 \mddefault 959, 965, 971, 1329, 1334

mode commands:

\mode_if_math:TF 908

msg commands:

\msg_critical:nnn 53, 677, 3965
 \msg_error:nn 4016
 \msg_error:nnn 19, 22, 38, 41, 668, 681, 684, 2305, 3766, 3848
 \msg_error:nnnn 3982, 4011
 \msg_fatal:nn 4903
 \msg_new:nn 4901
 \msg_new:nnn 141, 143, 145, 147, 149, 671,
 1031, 1101, 1183, 1890, 2336, 3655, 3966, 4025, 5088, 5165
 \msg_new:nnnn 12, 30, 45, 100, 2307, 3768, 3990, 4019
 \msg_warning:nn 1178, 1894, 2096, 2107, 2358
 \msg_warning:nnn
 171, 177, 205, 215, 230, 239, 248, 277, 287, 301,
 331, 337, 345, 350, 356, 1096, 3674, 3953, 4003, 5094, 5164
 \msg_warning:nnnn 1027, 1903, 1910, 2122, 2128, 2259, 2265

N

\newCJKfontfamily 1202
 \NewDocumentCommand
 1062, 1197, 1202, 1208, 1218, 1224, 1230, 1237, 1243,
 1249, 1255, 1388, 1480, 1585, 1620, 1679, 2117, 2120,
 2126, 2149, 2220, 2257, 2263, 2284, 2286, 2458, 3760,
 3945, 4575, 4576, 4577, 4578, 4580, 4581, 4583, 4585, 4586
 \newfontfeature 976, 977, 978
 \newluafunction 2054
 \newpage 2687, 2692
 \ngostype 44, 1795
 \nobreak 2601,
 2610, 2661, 2772, 2777, 2780, 2795, 2798, 2908, 3425, 3474
 \nobreakspace 2601, 2641
 nocap 30
 \noexpand 4617, 4619, 4620, 4621
 nofonts 30
 \noindent 2938, 2953
 noindent 30
 nopunct 30
 \normalem 1873
 \normalfont 534, 536,
 537, 1234, 1241, 1247, 1312, 1691, 2598, 2599, 2638,
 2639, 2655, 2656, 2673, 2674, 2767, 2768, 2791, 2792, 3003
 \normalsize 3558, 3575, 3589, 3859, 3898, 3936, 3944
 nospace 30
 nozhmap 30

\null 2573, 2690
 \numberline 2724, 2865, 2879, 2981, 2989, 3164, 3220

O

\onecolumn 2568
 or commands:
 \or: 407, 3854

P

\pagestyle 3701
 \par 2506, 2544, 2601, 2641, 2645, 2658,
 2676, 2772, 2777, 2795, 2807, 2908, 3309, 3310, 3317,
 3320, 3329, 3330, 3337, 3340, 3349, 3350, 3357, 3360,
 3403, 3425, 3436, 3454, 3462, 3474, 3515, 3517, 3549, 3551
 paragraph/afterskip 32
 paragraph/beforeskip 32
 paragraph/hang 22
 paragraph/numbering 15
 paragraph/runin 20
 \parindent 2216, 2217, 2239, 2260, 2261,
 2266, 2267, 2598, 2655, 2767, 2791, 3312, 3332, 3352, 3634
 \parsep 3874, 3875, 3885, 3886, 3913, 3914, 3924, 3925
 \part 2541
 part/beforeskip 32
 part/fixbeforeskip 32
 part/hang 22
 part/numbering 15
 part/pagestyle 22, 2468
 \partmark 2514, 2608, 2636
 \partname 2601, 2641, 3308, 3408
 \PassOptionsToClass 412, 417, 422, 427
 \PassOptionsToPackage 489, 1838, 2269, 5096
 \pdfmapline 4645,
 4646, 4647, 4648, 4649, 4650, 4651, 4653, 4654, 4655,
 4656, 4657, 4658, 4659, 4661, 4662, 4663, 4664, 4665,
 4666, 4668, 4669, 4670, 4671, 4672, 4673, 4774, 4775,
 4776, 4777, 4778, 4779, 4780, 4782, 4783, 4784, 4785,
 4786, 4787, 4788, 4790, 4791, 4792, 4793, 4794, 4795,
 4797, 4798, 4799, 4800, 4801, 4802, 4841, 4842, 4843,
 4844, 4845, 4846, 4848, 4849, 4850, 4851, 4852, 4853,
 4855, 4856, 4857, 4858, 4859, 4861, 4862, 4863, 4864, 4865
 \pdfoutput 4628
 \pdfstringdefDisableCommands 3948
 \pingfang 4586
 prg commands:
 \prg_do_nothing:
 918, 1427, 1519, 1919, 1990, 2083, 2965, 5286
 \prg_generate_conditional_variant:Nnn 1088
 \prg_new_conditional:Npnn 821, 1963
 \prg_new_protected_conditional:Npnn 1075, 5256
 \prg_return_false: ... 824, 1085, 1967, 1971, 1975, 5266
 \prg_return_true: 824, 1078, 1083, 1967, 1971, 1975, 5271
 \primitive 4997
 \ProcessKeysOptions 402
 \proofname 2310, 2325
 proofname 12

prop commands:

