

CT_EX 宏集手册

CTEX.ORG

2017/11/22 v2.4.11*

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	24
第 2 节 简明教程	3	8.1 字号与间距	24
2.1 CT _E X 宏集的组成	3	8.2 中文数字转换	25
2.2 CT _E X 宏集的安装和更新	3	8.3 杂项	25
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	25
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	26
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	27
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	27
4.1 编译方式	5	10.2 宏包载入后的配置	28
4.2 中文编码	6	10.3 配置标题中文翻译	28
4.3 中文字库	6	10.4 自定义字体集	28
第 5 节 排版格式设定	8	第 11 节 对旧版本的兼容性	28
5.1 文档默认字号	8	11.1 CT _E X 0.8a 及以前的版本	28
5.2 章节标题风格	9	11.2 CT _E X 0.9–CT _E X 1.0d	29
5.3 排版方案选项	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	30
第 6 节 文档汉化	11	11.4 CT _E X 2.2 之前的版本	31
6.1 日期汉化	11	11.5 CT _E X 2.4.1 和 2.4.2	31
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	31
6.3 页面格式设置与汉化	12	第 13 节 开发人员	32
第 7 节 章节标题格式设置	14	第 14 节 代码实现	34
		版本历史	137
		代码索引	138

*`ctex-kit rev. f5ff957`.

第 1 节 介绍

历史

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

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

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

2014 年 3 月，为了适应 L^AT_EX 的最新发展，特别是 L^AT_EX3 的逐渐成熟，李清用 L^AT_EX3 重构了整个宏包的代码，并重新使用 `doc` 和 `DocStrip` 工具进行代码的管理，升级版本号为 2.0，并改称 CT_EX 宏集。

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

最初，Knuth 在设计开发 T_EX 的时候没有考虑到多国语言支持，特别是对多字节的中日韩语言的支持。这使得 T_EX 以至后来的 L^AT_EX 对中文的支持一直不是很好。即使在 CJK 宏包解决了中文字符处理的问题以后，中文用户使用 L^AT_EX 仍然要面对许多困难。这些困难里，以章节标题的中文化为最。由于中文和西文语言习惯的差异，用户很难使用标准文档类中的代码结构来表达中文标题。于是，用户不得不对标准文档类做较大的修改。除此之外，日期格式、首行缩进、中文字号和字距等细节问题，也需要精细的调校。我们设计 CT_EX 宏集的目的之一就是解决这些 L^AT_EX 文档的汉化难题。

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

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

关于宏集名字的说明

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^AT_EX 宏包和文档类的集合。社区另有发布名为 [CT_EX 套装](#) 的 T_EX 发行版，与本文档所述的 CT_EX 宏集并非是一事物。

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

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 C_TE_X 宏集组成

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

表 1 C_TE_X 宏集的组成

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

2.2 C_TE_X 宏集的安装和更新

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

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

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

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

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

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

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

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

⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

2.3 使用 CT_EX 文档类

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

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

例 1

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

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

以下是使用 `ctexbeamer` 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

`\ctexset {<键值列表>}`

New: 2014-03-18

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

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

例 4

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

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

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

表 2 CT_EX 宏集的中文支持方式

编译方式	(pdf)L ^A T _E X	X _Y L ^A T _E X	LuaL ^A T _E X	upL ^A T _E X [*]
支持宏包	CJK	xeCJK	LuaTeX-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 $\text{X}_{\text{L}}\text{A}_{\text{T}}_{\text{E}}\text{X}$ 、 $\text{Lua}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 及 $\text{up}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 编译时, $\text{CT}_{\text{E}}\text{X}$ 宏集使用 (且仅能使用) UTF-8 编码; 而因为历史原因, 在用户使用 $\text{L}_{\text{A}}\text{_{T}}_{\text{E}}\text{X}$ 及 $\text{pdf}_{\text{L}}\text{_{T}}_{\text{E}}\text{X}$ 编译时, 宏包默认使用 GBK 编码。用户需要保证编译方式、源文件编码、宏包编码选项三者一致。关于宏包编码选项, 可以参考 4.2 节。

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

4.2 中文编码

GBK ☆
UTF8 ☆

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

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

4.3 中文字库

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

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

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

¹ Windows Vista 及以后的 Windows 操作系统。

² Windows XP 及以前的 Windows 操作系统。

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

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

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

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

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

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

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

Updated: 2014-03-08

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

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

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

New: 2014-03-08

指定 $\text{CT}_{\text{E}}\text{X}$ 宏集加载的字库。

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

$\text{CT}_{\text{E}}\text{X}$ 预定义了以下六种中文字库。

adobe	使用 Adobe 公司的四款中文字体, 不支持 $\text{pdf}\text{\LaTeX}$ 。
fandol	使用 Fandol 中文字体, 不支持 $\text{pdf}\text{\LaTeX}$ 。
founder	使用方正公司的中文字体。
mac	使用 Mac OS X 系统下的华文字体, 不支持 $(\text{pdf})\text{\LaTeX}$ 。
ubuntu	使用 Ubuntu 系统下的文泉驿和文鼎字体。
windows	使用简体中文 Windows 系统下的中文字体, 自动判断 Windows 系统版本, 采用 <code>windowsnew</code> 或 <code>windowsold</code> 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 $(\text{pdf})\text{\LaTeX}$ 编译的时候, 若设置 `zhmap = false` (比如需要使用 $\text{\LaTeX} + \text{Dvips}$ 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 $\text{CT}_{\text{E}}\text{X}$ 预定义的中文字库, 可以设置 `fontset` 为下述值之一。

none	不配置中文字体, 需要用户自己配置。
<code><name></code>	这里 <code><name></code> 为自定义的名字。 $\text{CT}_{\text{E}}\text{X}$ 宏集将载入名为 <code>ctex-fontset-<name>.def</code> 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 <code>ctex-fontset-<name>.def</code> 的文件, 在这个文件里面自定义中文字体。然后通过使用 <code>fontset=<name></code> 选项来调用它。字体配置文件的具体写法可以参考 $\text{CT}_{\text{E}}\text{X}$ 宏集 <code>fontset</code> 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}
```

$\text{CT}_{\text{E}}\text{X}$ 宏集预定义的中文字库还定义了一些字体命令。

<code>\songti</code>	宋体, CJK 等价命令 <code>\CJKfamily{zhsong}</code> 。
----------------------	--

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。

¹⁰`zhmCJK` 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`。

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

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

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

在 windowsnew 字库中,还有 `\yahei`。

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

第 5 节 排版格式设定

5.1 文档默认字号

`zihao` ☆
New: 2015-05-06

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

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

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

表 4 标准字体命令与字号的对应

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

10pt ☆
11pt ☆
12pt ☆

C_T_E_X 文档类是在 L^AT_EX 标准文档类之上开发的。因此,除了可以使用 C_T_E_X 宏包定义的字号选项之外,还可以使用标准文档类的同类选项(10pt、11pt 和 12pt)。在使用这些来自标准文档类的选项的时候,C_T_E_X 文档类的字号选项会被抑制。亦即,在 `zihao` 选项之后设置 10pt 选项,`zihao` 选项将不再起作用。

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

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

5.2 章节标题风格

heading ☆ heading = `(true|false)`

New: 2014-03-08

本选项只能在调用 `ctex.sty` 时作为宏包选项使用。

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

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

sub3section ☆

sub4section ☆

修改 `\paragraph` 和 `\subparagraph` 的格式。

默认情况下, `\paragraph` 和 `\subparagraph` 会将标题与随后的正文排版在同一个段落。启用 `sub3section` 会将 `\paragraph` 的格式修改为类似 `\section` 的格式, 并将 `\subparagraph` 的格式修改为原本 `\paragraph` 的格式。启用 `sub4section` 会将 `\paragraph` 和 `\subparagraph` 的格式都修改为类似 `\section` 的格式。

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

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

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

5.3 排版方案选项

scheme ☆ scheme = `(chinese|plain)`

New: 2015-04-11

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

`chinese` 对 `beamer` 以外的文档类, 调整默认字号为五号字, 并调整行距为 1.3; 汉化文档中的标题名字 (如“图”、“表”、“目录”和“参考文献”等, 见 6.2 节); 在 `heading = true` 的情况下¹³ (5.2 节), 还会将章节标题的风格修改为中文样式 (见 7 节)。

当关闭 `heading` 选项的 `ctex` 宏包与标准文档类或其衍生文档类联用时, 会载入 `indent-first` 宏包, 以实现章节标题后的段首缩进。

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

punct punct = `(quanjiao|banjiao|kaiming|CCT|plain)`

Updated: 2014-04-11

设置标点处理格式。预定义好的格式有:

`quanjiao` 全角式: 所有标点占一个汉字宽度, 相邻两个标点占 1.5 汉字宽度;

`banjiao` 半角式: 所有标点占半个汉字宽度;

`kaiming` 开明式: 句末点号¹⁴用占一个汉字宽度, 标号和句内点号占半个汉字宽度;

`CCT` CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

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

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

plain 原样(不调整标点间距)。

space space = $\langle true|false|auto \rangle$

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 (pdf)LaTeX 编译时);等同于 auto 的效果(使用 XeLaTeX 编译时)。不建议使用该选项。
- auto 根据空格后面的情况决定是否保留:如果空格后面是汉字,则忽略该空格,否则保留。

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

使用 LuaLaTeX 及 upLaTeX 编译的时候,该选项无效:汉字间的空格以及汉字与西文字符之间的空格总是有效,不会被忽略,但可以自动忽略掉由换行产生的空格。

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

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($\backslash baselineskip$)距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。对 beamer 不改变行距,即使用默认的单倍行距。
- scheme = plain CTeX 宏集默认不调整行距倍数,文档中的行距由所选文档类和其他宏包或用户设置决定。

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

New: 2014-03-13

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

$\langle \text{数值或带单位的数值} \rangle$ 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 $\backslash ccwd$;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 = $\langle \text{数值或长度} \rangle$

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 $\backslash ccwd$ 的倍数。

¹⁵LaTeX 将单个换行视作一个空格。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化,使之以中文的方式显示今天的日期。如编译本文档的日期就是“2017 年 11 月 22 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“2017 年 11 月 22 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一七年十一月二十二日”。使用全汉字的日期格式。
`old` 效果为“November 22, 2017”。使用文档原来的(英文)日期格式。

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

例 9

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

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

6.2 文档标题汉化

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

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

`contentsname` ★ `contentsname = <名字>`

设置目录标题名 `\contentsname`。中文默认为“目录”。

`listfigurename` ★ `listfigurename = <名字>`

设置插图目录标题名 `\listfigurename`。中文默认为“插图”。

`listtablename` ★ `listtablename = <名字>`

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

`figurename` ★ `figurename = <名字>`

设置图片环境标题名 `\figurename`。中文默认为“图”。

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

6.3 页面格式设置与汉化

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

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

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

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

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

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

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

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

例 10

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

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

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

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

例 11