\prop_get:NnN 1170
 \prop_get:NnNTF
 1077, 1129, 1167, 1420, 1468, 3764, 3819, 3846
 \prop_gpop:NnNTF 1016, 1020, 1555
 \prop_gput:Nnn ... 995, 996, 1043, 1157, 1543, 1579, 3781
 \prop_gremove:Nn 1025
 \prop_if_empty:NnNTF 1091, 1293
 \prop_map_break:n 1307
 \prop_map_inline:Nn 1305
 \prop_new:N 1011, 1012, 1013, 1162, 1563, 1584, 3775
 \protect 2528, 2534, 2724,
 2744, 2745, 2865, 2879, 2981, 2989, 3163, 3175, 3177,
 3181, 3183, 3187, 3189, 3197, 3202, 3214, 3215, 5217, 5218
 \ProvideDocumentCommand 2514
 \ProvidesFile 4633, 4641, 4708, 4739, 4770, 4837
 \providetranslation
 5002, 5003, 5004, 5005, 5006, 5007, 5008,
 5009, 5010, 5011, 5012, 5013, 5014, 5015, 5016, 5017,
 5018, 5019, 5020, 5021, 5022, 5023, 5024, 5025, 5026,
 5027, 5028, 5029, 5030, 5031, 5032, 5033, 5034, 5035,
 5036, 5037, 5038, 5039, 5040, 5041, 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
 punct 10, 30, 282, 2099
 \punctstyle 579, 2104

Q

\quad 2832,
 3254, 3255, 3257, 3258, 3261, 3262, 3274, 3278, 3286,
 3289, 3294, 3297, 3431, 3482, 3492, 3525, 3559, 3576, 3590
 quark commands:
 \q_mark 5260, 5263
 \q_stop 121, 126, 873,
 876, 1385, 1389, 1616, 1621, 5225, 5230, 5239, 5260, 5263
 \quotation 3958

R

\raggedright 2598, 2655, 2767, 2791, 3423, 3472
 \refname 2326, 2331, 2338, 2347, 2355, 2356
 refname 12
 \refstepcounter ... 2584, 2622, 2720, 2843, 3368, 3390, 3391
 \relax 2619, 2640, 2808,
 2844, 2851, 2856, 2871, 2887, 2892, 2899, 2900, 2905,
 2924, 2926, 3285, 3288, 3310, 3312, 3320, 3330, 3332,
 3340, 3350, 3352, 3360, 4591, 4595, 4598, 4626, 4627, 4632
 \renewcommand 2542, 2558, 2700
 \RenewDocumentCommand 1269, 1658, 1665, 1672, 3391
 \RequirePackage 27,
 56, 57, 352, 363, 490, 493, 494, 495, 661, 679, 682,
 719, 739, 1738, 1847, 1856, 1872, 1878, 2270, 3755, 5141
 \RequirePackageWithOptions 5097
 resetalternatfont 28, 1527
 reverse commands:
 \reverse_if:N 582

\rmdefault 1284, 1661, 1751
 \rmfamily 522, 525, 1658, 1660
 \romanfamily 1661, 1668, 1675

S

scan commands:

\scan_stop:
 ... 66, 121, 795, 914, 1520, 1521, 2187, 2195, 4925, 5107
 scheme 9, 317
 \scriptsize 3889, 3928
 \secdef 2555, 2577, 2712
 section/afterskip 32
 section/beforeskip 32
 section/hang 22
 section/numbering 15
 section/runin 20
 \sectionname 3328, 3509
 \selectfont 1069, 1546,
 1559, 1663, 1670, 1677, 1682, 2004, 2011, 2025, 2151, 3765

seq commands:

\seq_clear:N 989
 \seq_concat:NNN 1361
 \seq_const_from_clist:Nn . 441, 2367, 2377, 3038, 4905
 \seq_gput_left:Nn 2373, 2374
 \seq_gput_right:Nn 1095, 3787
 \seq_gset_eq:NN 2372
 \seq_if_empty:NTF 1376
 \seq_if_in:NnTF 447, 1093
 \seq_map_function:NN 3096, 3218
 \seq_map_inline:Nn
 ... 1384, 2485, 2993, 3041, 3051, 3060, 3155, 3379, 4928
 \seq_new:N 1100, 1369, 1370, 2371, 3776
 \seq_put_right:Nn 1632
 \seq_set_filter:NNn 1359
 \seq_set_split:Nnn 1358
 \seq_use:Nnnn 3773
 \setbox 2920
 \setCJKfamilyfont 1114, 1202, 4187, 4193, 4194,
 4195, 4196, 4197, 4198, 4241, 4242, 4250, 4251, 4253,
 4254, 4255, 4256, 4258, 4280, 4282, 4284, 4286, 4325,
 4326, 4327, 4328, 4352, 4357, 4362, 4364, 4406, 4408,
 4410, 4411, 4443, 4444, 4447, 4452, 4453, 4455, 4456,
 4457, 4470, 4471, 4472, 4473, 4474, 4475, 4515, 4516,
 4517, 4518, 4519, 4520, 4533, 4534, 4535, 4568, 4569, 4570
 \setCJKmainfont 1110, 1230, 4178, 4237, 4245,
 4272, 4318, 4340, 4397, 4428, 4432, 4466, 4511, 4529, 4564
 \setCJKmathfont 1230
 \setCJKmonofont 1112, 1230, 4192, 4240,
 4249, 4279, 4324, 4351, 4405, 4441, 4469, 4514, 4532, 4567
 \setCJKromanfont 1236, 1262
 \setCJKsansfont 1111, 1230, 4181, 4185, 4239, 4247,
 4278, 4323, 4346, 4403, 4429, 4439, 4468, 4513, 4531, 4566
 \SetSymbolFont 972, 1333, 1337, 1688
 \sfdefault 1285, 1668, 1752
 \sffamily 526, 529, 1665, 1667

skip commands:

\skip_horizontal:N 1953, 2946
 \skip_if_eq:nnTF 1970, 1974, 2031, 2525, 2531
 \skip_new:N 1962, 2044, 2510
 \skip_set:Nn 1943, 1946,
 2035, 2045, 2188, 2199, 2244, 2252, 2512, 2524, 2530, 2944
 \skip_set_eq:NN 1960, 2037
 \skip_sub:Nn 2501, 2508
 \skip_use:N 2528, 2534
 \skip_zero:N 2208
 \c_zero_skip 2423, 2424, 2476, 2477, 2525, 2531
 \small 3867, 3906
 \songti 4575
 \space 2721, 2772, 3408, 3470, 3509, 3540, 3642
 space 10, 30, 292, 2078
 \special 1720, 1721, 1724,
 1725, 1854, 1868, 4676, 4677, 4678, 4679, 4680, 4681,
 4682, 4684, 4685, 4686, 4687, 4688, 4689, 4690, 4692,
 4693, 4694, 4695, 4696, 4697, 4699, 4700, 4701, 4702,
 4703, 4704, 4715, 4716, 4717, 4718, 4719, 4721, 4722,
 4723, 4724, 4725, 4727, 4728, 4729, 4730, 4732, 4733,
 4734, 4735, 4746, 4747, 4748, 4749, 4750, 4752, 4753,
 4754, 4755, 4756, 4758, 4759, 4760, 4761, 4763, 4764,
 4765, 4766, 4805, 4806, 4807, 4808, 4809, 4810, 4811,
 4813, 4814, 4815, 4816, 4817, 4818, 4819, 4821, 4822,
 4823, 4824, 4825, 4826, 4828, 4829, 4830, 4831, 4832,
 4833, 4868, 4869, 4870, 4871, 4872, 4874, 4875, 4876,
 4877, 4878, 4880, 4881, 4882, 4883, 4885, 4886, 4887, 4888
 \SplitArgument 1481, 1586, 2459
 str commands:
 \c_backslash_str 5244
 \c_colon_str 900, 901, 902
 \str_case:nnTF 1749, 1795, 5215, 5230
 \str_case_e:nn 1282
 \str_case_e:nnTF 1108, 1119
 \str_const:Nn 43, 50, 1820, 1828
 \str_if_empty:NTF 5301, 5337, 5346
 \str_if_eq:eeTF 578, 1297, 4002, 5221
 \str_if_eq:nnTF 492, 583,
 947, 1276, 1843, 1851, 3706, 3733, 3747, 4005, 4022, 4029
 \str_lower_case:n 466
 \str_new:N 5141, 5250, 5251, 5252
 \str_set:Nn 5268, 5269, 5270
 \str_upper_case:n 1821
 \strutbox 3942
 sub3section 9, 309
 sub4section 9, 309
 subparagraph/afterskip 32
 subparagraph/beforeskip 32
 subparagraph/hang 22
 subparagraph/numbering 15
 subparagraph/runin 20
 subsection/afterskip 32
 subsection/beforeskip 32
 subsection/hang 22
 subsection/numbering 15

subsection/runin 20
 \subsectionname 3348, 3540
 subsubsection/afterskip 32
 subsubsection/beforeskip 32
 subsubsection/hang 22
 subsubsection/numbering 15
 subsubsection/runin 20
 \symmincho 1689

sys commands:

\c_sys_engine_str 44
 \sys_if_engine_pdftex:TF
 131, 4173, 4265, 4333, 4421, 4462, 4525
 \sys_if_engine_uptex:TF
 4218, 4306, 4385, 4424, 4495, 4553
 \sys_if_engine_xetex:TF 4899
 \sys_if_output_pdf:TF 106, 476, 1846, 4267, 4335

T

\tablename 2321
 tablename 12
 \TeX 3946

T_EX and L^AT_EX 2_ε commands:

\@italiccorr 1645
 \@par 2857, 2893, 3493, 3526, 3560, 3604, 3620
 \@afterheading 2615, 2666, 2751, 2760, 2913
 \@afterindentfalse 2553, 2706, 2710, 2811, 2817
 \@afterindenttrue 2551, 2708, 2809, 2815
 \@Alph 3638, 3643
 \@chapapp 2721, 2772, 3270, 3293
 \@chapter 2712, 2713
 \@chinese 2272, 2281
 \@clubpenalty 2928
 \@currentHref 3072
 \@currentlabel 3370, 3374
 \@currentxt 4, 5087, 5136
 \@currname 4, 5087, 5137
 \@dblarg 2830
 \@defaultunits 37
 \@empty 102, 2839, 2848
 \@endpart 2648, 2679, 2681
 \@EverySelectfont@Init 1991, 1999
 \@firstoftwo 3214
 \@hangfrom 2937, 2952
 \@ifclassloaded 3651
 \@ifpackagelater 37, 40, 667, 680, 683, 1879
 \@ifpackageloaded 18, 21, 676, 1831, 3135, 3238, 3387, 5118
 \@ifpackagewith 3080, 3089, 3140
 \@ifstar 2828
 \@listI 3865, 3904
 \@listi 3865, 3872, 3883, 3904, 3911, 3922
 \@M 2597,
 2637, 2654, 2672, 2769, 2793, 2857, 2858, 2893, 2894, 2921
 \@makechapterhead 2748, 2750, 2762
 \@makeschapterhead 2757, 2759, 2786