```
\documentclass{ctexart}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\usepackage{fancyhdr}
\fancyhf{}
\lhead{\textnormal{\kaishu\rightmark}}
\rhead{--\ thepage\ --}
\pagestyle{fancy}
% \sectionmark 的重定义需要在 \pagestyle 之后生效
\renewcommand\sectionmark[1]{%
  \markright{\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`, `toc line` 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 `toc line` 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/page style = empty,
  chapter = {
    format      = \raggedright,
    page style = empty,
  },
  section = {
    name       = {第,节},
    number    = \chinese{section},
  }
}

```

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

numbering = true|false

控制是否对章节标题编号。对各级标题的默认值均为 true。

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

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

如果没有特别说明，以下将用“...”代表各级章节标题名。

.../name	★
Updated: 2014-03-08	

name = {<前名字>,<后名字>}

name = {<前名字>}

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

例 13

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

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

表 5 name 选项的默认设置

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

.../number	★
------------	---

number = {<数字输出命令>}

设置章节编号的数字输出格式。<数字输出命令> 通常是对应章节编号计数器的输出命令，如 \thesection 或 $\text{\chinese{chapter}}$ 之类。

例 14

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

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

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

表 6 `number` 选项的默认设置

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

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

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

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

```
\CTEXifname
```

New: 2016-09-18

`\CTEXifname` `{<有名字时的格式命令>}` `{<无名字时的格式命令>}`

`\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。

表 8 nameformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\Large\bfseries
part (beamer)	同右	\usebeamerfont{part name} \usebeamercolor[fg]{part name}
part	{}	\huge\bfseries
chapter	{}	\huge\bfseries
section (beamer)	同右	\usebeamerfont{section name} \usebeamercolor[fg]{section name}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection name} \usebeamercolor[fg]{subsection name}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

...

.../numberformat ★ numberformat = {<格式命令>}

.../numberformat+ ★ numberformat+= {<格式命令>}

Updated: 2015-06-19

numberformat 选项用于控制章节编号的格式,作用域仅为编号数字本身。对各级标题默认均为空,当你需要编号的格式和前后的章节名字不一样时可以使用。

numberformat+ 选项用于在原有编号格式后面附加格式命令。

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

例如,我们可以使用 numberformat 特别强调章标题中的数字:

例 17

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

上面的代码在 `scheme = chinese` 时可以做出类似这样的章标题效果：

第 4 章

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

Updated: 2014-03-08

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

`aftername+` 选项用于在原有代码后面附加代码。

表 9 `aftername` 选项的默认设置

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

```
.../titleformat  ★ titleformat = {{格式命令}}
.../titleformat+ ★ titleformat+= {{格式命令}}
```

Updated: 2015-06-30

`titleformat` 选项用于控制标题内容的格式,作用域为章节标题内容。

`titleformat+` 选项用于在原有标题格式后面附加格式命令。

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

例 18

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

上面的代码可以做出类似这样的章标题效果：

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

表 10 titleformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\huge\bfseries
part (beamer)	同右	\usebeamerfont{part title}
part	{}	\Huge\bfseries
chapter	{}	\Huge\bfseries
section (beamer)	同右	\usebeamerfont{section title}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection title}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

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

New: 2015-06-19

aftertitle 选项的参数 {代码} 将被插入到章节标题内容之后。

aftertitle+ 选项用于在原有代码后面附加代码。

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

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

表 11 aftertitle 选项的默认设置

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

表 12 runin 选项的默认设置

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

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

```
subsection/runin  ★
```

```
subsubsection/runin  ★
```

```
paragraph/runin  ★
```

```
subparagraph/runin  ★
```

New: 2015-06-27

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

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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

.../afterindent *afterindent = true|false

New: 2015-06-27

afterindent 选项用于设置章节标题后首段的缩进。

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

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

表 13 afterindent 选项的默认设置

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

.../beforeskip *beforeskip = {<弹性间距>}

Updated: 2016-05-10

beforeskip 选项用于设置章节标题前的垂直间距。

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

表 14 beforeskip 选项的默认设置

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

.../afterskip *afterskip = {<弹性间距>}

Updated: 2015-06-27

afterskip 选项控制章节标题与后面下方之间的距离。

对于 \section 级以下标题,runin 选项会影响 afterskip 选项的意义: 若 runin 为 true,标题与随后正文排在同一段,<弹性间距> 给出水平间距。否则,正文另起一段,<弹性间距> 给出的是垂直间距。

afterskip 选项的默认值,在 scheme 选项的不同取值下相同。

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

.../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 beforeskip 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。

表 15 afterskip 选项的默认设置

标题名	默认值
part (article)	3ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	40pt
section (beamer)	0pt
section	2.3ex plus .2ex
subsection (beamer)	0pt
subsection	1.5ex plus .2ex
subsubsection	1.5ex plus .2ex
paragraph	1em
(sub3section)	1ex plus .2ex
(sub4section)	同上
subparagraph	1em
(sub4section)	1ex plus .2ex

表 16 indent 选项的默认设置

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

chapter/lofskip ★ lofskip = {{弹性间距}}

chapter/lotskip ★ lotskip = {{弹性间距}}

New: 2016-10-01

lofskip 选项控制插图目录(.lof)中,章之间的插图标题的距离。

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

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

.../indent ★ indent = {{缩进间距}}

Updated: 2015-06-27

indent 选项用于设置章节标题本身的首行缩进。

indent 选项的默认值,在 scheme 选项的不同取值下相同。

例如,设置 \section 标题缩进 20pt:

例 19

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

section/hang ★ hang = true|false

subsection/hang ★

subsubsection/hang ★

paragraph/hang ★

subparagraph/hang ★

New: 2016-10-22

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

part/pagestyle ★ pagestyle = {{页面格式}}

chapter/pagestyle ★

New: 2014-03-21

设置 book/ctexbook 或 report/ctexrep 文档类中,\part 与 \chapter 标题所在页的页面格式(page style)。

表 17 pagestyle 选项的默认设置

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

...

break

★

break = {<格式命令>}

...

break+

★

break+= {<格式命令>}

New: 2016-09-19

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

带加号的 break+ 选项则用于在已有格式之后追加新的格式命令。

break 选项的默认值,在 scheme 选项的不同取值下相同。

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

例 20

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

表 18 break 选项的默认设置

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

...

tocline

★

tocline = {<格式定义>}

New: 2016-10-25

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

表 19 tocline 选项的默认设置

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

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

appendix/numbering ★ numbering = true|false

New: 2015-06-21

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

用法与普通章节 numbering 选项类似。

appendix/name ★ name = {<前名字>,<后名字>}
name = {<前名字>}

Updated: 2014-03-08

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

用法与普通章节 name 选项类似。

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

表 20 appendix/name 选项的默认设置

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

appendix/number ★ number = {<数字输出命令>}

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

用法与普通章节的 number 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

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

表 21 appendix/number 选项的默认设置

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

我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 21

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

本例的设置效果大致如下:

熟悉 L^AT_EX

第 8 节 实用命令

8.1 字号与间距

`\zihao`

Updated: 2014-03-08

`\zihao {⟨字号⟩}`
用于调整字号大小。其中 ⟨字号⟩ 的有效值共有 16 个，如表 22 所示。使用 `\zihao` 命令调整字体大小时，西文字号大小会始终和中文字号保持一致。

表 22 中文字号

⟨字号⟩	大小 (bp)	大小 (pt)	意义
0	42	42.157 49	初号
-0	36	36.135	小初号
1	26	26.097 49	一号
-1	24	24.09	小一号
2	22	22.082 49	二号
-2	18	18.067 49	小二号
3	16	16.06	三号
-3	15	15.056 24	小三号
4	14	14.052 49	四号
-4	12	12.045	小四号
5	10.5	10.539 37	五号
-5	9	9.033 74	小五号
6	7.5	7.528 12	六号
-6	6.5	6.524 37	小六号
7	5.5	5.520 61	七号
8	5	5.018 74	八号

`\ziju`

Updated: 2014-03-28

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

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

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

8.2 中文数字转换

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

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

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

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

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

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

以中文格式输出数字。这里的数字可以是整数、小数和分数。

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

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

将阿拉伯数字转换为中文数字串。

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

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

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

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

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

8.3 杂项

 $\backslash\text{CTeX}$

用于显示 $\text{C}\text{T}\text{E}\text{X}$ 标志。

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

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

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

9.1 Lua \TeX 下替代字体的设置

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

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

CharRange <hr/> New: 2014-04-14	<pre> \setCJKfamilyfont {<family>} [CharRange = {<character range>} , <alternate font features>] {<alternate font name>} </pre>
---	---

只设置字体族 $\langle family \rangle$ 在字符范围 $\langle character\ range \rangle$ 内,对应字形的替代字体。

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

例 22