\@minus 3494, 3527, 3561, 3577, 3591, 3861,
 3863, 3869, 3871, 3873, 3874, 3880, 3882, 3884, 3885,
 3900, 3902, 3908, 3910, 3912, 3913, 3919, 3921, 3923, 3924
 \@namedef . 3175, 3177, 3181, 3183, 3187, 3189, 3197, 3202
 \@nameuse 3175, 3177, 3187, 3189, 3197
 \@ne 3133, 3257, 3288
 \@nil 848
 \@nobreakfalse 2915
 \@noskipsecfalse 2919
 \@noskipsectrue 2916
 \@onlypreamble 37, 438, 503,
 1258, 1259, 1260, 1261, 1262, 1263, 1715, 1729, 1737, 3989
 \@part 2555, 2577, 2579
 \@pkgextension 6, 7, 8, 9, 20, 23, 678, 5136
 \@plus 3437,
 3438, 3494, 3495, 3527, 3528, 3561, 3562, 3577, 3591,
 3605, 3621, 3861, 3862, 3863, 3869, 3870, 3871, 3873,
 3874, 3880, 3881, 3882, 3884, 3885, 3900, 3901, 3902,
 3908, 3909, 3910, 3912, 3913, 3919, 3920, 3921, 3923, 3924
 \@popfilename 133, 5134
 \@schapter 2712, 2753
 \@seccntformat 2831, 2844
 \@secondoftwo 3215
 \@secpenalty 2822, 3497, 3530, 3564, 3578, 3592
 \@sect 2830, 2835
 \@setfontsize 3851
 \@spart 2555, 2577, 2650
 \@ssect 2829, 2885
 \@startsection 94, 96, 103, 2805, 2998
 \@svsec 2839, 2844, 2848, 2856, 2872, 2873
 \@svsechd 2870, 2899, 2900, 2922
 \@tempdima 3223, 3225
 \@tempskipa 2808, 2810, 2811,
 2822, 2851, 2852, 2887, 2888, 2905, 2906, 2909, 2924, 2925
 \@tempswafalse 2571
 \@tempwattrue 2569
 \@topnewpage 2748, 2757
 \@topnum 2705
 \@xsect 2884, 2903, 2904
 \@zhdig 2280
 \@zhnum 2272, 2279
 \abstractname 12
 \addCJKfontfeature 66
 \AfterEndPreamble 133
 \algorithmname 12
 \appendixname 12, 24
 \AtBeginDocument 37, 46
 \AtBeginDvi 43
 \AtBeginShipoutFirst 43
 \AtEndOfClass 38
 \AtEndOfPackage 38
 \AtEndOfPackageFile* 133
 \AtEndPreamble 133
 \baselineskip 10, 11, 77, 80
 \bfseries 48
 \bibname 12