```

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

```

declarecharrange ★ <hr/> New: 2014-04-14	<pre> \ctexset { declarecharrange = { {<name₁>} {<character range₁>} , {<name₂>} {<character range₂>} , ... } } </pre>
--	--

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

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

```

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

```

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

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

New: 2014-04-15

清除与重置 CJK 字体族 $\langle family \rangle$ 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_TE_X 宏集的开始处, 定义过宏包选项之后, `\ProcessKeysOptions` 命令之前生效。最好只使用此配置文件修改宏包默认选项。

10.2 宏包载入后的配置

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

例 24

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

例 25

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

11.1 C_TE_X 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

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

hyperref

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

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

fntef

旧版本的 `fntef` 选项用于统一 `CCTfntef` 与 `CJKfntef` 的界面, 新版本 `CTEX` 宏集不再支持 `CCT`, 而是直接载入 `CJKfntef` 或 `xeCJKfntef` 宏包并禁用其彩色设置。该选项是过时选项。

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

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

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

在新版本中, 上述由 `fntef` 衍生的相关宏都成为过时命令。

\CTEXsetfont

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

\CTEXindent

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

\CTEXnoindent

设置 `\parindent=0pt`。过时命令。

\CTEXsetup

`\CTEXsetup[⟨选项⟩]{⟨标题⟩}`

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

\CTEXoptions

`\CTEXoptions[⟨选项⟩]`

相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

\Chinese

`\Chinese{⟨counter⟩}`

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

captiondelimter

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

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

例 26

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

11.3 CT_EX 1.02c 以后的 SVN 开发版

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

`CTEX` 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 C_TE_X 2.2 之前的版本

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

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

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

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

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

11.5 C_TE_X 2.4.1 和 2.4.2

这两个选项已经被删除, 相应功能由新的选项 fixskip 提供。

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

part/fixbeforeskip
chapter/fixbeforeskip

第 12 节 宏集依赖情况与手工安装方法

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

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

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 宏集。
- indentfirst 宏包, 属于 tools 宏集。
- everyisel 宏包, 属于 ms 宏集。
- ulem 宏包。
- zhnumber 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- CJK 宏集。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包, 它还依赖 oberdiek 宏集。
- ➡ 以上是使用 pdfL_AT_EX 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项, 其中 zhm-CJK 是可选的。
- xeCJK 宏包, 它还依赖
 - fontspec 宏包, 它还依赖
 - * euenc 宏包。
 - * xunicode 宏包, 它还依赖
 - graphics 宏集。

- graphics-cfg 宏包。
- graphics-def 宏包。
- environ 宏包, 它还依赖 trimspaces 宏包。
- ➡ 以上是使用 Xe_LA_TE_X 编译时的依赖项。
- LuaTeX-j_a 宏集, 它还依赖
 - adobemapping 宏包。
 - fontspec 宏包。
 - lualibs 宏包。
 - luaotfload 宏包。
 - luatexbase 宏包, 它还依赖 ctablestack 宏包。
 - oberdiek 宏集。
 - xkeyval 宏包。
- ➡ 以上是使用 Lua_AT_EX 编译时的依赖项。
- pxevery_{sel} 宏包, 属于 platex-tools 宏集。
- zhmetrics-uptex 宏包。
- ➡ 以上是使用 up_LA_TE_X 编译时的依赖项。

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

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

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

若要手工安装, 请遵循如下步骤:

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

第 13 节 开发人员

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

目前比较活跃的开发维护人员是刘海洋、李清和黄晨成。

参考文献

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

第 14 节 代码实现

1 <@@=ctex>

宏包载入检查。

2 <*class|ctex>

3 \tl_const:Nx \c__ctex_version_tl

4 { \cs_if_exist_use:cF { ver@ \currname . \@currentext } { 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 } { 2017/07/19 } { }

38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }

39 <*class>

40 \@ifpackagelater { l3keys2e } { 2015/12/20 } { }

41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }

42 </class>

\c__ctex_engine_str
\c__ctex_engine_file_str

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

43 \str_const:Nx \c__ctex_engine_str

44 { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }

45 \msg_new:nnnn { ctex } { engine-not-supported }

46 { Engine~\#1'~is~not~yet~supported,~ctex~will~abort! }

47 { You~can~switch~to~xelatex,~lualatex,~pdflatex,~uplatex,~or~aplatex. }

48 \file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }

49 {

50 \str_const:Nx \c__ctex_engine_file_str

51 { ctex-engine- \c__ctex_engine_str .def }

52 }

53 { \msg_critical:nnx { ctex } { engine-not-supported } { \c__ctex_engine_str } }

54 </class|ctex>

55 <*class|ctex|ctexheading>

```

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

```

14.1 内部函数与变量

```

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

```

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

```

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

```

`\ctex_file_input:n` 输入文件。

```

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

```

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

```

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

```

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

```

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

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

```

对旧版本的宏包给出错误信息。

```

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

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

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

```

`\ctex_if_preamble:TF` 测试是否在 $\text{\LaTeX}2_{\epsilon}$ 的导言区。在宏包内部初始为真,文档最开始位置再设置为假。注意,钩子 `\ctex_after_end_preamble:n` 在 `\AtBeginDocument` 之后执行,可以与 `\@onlypreamble` 的行为一致。

```

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

```

`\ctex_set_default_ccwd:Nn` 若参数 `#2` 带长度单位,则设置它为 `tl` 变量 `#1` 的值,否则以 `\ccwd` 为单位。

```

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

```

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

```

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

```

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

```

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

```

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

```

131 \bool_new:N \g__ctex_zhmCJK_bool

```

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

```

132 \tl_new:N \l__ctex_autoindent_tl

```

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

```

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

```

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

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

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

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

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

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

对无效选项给出警告。

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

对过时选项或命令给出警告。

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

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

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

14.2 宏包选项

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

```

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

```

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

```

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

```

autoindent 自动调整段落的首行缩进功能。

```

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

```

indent 仅为兼容性保留, 已过时。

```

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

```

```

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

```

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

```

226     fontset .tl_gset:N = \g__ctex_fontset_tl ,

```

```

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

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

```

punct 设置标点符号输出格式。

```

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

```

```

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

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

heading 305   heading .bool_set:N = \l__ctex_heading_bool ,

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

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

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

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

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

```

fntef 这些都是过时的宏包兼容选项,原选项功能总是打开的。
 fancyhdr
 hyperref 342 fntef .code:n =

```

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

365 </class>ctex>
366 <*class>ctex|ctexsize>

```

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

```

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

```

将未知选项传给标准文档类。

```

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

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

```

```

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

```

五号字使用标准文档类的 10pt 字体大小设置,小四号字则使用 12pt。

```

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

```

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