<code>\c@chapter</code>	3643	<code>\CTEX@chapter@numbering</code>	2718, 3033
<code>\c@secnumdepth</code>	2581, 2600, 2619, 2640, 2714, 2770, 2836, 2864, 2878, 3254, 3257, 3261, 3268, 3269, 3277, 3285, 3288, 3293, 3296	<code>\CTEX@chapter@pagestyle</code>	2473, 2704
<code>\c@section</code>	3638	<code>\CTEX@chapter@titleformat</code>	2778, 2796
<code>\caption</code>	31	<code>\CTEX@chapter@tocline</code>	2970
<code>\catcode</code>	36, 45, 45, 124, 134	<code>\CTEX@chaptername</code>	2776
<code>\ccwd</code> ...	11, 11, 25, 26, 31, 31, 37, 75, 76, 76, 79, 80, 80, 81	<code>\CTEX@document@left@hook</code>	5104, 5110
<code>\chapter</code>	9, 15, 16, 22, 24, 24, 86, 93, 107	<code>\CTEX@document@right@hook</code>	5106, 5112
<code>\chaptermark</code>	13	<code>\CTEX@Family@CMap</code>	452, 480, 483
<code>\chaptername</code>	15	<code>\CTEX@fixheadingskip</code>	2499, 2504, 2548, 2613, 2664, 2685, 2783, 2801, 2825, 2911
<code>\char</code>	50	<code>\CTEX@fixskip</code>	2825, 2911, 2960, 2967
<code>\CheckCommand</code>	76	<code>\CTEX@fixtopskip</code>	2496, 2575, 2765, 2789
<code>\Chinese</code>	31	<code>\CTEX@fontfamily</code>	1040, 1269
<code>\chinese</code>	26, 31, 82	<code>\CTEX@hang</code>	2951, 2961, 2968
<code>\CJK@@ignorespaces</code>	47, 596	<code>\CTEX@hangfrom</code>	2856, 2892, 2949
<code>\CJK@encoding</code>	56, 57, 65, 754, 758, 842, 954, 1328, 1331, 1333, 1337, 1478, 1493, 1494, 1511, 1512	<code>\CTEX@hangindent</code> 2603, 2643, 2657, 2675, 2775, 2794, 2934	
<code>\CJK@envStart</code>	46, 46, 47, 590	<code>\CTEX@headingskip</code>	2513, 2549, 2576, 2614, 2665, 2686, 2766, 2784, 2790, 2802, 2826, 2912
<code>\CJK@family</code>	51, 746, 759, 806, 842, 1066, 1545, 1555, 1557, 2025, 2026	<code>\CTEX@hyperheadinghook</code>	3046, 3076
<code>\CJK@ignorespaces</code>	47, 47, 601, 605	<code>\CTEX@ifnamefalse</code>	2516, 2587, 2592, 2625, 2630, 2652, 2670, 2726, 2732, 2738, 2754, 2837, 2846, 2886
<code>\CJK@input</code>	46, 539	<code>\CTEX@ifnametrue</code>	2516, 2583, 2621, 2719, 2842
<code>\CJK@loadBinding</code>	45, 46, 586	<code>\CTEX@makeanchor</code>	2588, 2593, 2626, 2631, 2727, 2847, 3036, 3044, 3056, 3061, 3067
<code>\CJK@loadEncoding</code>	45	<code>\CTEX@makeanchor@chapter</code>	2733, 2739
<code>\CJK@makeActive</code>	46, 585	<code>\CTEX@makeanchor@schapter</code>	2755
<code>\CJK@plane</code>	46, 77, 455, 456, 466, 2014	<code>\CTEX@makeanchor@sect</code>	2838
<code>\CJK@surr</code>	46, 555	<code>\CTEX@makeanchor@spart</code>	2653, 2671
<code>\CJK@upperReset</code>	47	<code>\CTEX@makeanchor@ssect</code>	2829
<code>\CJKfamily</code>	57	<code>\CTEX@part@afterindent</code>	2550
<code>\CJKfamilydefault</code>	46, 47, 61, 61, 69, 70, 70, 71, 71	<code>\CTEX@part@aftername</code>	2604, 2644, 3314
<code>\CJKglue</code>	80, 112	<code>\CTEX@part@afterskip</code>	2612, 2663, 2684, 3320
<code>\CJKhook</code>	47, 47	<code>\CTEX@part@aftertitle</code> ...	2609, 2647, 2660, 2678, 3318
<code>\CJKrmdefault</code>	44	<code>\CTEX@part@beforeskip</code>	2547, 2574, 3310
<code>\CJKsymbol</code>	77	<code>\CTEX@part@break</code>	2545, 2564
<code>\CJKunderdotbasesep</code>	31	<code>\CTEX@part@fixskip</code>	2548, 2575, 2613, 2664, 2685
<code>\CJKunderline</code>	31	<code>\CTEX@part@format</code>	2599, 2639, 2656, 2674, 3311
<code>\clearpage</code>	47	<code>\CTEX@part@indent</code>	3312
<code>\contentsname</code>	11, 12	<code>\CTEX@part@numbering</code>	2582, 2620, 3313
<code>\CS</code>	52	<code>\CTEX@part@pagestyle</code>	2472, 2566
<code>\CTEX@addloflotskip</code>	2521, 2746	<code>\CTEX@part@titleformat</code> ..	2606, 2646, 2659, 2677, 3318
<code>\CTEX@addtocline</code> ...	2538, 2596, 2634, 2742, 2868, 2882	<code>\CTEX@part@tocline</code>	2970
<code>\CTEX@afterindent</code>	2814, 2959, 2966	<code>\CTEX@partname</code>	2604, 2644, 3314
<code>\CTEX@aftertitle</code>	2896, 2901, 2958, 2965	<code>\CTEX@postappendix</code>	3018, 3026, 3032
<code>\CTEX@appendix@number</code>	3013, 3025, 3031	<code>\CTEX@postchapter</code>	3032
<code>\CTEX@appendix@numbering</code>	3014, 3027, 3033	<code>\CTEX@postsection</code>	3026
<code>\CTEX@chapter@afterindent</code>	2707	<code>\CTEX@preappendix</code>	3017, 3024, 3030
<code>\CTEX@chapter@aftername</code>	2776	<code>\CTEX@prechapter</code>	3030
<code>\CTEX@chapter@afterskip</code>	2782, 2800	<code>\CTEX@presection</code>	3024
<code>\CTEX@chapter@aftertitle</code>	2779, 2797	<code>\CTEX@runin</code>	2853, 2889, 2907, 2962, 2969
<code>\CTEX@chapter@beforeskip</code>	2764, 2788	<code>\CTEX@save@appendix</code>	3019, 3022
<code>\CTEX@chapter@break</code>	2703	<code>\CTEX@save@refstepcounter</code>	3390, 3393
<code>\CTEX@chapter@fixskip</code>	2765, 2783, 2789, 2801	<code>\CTEX@section@aftername</code>	3334
<code>\CTEX@chapter@format</code>	2768, 2792	<code>\CTEX@section@afterskip</code>	3340
<code>\CTEX@chapter@lofskip</code>	2474	<code>\CTEX@section@aftertitle</code>	3338
<code>\CTEX@chapter@lotskip</code>	2475	<code>\CTEX@section@beforeskip</code>	3330

\CTEX@section@format	3331	\f@family	51, 759, 798, 806, 1271, 1282
\CTEX@section@indent	3332	\f@series	754, 842
\CTEX@section@numbering	3027, 3333	\f@shape	754, 842
\CTEX@section@titleformat	3338	\f@size	754, 760, 837, 845, 848, 849, 1939, 1982, 3226
\CTEX@sectionname	3334	\fangsong	8
\CTEX@selectfont@hook	1985, 2014, 2017	\figurename	12
\CTEX@selectfont@save	1996, 2000	\font	52
\CTEX@setcurrentlabel@n	3368, 3394	\font@name	51, 52, 763, 770, 771, 774, 777, 795, 796, 813, 891, 892, 893, 894, 895, 902
\CTEX@setheadingskip	2511, 2547, 2574, 2612, 2663, 2684, 2764, 2782, 2788, 2800, 2824, 2910	\fontcharwd	76
\CTEX@subsection@aftername	3354	\fontfamily	61
\CTEX@subsection@afterskip	3360	\footnotesep	112, 112
\CTEX@subsection@aftertitle	3358	\footnotesize	112
\CTEX@subsection@beforeskip	3350	\get@external@font	789
\CTEX@subsection@format	3351	\getanddefine@fonts	880
\CTEX@subsection@indent	3352	\globaljfont	51
\CTEX@subsection@numbering	3353	\H@old@chapter	3050
\CTEX@subsection@titleformat	3358	\hbox	86
\CTEX@subsectionname	3354	\heiti	8
\CTEX@thechapter	3031	\hrule	86
\CTEX@thesection	3025	\Hy@chapapp	2727, 2733, 2739, 2755
\CTEX@titleformat@n	94, 2895, 2901, 2957, 2964	\Hy@driver	1851
\CTEX@titleslabel@clear	3184, 3194	\Hy@MakeCurrentHrefAuto	3069
\CTEX@titleslabel@set	3178, 3190, 3194	\Hy@org@chapter	3050
\CTEX@toc@width@n	3220, 3235, 3245	\Hy@raisedlink	3070
\CTEX@todayold	2288, 2293	\Hy@unicodetrue	73
\CTEX@update@sectionformat@n	2813, 2955	\hyper@anchorend	3073
\CTEX@verbatim@font@hook	1638, 1642	\hyper@anchorstart	3072
\ctex@zhmap@endinput	4893	\HyPsd@ConvertToUnicode	73
\CTEXdigits	26	\HyPsd@LoadUnicode	74
\CTEXifname	16	\HyPsd@pdfencoding	73
\CTEXnumber	26, 26	\HyPsd@ToBigChars	73
\CTEXnumberline	23	\if@mainmatter	2716, 2771, 3268, 3669, 3670
\CTEXoptions	31, 31, 79	\if@nobreak	2819
\ctexset	5, 5, 7, 11, 14, 27, 28, 31, 31, 113	\if@noskipsec	2543, 2806, 2918
\CTEXsetup	31	\if@openright	2559, 2689, 2701, 3440, 3468
\CTEXthechapter	13, 16, 97	\if@tempswa	2695
\CTEXthesection	13, 16, 100	\if@twocolumn	2567, 2747, 2756
\CTEXunderdotbasesep	31	\if@twoside	2688, 3252, 3275
\CTEXunderline	31	\iffalse	54
\curr@fontshape	760, 790, 799, 803, 849	\ifHy@implicit	3049
\DeclareAlternateKanjiFont	65	\ifin@	54, 870
\DeclareFontEncoding	54	\iftrue	54
\DeclareFontFamily	43, 52	\indexname	12
\DeclareFontShape	52	\insertcontinuationtext	12
\DeclareFontShape@	64, 1437	\itdefault	52
\DeclareRobustCommand	135	\itshape	48
\define@newfont	51, 782	\jfam	69
\do@subst@correction	51, 52, 781	\kaishu	8
\em	69	\kanjiskip	77
\escapechar	54	\labelformat	102
\EverySelectfont	76	\leftmark	13
\external@font	792, 793, 795	\linewidth	80
\extract@font	51, 780	\lishu	8
\f@baselineskip	2187	\listfigurename	12
\f@encoding	758, 798, 911	\listtablename	12

<code>\ltj@does@alt@set</code>	53	<code>\ProvidesFile</code>	125
<code>\ltj@getjfontnumber</code>	53, 858	<code>\ps@fancy</code>	3281
<code>\ltj@set@stackfont</code>	900, 901, 902	<code>\ps@headings</code>	3251
<code>\ltj@allalchar</code>	724	<code>\refname</code>	12, 12
<code>\ltj@curjfont</code>	52, 77	<code>\refstepcounter</code>	102
<code>\ltj@pickup@altfont@auxy</code>	843	<code>\relax</code>	46, 80
<code>\ltj@pickup@altfont@copy</code>	856	<code>\reset@font</code>	536
<code>\ltj@setpar@global</code>	68, 899, 1648, 1650	<code>\rightmark</code>	13
<code>\ltj@tempcntc</code>	53, 862	<code>\rmfamily</code>	49
<code>\ltjalchar</code>	50	<code>\roman</code>	26
<code>\ltjsetkanjiskip</code>	68	<code>\scantokens</code>	135
<code>\ltjsetparameter</code>	68	<code>\scriptfont@name</code>	894, 901
<code>\ltjsetxkanjiskip</code>	68	<code>\section</code> 9, 14, 15, 20, 20, 21, 21, 22, 23, 24, 24, 32, 84, 100	
<code>\m@ne</code>	2581, 2600, 2714, 2770, 3261, 3268, 3269, 3293, 4049, 4050, 4051, 4052, 4053, 4054	<code>\section*</code>	15
<code>\markboth</code>	13	<code>\sectionmark</code>	13
<code>\markright</code>	13	<code>\sectionname</code>	15
<code>\math@bgroup</code>	910	<code>\sectionpage</code>	14, 101
<code>\math@egroup</code>	915	<code>\selectfont</code>	53, 70, 76, 77, 80
<code>\math@fonts</code>	897	<code>\setCJKfamilyfont</code>	27, 27
<code>\mathbb</code>	112	<code>\setCJKmainfont</code>	27
<code>\maxdimen</code>	11, 80	<code>\sf@size</code>	893
<code>\meaning</code>	135, 136, 136, 137	<code>\sffamily</code>	49
<code>\newcommand</code>	135	<code>\size@update</code>	80, 80, 2008, 2009, 2151, 2165
<code>\newrobustcmd</code>	135	<code>\sldefault</code>	52
<code>\newtitlemark</code>	99	<code>\songti</code>	8
<code>\nfss@catcodes</code>	56	<code>\special</code>	7, 72
<code>\ngostype</code>	71	<code>\split@name</code>	848
<code>\normalsize</code>	8, 108	<code>\SplitArgument</code>	85
<code>\not@math@alphabet</code>	1660, 1667, 1674	<code>\ssf@size</code>	895
<code>\nouppercase</code>	46	<code>\strutbox</code>	112
<code>\p@</code>	1939, 1982, 2641, 2744, 2745, 2763, 2772, 2781, 2787, 2799, 3226, 3436, 3464, 3465, 3466, 3467, 3474, 3861, 3862, 3863, 3869, 3870, 3871, 3873, 3874, 3880, 3881, 3882, 3884, 3885, 3900, 3901, 3902, 3908, 3909, 3910, 3912, 3913, 3919, 3920, 3921, 3923, 3924	<code>\subparagraph</code>	9, 20, 36
<code>\pagenumbering</code>	26	<code>\subsectionname</code>	15
<code>\pagestyle</code>	10	<code>\subsectionpage</code>	14, 101
<code>\paragraph</code>	9, 20, 36	<code>\subst@correction</code>	809, 816
<code>\parindent</code>	10, 31, 79, 81, 112	<code>\tablename</code>	12
<code>\parskip</code>	86	<code>\textfont@name</code>	892, 900
<code>\part</code>	14, 20, 21, 22, 86, 93	<code>\tf@size</code>	891
<code>\partmark</code>	87	<code>\thechapter</code>	13, 16, 24
<code>\partname</code>	15	<code>\thesection</code>	13, 16, 24, 100
<code>\partpage</code>	14, 101	<code>\titleformat</code>	96
<code>\PassOptionsToClass</code>	43	<code>\titlespacing</code>	96
<code>\PassOptionsToPackage</code>	72	<code>\today</code>	11, 11
<code>\pdfmapline</code>	118, 121	<code>\trans@languagepath</code>	3718, 3744
<code>\pdfstringdef</code>	73	<code>\ttfamily</code>	49
<code>\pickup@font</code>	51	<code>\ttl@a</code>	3153, 3161
<code>\pickup@jfont</code>	786	<code>\ttl@chapterout</code>	3148
<code>\pingfang</code>	8	<code>\ttl@extract</code>	96
<code>\ProcessKeysOptions</code>	29	<code>\ttl@setifthe</code>	3207
<code>\proofname</code>	12, 83	<code>\ttl@setsubmark</code>	98, 3180, 3186
<code>\protected</code>	98	<code>\ttl@settopmark</code>	98, 3174, 3205
<code>\protected@edef</code>	2285, 2287, 2844, 3370	<code>\ttl@tocpart</code>	3150, 3152
		<code>\ttlh@hang</code>	3114, 3124
		<code>\ttlh@runin</code>	3113, 3123
		<code>\uppercase</code>	46
		<code>\use@mathgroup</code>	906, 937
		<code>\usepackage</code>	30, 31