```

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

```

14.3 特定引擎支持与设置

14.3.1 ctex-engine-pdfTeX.def

\ctex_set_zhmap:n 设置 upTeX 字体映射,同时作用于 \AtBeginDvi 与 \AtBeginShipoutFirst。该宏对 pdfTeX 和 upTeX 均有用。

```

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

```

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

```

440 \seq_new:N \c__ctex_cmap_encoding_seq
441 \seq_set_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 \pdfTeX{pdfobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
468   \cs_new_protected_nopar:Npx #1
469   {
470     \exp_not:N \pdfTeX{pdffontattr:D \exp_not:N \tex_font:D
471     { /ToUnicode ~ \int_use:N \pdfTeX{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

```

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

```

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 < \c_two_hundred_fifty_six
562     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
563     {
564       \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
565       \tl_gset:Nx #1
566       {
567         \int_to_Hex:n
568         { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
569         \int_to_Hex:n
570         { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "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:x:nnF { \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%
  \CJK@makeActive%
}

```

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

```

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

```

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

```

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

```

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

```

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 \ifpackageafter { luatexja }
677 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
678 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
679 \RequirePackage { luatexja }
680 \ifpackageafter { luatexja } { 2015/09/21 } { }
681 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
682 \RequirePackage { fontspec }
683 \ifpackageafter { fontspec } { 2014/05/25 } { }
684 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

14.3.3.1 LuaTeX-ja 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, 我们把它们归入字符范围 1, 改成 ALchar。

```

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

```

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

```

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

```

以下设置抄录自 `ltj-latex.sty`。

```

704 \directlua{

```

```

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

```

14.3.3.2 LuaTeX-jā 的补丁

```
715 <@@-ctex_ltj>
```

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

```

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

```

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

```

724 \cs_set_protected:Npn \fontspec_visible_space:
725 {
726   \etex_iffontchar:D \tex_font:D "2423 \exp_stop_f:
727   \ltjalchar "2423 \exp_stop_f:
728   \else:
729     \fontspec_visible_space_fallback:
730   \fi:
731 }

```

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

```

732 \ctex_at_end_package:nn { listings }
733 {
734   \use:x
735   {
736     \exp_not:N \RequirePackage { lltpj-listings }
737     \tl_set:Nn \exp_not:N \lstlistingname
738     { \exp_not:o { \lstlistingname } }
739     \tl_set:Nn \exp_not:N \lstlistlistingname
740     { \exp_not:o { \lstlistlistingname } }
741   }
742 }

```

14.3.3.3 字体切换方式

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

```

743 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
744 {
745   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
746   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
747 }
748 \tl_new:N \CJK@family
749 \tl_new:N \l__ctex_ltj_current_font_tl
750 \tl_set:Nn \l__ctex_ltj_current_font_tl
751 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

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

```

752 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
753 {
754   \group_begin:
755     \tl_set_eq:NN \f@encoding \CJK@encoding
756     \tl_set_eq:NN \f@family \CJK@family
757     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
758     \ctex_ltj_pickup_font:
759   \group_end:
760   \font@name
761   \__ctex_ltj_pop_fontname:

```

当字形未定义的时候,NFSS 就会启动替换机制(`\wrong@fontshape`)。第一次启动后,`\l__ctex_ltj_current_font_tl` 还是没有定义。为此,我们再次选择字体,确保它有定义和指向正确的 `font.id`。这对 `AlternateFont` 的设置特别重要。

```

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

```

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

```

772 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
773 {
774   \exp_after:wN \cs_if_exist:NF \font@name
775   {
776     \group_begin:
777       \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
778       \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
779       \define@newfont
780     \group_end:
781   }
782 }
783 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

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

```

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

```

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

```

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

```

784 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
785 {
786   \get@external@font

```

```

787 \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
788 {
789     \tl_set:Nx \external@font
790     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
791 }
792 \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。

```

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

```

`\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。

```

798 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
799 {
800     \ctex_ltj_if_alterate_shape_exist:nF { \curr@fontshape }
801     {
802         \group_begin:
803         \tl_set_eq:NN \CJK@family \f@family
804         \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
805         {
806             \cs_gset_protected_nopar:Npx \subst@correction
807             {
808                 \cs_new_eq:NN
809                 \exp_not:c { \l__ctex_ltj_current_font_tl }
810                 \font@name
811             }
812             \group_insert_after:N \group_insert_after:N
813             \group_insert_after:N \subst@correction
814         }
815         \group_end:
816     }
817 }

```

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

```

818 \prg_new_conditional:Npnn \ctex_ltj_if_alterate_shape_exist:n #1 { T , F , TF }
819 {
820     \lua_now_x:n { luatexja.jfont.does_alt_set (' \lua_escape_x:n {#1}') }
821     \prg_return_true: \else: \prg_return_false: \fi:
822 }

```

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

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

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

```
823 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
824 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }
```

在 `\selectfont` 中更新替代字体。

```
825 \cs_new_protected_nopar:Npn \ctex_ltj_select_alterate_font:
826 {
827   \ctex_ltj_if_alterate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
828   {
829     \lua_now_x:n
830     {
831       luatexja.jfont.output_alt_font_cmd
832       ('y', '\lua_escape_x:n { \l__ctex_ltj_current_shape_tl }')
833     }
834     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a ('f@size') }
835   }
836 }
837 \tl_new:N \l__ctex_ltj_current_shape_tl
838 \tl_set:Nn \l__ctex_ltj_current_shape_tl
839 { \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。

```
840 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
841 {
842   \cs_if_exist:cF { #1/\f@size }
843   {
844     \group_begin:
845     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
846     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
847     \ctex_ltj_pickup_font:
848     \group_end:
849     \__ctex_ltj_pop_fontname:
850   }
851 }
852 <@@=)
```

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。

```
853 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
854 {
855   \ltj@@getjfontnumber #1
856   \lua_now_x:n
857   {
858     luatexja.jfont.pickup_alt_font_b
859     ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
860   }
861 }
```

14.3.3.4 数学字体族

```
862 <@@=-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:

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

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

```

863 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
864 {
865   \lua_now_x:n
866   { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[~/]+' ) ) }
867   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
868 }

```

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

```

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

```

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

```

877 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
878 {
879   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
880   { \ctex_ltj_get_and_define_fonts_ja:nN }
881   { \ctex_ltj_get_and_define_fonts_al:nN }
882   {#1} #2
883 }
884 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
885 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
886 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
887 {
888   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
889   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
890   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
891   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
892   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
893   \ctex_ltj_pickup_font:
894   \tl_put_right:Nx \math@fonts
895   {
896     \ltj@setpar@global
897     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
898     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
899     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
900   }
901 }
902 <@@=ctex_ltj>

```

$\backslash use@mathgroup$
 $\backslash ctex_ltj_use_math_group:Nn$ 在使用 **unicode-math** 宏包时, $\backslash ctex_ltj_math_group_hook$: 将被重定义。

```

903 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
904 {
905   \mode_if_math:T
906   {
907     \math@bgroup
908     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
909     \ctex_ltj_math_group_hook:
910     \ctex_ltj_if_jfont_math:NTF #1
911     { \jfam } { \mathgroup } #2 \scan_stop:
912     \math@egroup
913   }
914 }

```

```

915 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
916 \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。

```

917 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
918 {
919   \group_begin:
920   \cs_set_protected:Npn #1 ##1##2##3
921     { \ltjsetmathletter { ##1 } }
922   #2
923   \group_end:
924 }
925 \ctex_at_end_package:nn { unicode-math }
926 {
927   \cs_if_exist:NTF \um_input_math_symbol_table:
928   {
929     \ctex_ltj_set_math_letter:NN
930     \um_sym:nnn
931     \um_input_math_symbol_table:
932   }
933   {
934     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
935     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
936       { \__um_switchto_literal: }
937     \ctex_ltj_set_math_letter:NN
938     \__um_sym:nnn
939     \__um_input_math_symbol_table:
940   }
941 }

```

14.3.3.5 字体族的定义与使用

```

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

```

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

```

942 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
943 {
944   \str_if_eq:nnTF {#1} { plain }
945   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
946   { \tl_set:Nn \l__ctex_ltj_jfm_tl { #1 } }
947 }
948 \tl_new:N \l__ctex_ltj_jfm_tl
949 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
950 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

```

\CJK@encoding
\__ctex_ltj_change_encoding:

```

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

```

951 \tl_const:Nn \CJK@encoding { LTJY3 }
952 \DeclareFontEncoding { \CJK@encoding } { } { }
953 \use:x
954 {
955   \exp_not:N \DeclareFontSubstitution
956   { \CJK@encoding } { song } { \mddefault } { \updefault }
957 }
958 \lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
959 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
960 { \tl_set_eq:NN \g_fonts_spec_encoding_tl \CJK@encoding }
961 \DeclareFontFamily { \CJK@encoding } { song } { }
962 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }

```

```

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

```

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

973 \newfontfeature { CID } { cid = #1 }
974 \newfontfeature { JFM } { jfm = #1 }
975 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

976 \keys_define:nn { fontspec-preparse-external }
977 {
978   NoEmbed .code:n =
979     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
980 }
981 \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。

```

982 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
983 {
984   \group_begin:
985   \clist_clear:N \l__ctex_ltj_char_range_clist
986   \seq_clear:N \l__ctex_ltj_alternate_seq
987   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
988   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
989   \clist_set:Nn \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
990   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
991   {
992     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
993     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
994       {#1} { \l__ctex_ltj_font_options_clist }
995     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
996     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
997     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
998   }
999   \group_end:
1000 }
1001 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1002 \clist_new:N \l__ctex_ltj_font_options_clist
1003 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1004 {
1005   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1006   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1007 }

```

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

```

\g__ctex_ltj_family_font_name_prop
\g__ctex_ltj_family_font_options_prop
1008 \prop_new:N \g__ctex_ltj_family_name_prop
1009 \prop_new:N \g__ctex_ltj_family_font_name_prop
1010 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

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

```

1011 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1012 {
1013   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1014   {

```

```

1015 \cs_undefine:c { \l__ctex_ltj_family_csname:n {#1} }
1016 \cs_undefine:c { \l__ctex_ltj_alternate_cs:n {#1} }
1017 \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1018 {
1019 \use:c { \l__ctex_ltj_alternate_cs:n { clear / #1 } }
1020 \cs_undefine:c { \l__ctex_ltj_alternate_cs:n { clear / #1 } }
1021 \cs_undefine:c { \l__ctex_ltj_alternate_cs:n { reset / #1 } }
1022 \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1023 }
1024 \msg_warning:nxxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1025 }
1026 }
1027 \tl_new:N \l__ctex_ltj_tmp_tl
1028 \msg_new:nnn { ctex } { redefine-family }
1029 { Redefining~CJKfamily~\l__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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

```

1030 \cs_new_protected_nopar:Npn \l__ctex_ltj_gset_family_cs:nn #1#2
1031 {
1032 \cs_gset_protected_nopar:cpx { \l__ctex_ltj_family_csname:n {#1} }
1033 {
1034 \group_begin:
1035 \l__ctex_ltj_change_encoding:
1036 \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1037 \exp_not:n { \cs_set_eq:NN \CTEX@fontfamily \use_none:n }
1038 \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1039 { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1040 \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1041 { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1042 \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1043 \exp_not:N \g__ctex_ltj_fontspec_family_tl
1044 \l__ctex_ltj_set_alternate_family:n {#1}
1045 \group_end:
1046 }
1047 }
1048 \tl_new:N \l__ctex_ltj_base_family_tl
1049 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1050 \cs_new_nopar:Npn \l__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1051 \cs_new_protected_nopar:Npn \l__ctex_ltj_set_alternate_family:n #1
1052 {
1053 \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1054 \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1055 \cs_if_exist_use:c { \l__ctex_ltj_alternate_cs:n { reset / #1 } }
1056 \cs_if_exist_use:c { \l__ctex_ltj_alternate_cs:n {#1} }
1057 }
1058 \cs_new:Npn \l__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

\CJKfamily 切换字体。

```

1059 \NewDocumentCommand \CJKfamily { m }
1060 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1061 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1062 {
1063 \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1064 {
1065 \tl_set:Nn \l__ctex_ltj_family_tl {#1}
1066 \selectfont
1067 }
1068 { \l__ctex_ltj_family_unknown_warning:n {#1} }
1069 }
1070 \tl_new:N \l__ctex_ltj_family_tl
1071 \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。
1072 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
1073 {
1074   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1075   { \prg_return_true: }
1076   {
1077     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1078     {
1079       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1080       \prg_return_true:
1081     }
1082     { \prg_return_false: }
1083   }
1084 }
1085 \cs_generate_variant:Nn \ctex_ltj_family_if_exist:nNT { x }
1086 \cs_generate_variant:Nn \ctex_ltj_family_if_exist:nNF { x }
1087 \cs_generate_variant:Nn \ctex_ltj_family_if_exist:nNTF { x }

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

\ctex_ltj_fontspec:nn 1126 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1127 {
1128   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1129   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1130   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1131   {
1132     \int_gincr:N \g__ctex_ltj_family_int
1133     \__ctex_ltj_fontspec:xnn
1134     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1135     {#1} {#2}

```

```

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

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

```

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

```

1187 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1188 {
1189     \IfNoValueTF {#2}
1190     { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1191     {
1192         \use:x { #1 {#2} {#3} }
1193         #4
1194     }
1195 }
1196 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1197 {
1198     \use:x { #1 {#4} {#2} }

```

```
1199     #3
1200 }
```

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

\setCJKmainfont 1229 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1230 {
\setCJKmonofont 1231   \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1232   { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1233   { \normalfont }
1234 }
1235 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1236 \NewDocumentCommand \setCJKsansfont { o m }
1237 {
1238   \__ctex_ltj_pass_args:nnnn
1239   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1240   { \normalfont }
1241 }
1242 \NewDocumentCommand \setCJKmonofont { o m }
1243 {
1244   \__ctex_ltj_pass_args:nnnn
1245   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1246   { \normalfont }
1247 }
1248 \NewDocumentCommand \setCJKmathfont { o m }
1249 {
1250   \__ctex_ltj_pass_args:nnnn
1251   { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1252   { }
1253 }
1254 \NewDocumentCommand \defaultCJKfontfeatures { m }
1255 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1256 \clist_new:N \g__ctex_ltj_default_features_clist
1257 \@onlypreamble \setCJKmainfont
1258 \@onlypreamble \setCJKsansfont
1259 \@onlypreamble \setCJKmonofont
1260 \@onlypreamble \setCJKmathfont
1261 \@onlypreamble \setCJKromanfont
1262 \@onlypreamble \defaultCJKfontfeatures
```

```

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

```

\fontfamily
\CTEX@fontfamily

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

```

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

```

在导言区末尾更新 \CJKfamilydefault。

```

1289 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

\ctex_ltj_ensure_default_family:

在导言区结束确认 \CJKfamilydefault 确实存在。

```

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

```

\ctex_ltj_update_mathfont:

更新数学字体为实际的字体。

```

1315 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1316 {
1317   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl

```

```

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

```

14.3.3.6 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

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

```

\ctex_ltj_set_alternate_seq:n

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

```

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

```

\ctex_ltj_set_alternate_family:nnF

如果在字体的选项中设置了 CharRange,则只设置替代字体。

```

1370 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1371 {

```

```

1372 \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1373 {
1374   \__ctex_ltj_check_family:n {#1}
1375   \seq_if_empty:NF \l__ctex_ltj_altername_seq
1376   { \ctex_ltj_save_altername_seq:cn { \__ctex_ltj_altername_cs:n {#1} } {#2} }
1377   #3
1378 }
1379 { \ctex_ltj_set_altername_family:nn {#1} {#2} }
1380 }

```

保存由 AlternateFont 设置的替代字体序列。

```

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

```

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

```

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

```

保存替代字体序列的定义,以备定义主字体时使用。

```

1424 \cs_new_protected_nopar:Npn \ctex_ltj_save_altername_family:Nnnn #1#2#3#4
1425 {
1426   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1427   \cs_gset_protected_nopar:Npx #1
1428   { \exp_not:o { #1 \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} } }

```

```
1429 }
```

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

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

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

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

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

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

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

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

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

```
1463 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1464 {
1465   \clist_map_inline:Nn #1
1466   {
1467     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1468     {
1469       \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1470       \l__ctex_ltj_char_range_tl
1471     }
1472     { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1473   }
1474   \__ctex_ltj_save_alternate_shape:cnn
```

```

1475 { \__ctex_ltj_alterate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1476 { luatexja.jfont.clear_alt_font_latex }
1477 { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1478 }

```

\ctex_ltj_set_alterate_shape:nnn

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

```

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

```

\ctex_ltj_set_alterate_shape:nnN

若字符范围预先由 declarecharrange 声明,则可以直接使用。

```

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

```

__ctex_ltj_save_alterate_shape:Nnn

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

```

1515 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alterate_shape:Nnn #1#2#3
1516 {
1517   \group_begin:
1518   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1519   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1520   \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1521   \cs_gset_protected_nopar:Npx #1
1522   { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1523   \group_end:
1524 }
1525 \cs_generate_variant:Nn \__ctex_ltj_save_alterate_shape:Nnn { c }

```

clearalternatefont
resetalternatefont

清除和重置操作总是全局的。

```

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

```

declarecharrange

预先声明字符范围。

```

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

```

\ctex_ltj_declare_char_range:nn
\g__ctex_ltj_char_range_prop

#1 是名字, #2 是范围。

```

1574 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1575 {
1576   \tl_clear:N \l__ctex_ltj_char_range_tl
1577   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1578   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1579   \ctex_ltj_def_char_range_key:n {#1}
1580   \tl_clear:N \l__ctex_ltj_char_range_tl
1581 }
1582 \tl_new:N \l__ctex_ltj_char_range_tl
1583 \prop_new:N \g__ctex_ltj_char_range_prop

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

14.3.3.7 其它设置

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

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

1633 \AtBeginDocument
1634 {
1635   \ctex_appto_cmd:NnnTF \verbatim@font
1636   { \char_set_catcode_letter:n { 64 } }
1637   { \CTEX@verbatim@font@hook }
1638   { }
1639   { \ctex_patch_failure:N \verbatim@font }
1640 }
1641 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1642 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1643 <@@= >
```

\@@italiccorr L^AT_EX 的倾斜校正也要重新定义。

```
1644 \cs_set_eq:NN \@@italiccorr \/
1645 <@@=ctex>
1646 </luatex>
```

14.3.4 ctex-engine-uptex.def

```
1647 <*uptex|aptex>
```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```
1648 \tl_set:Nn \CJKrmdefault { rm }
1649 \tl_set:Nn \CJKsfdefault { sf }
1650 \tl_set:Nn \CJKttdefault { tt }
1651 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1652 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1653 { \kanjifamily { \CJKrmdefault } }
1654 { }
1655 { \ctex_patch_failure:N \rmfamily }
1656 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1657 { \kanjifamily { \CJKsfdefault } }
1658 { }
1659 { \ctex_patch_failure:N \sffamily }
1660 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1661 { \kanjifamily { \CJKttdefault } }
1662 { }
1663 { \ctex_patch_failure:N \ttfamily }
1664 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1665 { \kanjifamily { \CJKfamilydefault } }
1666 { \cs_set_eq:NN \reset@font \normalfont }
1667 { \ctex_patch_failure:N \normalfont }
1668 \NewDocumentCommand \CJKfamily { m }
1669 {
1670   \kanjifamily { #1 }
```

```

1671 \selectfont
1672 }
1673 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```

在导言区末尾更新 \CJKfamilydefault。

```

1674 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

\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。 $\text{up}\text{\LaTeX} 2016/05/07$ 的定义有所变化，这一行为可以由用户通过 **platexrelease** 包改变，需要分支处理。

```

1675 \ctex_patch_cmd_once:NnnTF \em
1676 { \ExplSyntaxOff }
1677 { \eminnershape \else \gtfamily \itshape }
1678 { \eminnershape \else \itshape }
1679 { }
1680 {
1681   \ctex_patch_cmd:Nnn \em
1682     { \mcfamily \upshape \else \gtfamily \itshape }
1683     { \eminnershape \else \itshape }
1684 }
1685 \cs_set_nopar:Npn \eminnershape { \upshape }

```

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

```

1686 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1687 {
1688   \DeclareKanjiFamily{JY2}{#1}{}
1689   \DeclareKanjiFamily{JT2}{#1}{}
1690   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1691   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1692   \tl_if_empty:nF { #3 }
1693   {
1694     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1695     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1696   }
1697 }
1698 \@onlypreamble \ctex_set_upfamily:nnn

```

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

```

1699 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1700 {
1701   \ctex_set_zhmap:n
1702   {
1703     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1704     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1705     \tl_if_empty:nF { #3 }
1706     {
1707       \special{ pdf:mapline~ #1b-h~  UniGB-UTF16-H~ #3 }
1708       \special{ pdf:mapline~ #1b-v~  UniGB-UTF16-V~ #3 }
1709     }
1710   }
1711 }
1712 \@onlypreamble \ctex_set_upmap:nnn

```

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

```

1713 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1714 {

```

```

1715 \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1716 \ctex_set_upmap:nnn { upserifit } { #3 } {}
1717 \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1718 \ctex_set_upmap:nnn { upmono } { #6 } {}
1719 }
1720 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

`everyysel` 宏包 (2011/10/28) 未考虑 `upLATEX` 对 `\selectfont` 的修改, 需要引入 `pxe-`
`everyysel` 宏包。

```

1721 \RequirePackage { pxeveryysel }
1722 </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。

```

1723 <*pdf|luatex|uptex|aptex>
1724 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1725 {
1726   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1727   {
1728     \group_begin:
1729     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1730     \tl_gset:Nx \CJKfamilydefault
1731     {
1732       \str_case:onF { \familydefault }
1733       {
1734         { \rmdefault } { \exp_not:N \CJKrmdefault }
1735         { \sfdefault } { \exp_not:N \CJKsfdefault }
1736         { \ttdefault } { \exp_not:N \CJKttdefault }
1737       }
1738       { \CJKfamilydefault }
1739     }
1740     \group_end:
1741   }

```

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

```

1742 <*luatex>
1743 \ctex_ltj_ensure_default_family:
1744 </luatex>
1745 }

```

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

```

1746 \tl_new:N \l__ctex_family_default_init_tl
1747 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1748 \tl_set:Nx \l__ctex_family_default_init_tl
1749 {
1750   \exp_not:N \__ctex_family_default_wrap:n
1751   { \exp_not:o { \CJKfamilydefault } }
1752 }
1753 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1754 </pdf|luatex|uptex|aptex>

```

14.3.6 操作系统的判断

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

```

1755 <*luatex>

```

```

1756 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1757 {
1758   \tl_gset:Nx \g__ctex_fontset_tl
1759   {
1760     \lua_now_x:n
1761     {
1762       if ~ os.name == 'windows' then ~
1763         tex.sprint ( 'windows' )
1764       elseif ~ os.name == 'macosx' then ~
1765         tex.sprint ( 'mac' )
1766       else ~
1767         tex.sprint ( 'fandol' )
1768       end
1769     }
1770   }
1771 }
1772 </luatex>

```

ApTeX 可以使用 \ngostype 来判断。

```

1773 <*aptex>
1774 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1775 {
1776   \tl_gset:Nx \g__ctex_fontset_tl
1777   {
1778     \str_case:onF { \ngostype }
1779     {
1780       { Win32 } { windows }
1781       { Win64 } { windows }
1782       { Darwin } { mac }
1783     }
1784     { fandol }
1785   }
1786 }
1787 </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!

```

1788 <*xetex|pdfTeX|uptex>
1789 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1790 {
1791   \file_if_exist:nTF { /dev/null }
1792   {
1793     \file_if_exist:nTF { nul: }
1794     {
1795       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1796       { \ctex_if_macosx:TF { mac } { windows } }
1797       { \ctex_if_macosx:TF { mac } { fandol } }
1798     }
1799     { \ctex_if_macosx:TF { mac } { fandol } }
1800   }
1801   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1802 }
1803 \str_const:Nx \c__ctex_upper_case_file_str
1804 { \exp_args:No \str_upper_case:n { \g_file_current_name_tl } }

```

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

```

1805 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1806 {
1807   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1808   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1809   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1810 }
1811 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }

```

```
1812 </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。

```
1813 <*class|ctex>
1814 \ifpackageloaded { hyperref }
1815 {
1816   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1817     { \hypersetup {#1} }
1818 }
1819 {
1820   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1821     { \PassOptionsToPackage {#1} { hyperref } }
1822 }
1823 </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。

```
1824 <*pdfTeX>
1825 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1826 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1827 {
1828   \ctex_hypersetup:n { CJKbookmarks = true }
1829   \sys_if_output_pdf:TF
1830     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1831     {
1832       \ctex_at_end_package:nn { hyperref }
1833       {
1834         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1835         {
1836           \AtBeginShipoutFirst
1837             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1838         }
1839         { \RequirePackage { xCJK2uni } }
1840       }
1841     }
1842 }
1843 { \ctex_hypersetup:n { pdfencoding = unicode } }
1844 </pdfTeX>
```

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

```
\ifHy@unicode
  \HyPsd@ConvertToUnicode#1%
  \ifx\HyPsd@pdfencoding\HyPsd@pdfencoding@auto
    \ltx@ifundefined{StringEncodingConvertTest}{%
      }{%
        \EdefUnescapeString\HyPsd@temp#1%
      }
    \ifxetex
```

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

```

\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@unicodedefalse
}%
\HyPsd@ToBigChars#1%
}%

```

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

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

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

```

1845 <*xetex| luatex>
1846 \ctex_hypersetup:n { pdfencoding = unicode }
1847 </xetex| luatex>

```

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

```

1848 <*uptex| aptex>
1849 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1850 \ctex_at_end_package:nn { hyperref }
1851 { \AtBeginShipoutFirst { \special { pdf:tounicode~ UTF8-UTF16 } } }
1852 </uptex| aptex>
1853 <*pdfTeX|xetex| luatex| uptex| aptex>