- \verbatim@font 1636, 1640
- \wrong@fontshape 51
- \x@protect 5218
- \xdef 98
- \XeTeXglyphbounds 129
- \xkanjiskip 77
- \yahei 8
- \youyuan 8
- \z@ 711, 714, 2598, 2655,
2705, 2767, 2791, 2810, 2852, 2888, 2906, 2920, 3133,
3254, 3277, 3285, 3296, 3862, 3870, 3881, 3901, 3909, 3920
- \zhdig 82
- \zhdigits 26
- \zhnum 82
- \zhnumber 26, 26
- \zihao 25
- \ziju 25
- \zw 76
- tex commands:
- \tex_def:D 5294
- \tex_dimexpr:D 121, 2195
- \tex_divide:D 80, 2196, 2237
- \tex_edef:D 5348
- \tex_endlinechar:D 545, 551
- \tex_font:D 470, 729, 4925
- \tex_glueexpr:D 2187
- \tex_iffontchar:D 729
- \tex_iftrue:D 3670
- \tex_ignorespaces:D
..... 1063, 1222, 1227, 2089, 2221, 3761, 5107
- \tex_immediate:D 467
- \tex_kanjiskip:D 1946, 1960, 1974
- \tex_luafunction:D 2075
- \tex_pagegoal:D 2500
- \tex_parskip:D 2508
- \tex_pdffontattr:D 470
- \tex_pdflastobj:D 471
- \tex_pdfobj:D 467
- \tex_prevdepth:D 2507
- \tex_topskip:D 2501
- \tex_XeTeXcharglyph:D 4929
- \tex_XeTeXglyphbounds:D 4953
- \tex_xkanjiskip:D 2033, 2037, 2047
- \the 4595, 4598, 4599, 4618, 4622
- \thechapter 2721, 2724, 2772, 3270, 3293, 3471
- \theparagraph 3574
- \thepart 2585, 2601, 2623, 2641, 3410
- \thesection 3254, 3261, 3277, 3285, 3296, 3491
- \thesubparagraph 3588
- \thesubsection 3257, 3288, 3523
- \thesubsubsection 3557
- \thispagestyle 2565, 2566, 2691, 2702, 2704
- \tiny 3890, 3929
- tl commands:
- \c_space_tl 471
- \tl_clear:N 195, 391,
1577, 1581, 1875, 1886, 2026, 2142, 2465, 2483, 3816, 5302
- \tl_clear_new:N 58, 368
- \tl_const:Nn
.. 3, 411, 416, 421, 426, 518, 656, 678, 954, 969, 1265,
1266, 1267, 1268, 1327, 2161, 2384, 2390, 3652, 3671, 3673
- \tl_gput_right:Nn 448, 5101, 5103, 5125
- \tl_gset:Nn 562, 565, 891,
893, 895, 1301, 1747, 1775, 1793, 1818, 1825, 1826, 4007
- \tl_gset_eq:NN 1045, 1770
- \tl_gset_rescan:Nnn 1308
- \tl_head:N 3820
- \tl_if_blank:nTF 1394, 1490, 1491, 1595, 1596, 1610, 1628
- \tl_if_blank_p:n 1360
- \tl_if_empty:NTF 749, 2214, 2396, 3384, 3970
- \tl_if_empty:nTF 1709, 1722, 5265
- \tl_if_eq:NNTF 1743, 2154, 3731
- \tl_if_exist:NTF 84, 517, 519, 520, 521, 1264,
1266, 1267, 1268, 2310, 2341, 2352, 2355, 3664, 3754, 5124
- \tl_if_in:NnTF 5329
- \tl_map_inline:Nn 1506
- \tl_map_inline:nn 3817
- \tl_new:N 129, 133, 519, 520, 521,
751, 752, 840, 951, 1004, 1030, 1051, 1052, 1073, 1442,
1583, 1763, 2040, 2162, 2175, 2312, 2382, 2383, 3017,
3018, 4164, 5114, 5115, 5124, 5185, 5186, 5305, 5306, 5307
- \tl_put_left:Nn 1999, 5134
- \tl_put_right:Nn
..... 375, 897, 1590, 1623, 2412, 2454, 2456, 3820, 3821
- \tl_replace_all:Nnn 1357, 1398
- \tl_rescan:nn 137, 5347, 5356
- \tl_set:Nn 115, 116, 130,
190, 223, 224, 466, 498, 499, 500, 558, 740, 742, 753,
792, 841, 948, 949, 990, 1056, 1068, 1210, 1271, 1356,
1395, 1397, 1438, 1611, 1615, 1653, 1654, 1655, 1656,
1657, 1765, 1995, 2025, 2041, 2103, 2112, 2138, 2176,
2224, 2313, 2463, 2466, 2470, 3110, 3129, 3148, 3153,
3161, 3730, 3948, 4165, 4169, 4212, 4213, 4214, 5179, 5180
- \tl_set_eq:NN
758, 759, 806, 892, 894, 963, 1057, 1082, 1545, 1557, 2163
- \tl_set_rescan:Nnn 5303
- \tl_tail:N 3115
- \tl_tail:n 3125
- \tl_to_str:n 126, 127, 5246, 5275, 5316, 5317
- \tl_trim_spaces:n 1574
- \today 2288, 2293, 2296, 2301
- today 11, 2288
- token commands:
- \token_to_meaning:N 5260
- \token_to_str:N
.. 54, 873, 882, 891, 893, 895, 1110, 1111, 1112, 1114,
1121, 1122, 1123, 1185, 2338, 2339, 3769, 4932, 5092, 5164
- \toks 4595, 4596, 4598, 4599, 4618, 4622
- \topsep 3873, 3884, 3912, 3923
- \TrimSpaces 2120
- \ttdefault 1286, 1675, 1753