```

14.3.8 CJKfntef、xeCJKfntef 相关设置

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

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

```

1854 <*pdfTeX>
1855 \RequirePackage { CJKfntef }
1856 \normalem
1857 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1858 { \tl_clear:c { CJK#1color } }
1859 </pdfTeX>
1860 <*xetex>
1861 \RequirePackage { xeCJKfntef }
1862 \ifpackageclater { xeCJKfntef } { 2014/11/04 }
1863 {
1864   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1865   { \xeCJKsetup { #1 / format = { } } }
1866 }
1867 {

```

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

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

```

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

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1918 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1919 <*pdfTEX|xetex>
1920 {
1921 \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1922 \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1923 }
1924 </pdfTEX|xetex>
1925 <*luatex>
1926 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1927 </luatex>
1928 <*uptex|aptex>
1929 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1930 </uptex|aptex>

```

```
1931 \dim_new:N \ccwd
```

\ctex_update_ccglue: 更新字间距。

```
1932 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1933 <pdfTeX>
1934 {
1935     \cs_set_protected_nopar:Npn \CJKglue
1936     { \skip_horizontal:N \l__ctex_ccglue_skip }
1937 }
1938 </pdfTeX>
1939 <xetex>
1940 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1941 </xetex>
1942 <luatex>
1943 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1944 </luatex>
1945 <uptex|aptex>
1946 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1947 </uptex|aptex>
1948 \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

```
1949 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1950 {
1951 <pdfTeX|xetex>
1952     \if_meaning:w \CJKglue \__ctex_ccglue:
1953     \prg_return_false: \else: \prg_return_true: \fi:
1954 </pdfTeX|xetex>
1955 <luatex>
1956     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1957     { \prg_return_false: } { \prg_return_true: }
1958 </luatex>
1959 <uptex|aptex>
1960     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1961     { \prg_return_false: } { \prg_return_true: }
1962 </uptex|aptex>
1963 }
```

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

```
1964 <pdfTeX|xetex>
1965 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1966 </pdfTeX|xetex>
```

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

```
1967 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
<pdfTeX|xetex> 1968 { \dim_set:Nn \ccwd { \f@size\p@ } }
<luatex> 1969 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1970 { \dim_set:Nn \ccwd { 1zw } }
```

14.3.10 其它

\ctex_add_to_selectfont:n \EverySelectfont 直到文档开始时才有效。为了 \ccwd 和 LuaTeX-jan 的字体设置在导言区也可用, 我们还需要在这里手工修改 \selectfont。everyysel 宏包会用 \CheckCommand 来检

查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。`pxeveryysel` 宏包取消了检查, 但也需要恢复定义, 避免重复使用钩子。

```

1971 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1972 {
1973   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1974   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1975 }
1976 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1977 \if_cs_exist:N \@EverySelectfont@Init
1978   \group_begin:
1979   \cs_set:Npn \__ctex_tmp:N #1
1980   {
1981     \tl_set:Nn \l__ctex_tmp_tl {#1}
1982     \cs_new_eq:NN \CTEX@selectfont@save #1
1983     \cs_new_protected_nopar:Npn \__ctex_restore_selectfont:
1984     {
1985       \tl_put_left:Nn \@EverySelectfont@Init
1986       { \let #1 \CTEX@selectfont@save }
1987       \cs_undefine:N \__ctex_restore_selectfont:
1988     }
1989   }
1990   \ctex_parse_name:NN \__ctex_tmp:N \selectfont
1991   \exp_last_unbraced:NNo \group_end:
1992   \ctex_preto_cmd:NnnTF { \l__ctex_tmp_tl } { \ExplSyntaxOff }
1993   { \CTEX@selectfont@hook }
1994   { \__ctex_restore_selectfont: }
1995   { \ctex_patch_failure:N \selectfont }
1996 \fi:

```

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

```

1997 <*/pdfTeX>
1998 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1999 </pdfTeX>
2000 <*/xetex|luatex|uptex|aptex>
2001 \EverySelectfont { \CTEX@selectfont@hook }
2002 </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。

```

2003 <*/luatex>
2004 \ctex_add_to_selectfont:n
2005 {
2006   \ctex_ltj_select_font:
2007   \ctex_ltj_select_alternate_font:
2008 }
2009 \tl_set:Nn \CJK@family { song } \selectfont
2010 \tl_clear:N \CJK@family
2011 </luatex>

```

`\ctex_update_xkanjiskip:` `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。

```

2012 <*/luatex|uptex|aptex>
2013 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
2014 {
2015   \skip_if_eq:nnT
2016   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
2017   { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2018   {

```

```

2019         \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
<luatex> 2020         \ltjsetxkanjiskip \l__ctex_xkanjiskip_skip
<uptex|aptex> 2021         \skip_set_eq:NN \ptex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2022     }
2023 }
2024 \tl_new:N \l__ctex_xkanjiskip_tl
2025 \tl_set:Nn \l__ctex_xkanjiskip_tl
<luatex> 2026 { .25\zw plus 1pt minus 1pt }
<uptex|aptex> 2027 { .25zw plus 1pt minus 1pt }
2028 \skip_new:N \l__ctex_xkanjiskip_skip
2029 \skip_set:Nn \l__ctex_xkanjiskip_skip
<luatex> 2030 { \ltjgetparameter { xkanjiskip } }
<uptex|aptex> 2031 { \ptex_xkanjiskip:D }

2032 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2033 </luatex|uptex|aptex>

```

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

```

\cdp
\cwg
2034 <*luatex>
2035 \dim_new:N \cht
2036 \dim_new:N \cdp
2037 \dim_new:N \cwg
2038 \newluafunction \g__ctex_kanjisize_func
2039 \group_begin:
2040 \char_set_catcode_space:n { 32 }
2041 \lua_now_x:n
2042 {
2043     local nulltable = { }
2044     local t = lua.get_functions_table()
2045     local fmt = luatexja.jfont.font_metric_table
2046     local getattribute = tex.getattribute
2047     local setdimen = tex.setdimen
2048     t[\int_use:N \g__ctex_kanjisize_func] = function ()
2049         local ft = fmt[getattribute('ltj@curjfont')] or nulltable
2050         local ft = ft and ft.char_type or nulltable
2051         local fk = ft and ft[0] or nulltable
2052         setdimen('cht', fk.height or 0)
2053         setdimen('cdp', fk.depth or 0)
2054         setdimen('cwg', fk.width or ft.zw or 0)
2055     end
2056 }
2057 \group_end:
2058 \cs_new_protected_nopar:Npn \ctex_update_kanjisize:
2059 { \luafunction:D \g__ctex_kanjisize_func }
2060 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2061 </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。

```

2062 \keys_define:nn { ctex }
2063 {
2064     <*pdf|xtex>
2065     space .choice: ,
2066     space / true .code:n =
<pdf|xtex> 2067     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<xtex> 2068     { \xeCJKsetup { CJKspace = true } } ,
2069     space / auto .code:n =
<pdf|xtex> 2070     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
<xtex> 2071     { \xeCJKsetup { CJKspace = false } } ,
2072     space / false .code:n =
<pdf|xtex> 2073     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
<xtex> 2074     { \xeCJKsetup { CJKspace = false } } ,
2075     space .default:n = { true } ,
2076     space .initial:n = { auto }
2077 </pdf|xtex>

```

```

2078 <*luatex|uptex|aptex>
2079     space .code:n =
2080     { \msg_warning:nn { ctex } { invalid-option } }
2081 </luatex|uptex|aptex>
2082 }

```

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-ja 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。

```

2083 \keys_define:nn { ctex }
2084 {
2085     punct .code:n =
2086     {
2087         \tl_set:Nx \l__ctex_punct_tl { #1 }
2088         \punctstyle { \l__ctex_punct_tl }
2089         \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2090         \ctex_mono_jfm:o { \l__ctex_punct_tl }
2091         \msg_warning:nn { ctex } { invalid-option }
2092     } ,
2093     punct .default:n = { quanjiao } ,
2094 }

```

X_YLaTeX、LuaLaTeX 和 upLaTeX 总是使用 UTF8 编码。

```

2095 <*xetex|luatex|uptex|aptex>
2096 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2097 </xetex|luatex|uptex|aptex>
2098 </pdfTeX|xetex|luatex|uptex|aptex>
2099 <*class|ctex>

```

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。

```

2100 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.4 用户设置接口

```

\ctexset 2101 <*class|ctex|ctexheading>
2102 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
2103 </class|ctex|ctexheading>

```

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

```

2104 <*class|ctex>
2105 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
2106 {
2107     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
2108     { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
2109     \IfNoValueF {#1} { \keys_set:nn { ctex } / #2 } {#1} }
2110 }
2111 \NewDocumentCommand \CTEXoptions { +o }
2112 {
2113     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
2114     { \ctexset~ {~ #1~ }~ is~ set. }
2115     \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
2116 }
2117 </class|ctex>