\ttfamily 530, 533, 1672, 1674
 \twocolumn 2696
 \typeout 2721, 2722, 3148

U

um commands:

\um_input_math_symbol_table: 930, 934
 \um_sym:nnn 933

um internal commands:

_um_input_math_symbol_table: 942
 _um_switchto_literal: 939
 _um_sym:nnn 941
 \undefined 4620
 \unless 2853, 2889, 2907
 \unskip 2923
 \updefault ... 959, 965, 967, 971, 973, 1329, 1331, 1334, 1338
 \upshape 1699, 1702

use commands:

\use:N 760, 798, 799, 849, 891, 893, 895, 974,
 1022, 1558, 2524, 2530, 2539, 2944, 2989, 5330, 5350, 5364
 \use:n ... 64, 123, 134, 390, 737, 848, 956, 1140, 1193,
 1199, 1298, 1302, 1400, 1574, 1764, 1917, 2490, 2938,
 2953, 2964, 3231, 3662, 3779, 3979, 5199, 5241, 5292, 5313
 \use:nn 5358
 \use_i:nn 109, 870, 2518, 5145, 5150, 5170, 5175
 \use_i:nnn 263
 \use_ii:nn ... 110, 870, 2516, 2520, 5145, 5150, 5170, 5175
 \use_ii:nnn 135, 268
 \use_iii:nnn 273
 \use_none:n 191, 196, 201, 1039, 1040, 1301, 1435, 1918, 2274
 \use_none:nn 483
 \usebeamercolor 3308, 3328, 3348, 3453, 3514, 3548
 \usebeamerfont 3307, 3317,
 3327, 3337, 3347, 3357, 3452, 3455, 3513, 3516, 3547, 3550
 \uselanguage 3709, 3714, 3735, 3740
 \usepackage 5092
 UTF8 6, 223

V

\verse 3958
 \vfil 2573, 2683
 \vskip 2611, 2614,
 2641, 2662, 2665, 2686, 2772, 2781, 2784, 2799, 2802,
 2909, 2912, 3309, 3329, 3349, 3436, 3454, 3474, 3515, 3549
 \vspace 2576, 2763, 2766, 2787, 2790

W

winfonts 30
 \write 4635

X

\x 4597, 4600,
 4601, 4602, 4603, 4604, 4605, 4606, 4607, 4608, 4609,
 4610, 4611, 4612, 4613, 4614, 4615, 4616, 4623, 4633, 4637
 \xdef 4636
 \xeCJKsetup 662, 1882, 2084, 2087, 2090, 2105

Y

\yahei 4583, 4585
 \youyuan 4581

Z

\zhdig 2280
 \zhdigits 26, 2287
 zhmap 7, 30, 258
 \zhnum 2279
 zhnum commands:
 \zhnum_counter:n 2271
 \zhnumber 26, 2285
 \zhnumsetup 2297, 2302
 \zhtoday 2296, 2301
 \zihao 25, 3760, 3769
 zihao 8, 160
 \ziju 25, 2220
 \zw 712, 1943, 1983, 2042