```

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。

```

2118 \keys_define:nn { ctex }
2119 {
2120   autoindent .choice: ,
2121   autoindent .default:n = { true } ,
2122   autoindent / true .code:n =
2123   {
2124     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2125     \ctex_select_size:
2126   } ,
2127   autoindent / false .code:n =
2128   { \tl_clear:N \l__ctex_autoindent_tl } ,
2129   autoindent / unknown .code:n =
2130   {
2131     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2132     \ctex_select_size:
2133   }
2134 }

```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2135 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2136 \cs_new_protected_nopar:Npn \ctex_select_size:
2137 { \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。

```

2138 \cs_new_protected_nopar:Npn \ctex_update_size:
2139 {
2140   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2141   {
2142     \ctex_update_stretch:
2143     \ctex_update_parindent:
2144   }
2145   { \ctex_update_ziju: }
2146 }
2147 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2148 \tl_new:N \l__ctex_ziju_tl
2149 \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。

```

2150 \ctex_add_to_selectfont:n
2151 { \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。

```

2152 \keys_define:nn { ctex }
2153 {
2154   linestretch .code:n =
2155   {
2156     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2157     \ctex_select_size:
2158   } ,
2159   linestretch .value_required:n = true
2160 }
2161 \tl_new:N \l__ctex_line_stretch_tl
2162 \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。

```

2163 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2164 {
2165   \ctex_update_em_unit:
2166   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2167   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2168   {
2169     \ctex_if_ccglue_touched:TF
2170     { \ctex_update_ccwd: }
2171     {
2172       \dim_set:Nn \l__ctex_tmp_dim
2173       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2174       \skip_set:Nn \l__ctex_ccglue_skip
2175       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2176       \ctex_update_ccglue:
2177     }
2178   }
2179   {
2180     \int_set:Nn \l__ctex_tmp_int
2181     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2182     \tex_divide:D \l__ctex_tmp_int \ccwd
2183     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2184     {
2185       \skip_set:Nn \l__ctex_ccglue_skip
2186       {
2187         \c_zero_dim plus \dim_eval:n
2188         {
2189           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2190           \l__ctex_tmp_int
2191         }
2192       }
2193     }
2194     { \skip_zero:N \l__ctex_ccglue_skip }
2195     \ctex_update_ccglue:
2196   }
2197 }

```

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

```

2198 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2199 {
2200   \tl_if_empty:NF \l__ctex_autoindent_tl
2201   {
2202     \dim_compare:nNnF \parindent = \c_zero_dim
2203     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2204   }
2205 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2206 \NewDocumentCommand \ziju { m }
2207 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2208 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2209 {
2210   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2211   \ctex_select_size:
2212 }

```

`\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。

```

2213 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2214 {
2215   \ctex_update_em_unit:
2216   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2217   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2218   \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。

```

2219   {
2220     \dim_set:Nn \l__ctex_tmp_dim
2221     { \linewidth - \ccwd + \l__ctex_ziju_dim }
2222     \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2223     \tex_divide:D \l__ctex_tmp_int \ccwd
2224     \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。

```

2225     \dim_compare:nNnF \parindent = \c_zero_dim
2226     {
2227       \int_compare:nNnF \l__ctex_tmp_int < \c_three
2228       { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2229     }
2230     \skip_set:Nn \l__ctex_ccglue_skip
2231     {
2232       \l__ctex_ziju_dim
2233       plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2234       minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2235       { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2236     }
2237   }
2238   { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2239   \ctex_update_ccglue:

```

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

```

2240   \ctex_update_parindent:
2241 }
2242 \dim_new:N \l__ctex_ziju_dim

```

`\CTEXindent`
`\CTEXnoindent`

过时命令。

```

2243 \NewDocumentCommand \CTEXindent { }
2244 {
2245   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2246   { \parindent is~ set~ to~ 2\ccwd. }
2247   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2248 }
2249 \NewDocumentCommand \CTEXnoindent { }
2250 {
2251   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2252   { \parindent is~ set~ to~ 0pt. }
2253   \dim_zero:N \parindent
2254 }

```

14.6 中文数字与日期

```

2255 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2256 \RequirePackage { zhnumber }

```

```

\chinese 2257 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2258 \cs_new_eq:NN \@chinese \@zhnum
2259 \cs_new_eq:NN \Chinese \chinese
2260 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

```

2261 \ctex_at_end_package:nn { enumitem }
2262 {
2263   \cs_if_free:NF \AddEnumerateCounter
2264   {
2265     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2266     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2267     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2268   }
2269 }

\CTEXnumber 2270 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 2271 { \protected@edef #1 { \zhnumber {#2} } }
2272 \NewDocumentCommand \CTEXdigits { m m }
2273 { \protected@edef #1 { \zhdigits {#2} } }

today 2274 \cs_set_eq:NN \CTEX@todayold \today
2275 \keys_define:nn { ctex }
2276 {
2277   today .choice: ,
2278   today / old .code:n =
2279     { \cs_set_eq:NN \today \CTEX@todayold } ,
2280   today / small .code:n =
2281     {
2282       \cs_set_eq:NN \today \zhtoday
2283       \zhnumsetup { time = Arabic }
2284     } ,
2285   today / big .code:n =
2286     {
2287       \cs_set_eq:NN \today \zhtoday
2288       \zhnumsetup { time = Chinese }
2289     } ,
2290   today / unknown .code:n =
2291     { \msg_error:nxx { ctex } { today-undef } {#1} }
2292 }
2293 \msg_new:nnnn { ctex } { today-undef }
2294 { Today~format~`#1'~is~undefined. }
2295 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

14.7 其它中文标题定义

`\proofname` 未在标准文档类中定义, 需要确保它非空。

```

2296 \tl_if_exist:NF \proofname
2297 {
2298   \tl_new:N \proofname
2299   \tl_set:Nn \proofname { Proof }
2300 }

2301 \keys_define:nn { ctex }
2302 {
2303   contentsname .tl_set:N = \contentsname ,
2304   listfigurename .tl_set:N = \listfigurename ,
2305   listtablename .tl_set:N = \listtablename ,
2306   figurename .tl_set:N = \figurename ,
2307   tablename .tl_set:N = \tablename ,
2308   abstractname .tl_set:N = \abstractname ,
2309   indexname .tl_set:N = \indexname ,
2310   appendixname .tl_set:N = \appendixname ,
2311   proofname .tl_set:N = \proofname ,
<article> 2312   bibname .tl_set:N = \refname

```

```

2313 bibname .tl_set:N = \bibname
2314 <*beamer>
2315 algorithmname .tl_set:N = \algorithmname ,
2316 bibname .tl_set:N = \bibname ,
2317 refname .tl_set:N = \refname ,
2318 continuation .tl_set:N = \insertcontinuationtext
2319 </beamer>
2320 }

2321 <*ctex>
2322 \msg_new:nnn { ctex } { ctexbibname }
2323 {
2324   Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.~\
2325   The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2326 }
2327 \tl_if_exist:NTF \insertcontinuationtext
2328 {
2329   \keys_define:nn { ctex }
2330   {
2331     algorithmname .tl_set:N = \algorithmname ,
2332     bibname .tl_set:N = \bibname ,
2333     refname .tl_set:N = \refname ,
2334     continuation .tl_set:N = \insertcontinuationtext
2335   }
2336 }
2337 {
2338   \tl_if_exist:NTF \bibname
2339   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2340   {
2341     \tl_if_exist:NTF \refname
2342     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2343     {
2344       \msg_warning:nn { ctex } { ctexbibname }
2345       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2346     }
2347   }
2348 }
2349 </ctex>

2350 </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。

```
2351 <*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。

```

2352 <*article|book|report>
2353 \seq_new:N \c__ctex_section_headings_seq
2354 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2355 { section , subsection , subsubsection , paragraph , subparagraph }
2356 </article|book|report>

```

```

\c__ctex_headings_seq 2357 \seq_new:N \c__ctex_headings_seq
2358 <*article|book|report>
2359 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2360 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2361 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2362 </article|book|report>
2363 <*beamer>
2364 \seq_gset_from_clist:Nn \c__ctex_headings_seq

```

```

2365 { part , section , subsection }
2366 </beamer>

```

```

__ctex_initial_heading:n 2367 \cs_new_protected_nopar:Npn __ctex_initial_heading:n #1
2368 {
2369   \tl_new:c { CTEX@pre#1 }
2370   \tl_new:c { CTEX@post#1 }
2371   \tl_const:cx { CTEX@the#1 }
2372   {
2373     \exp_not:c { CTEX@pre#1 }
2374     \exp_not:c { CTEX@the#1 }
2375     \exp_not:c { CTEX@post#1 }
2376   }
2377   \tl_const:cx { CTEX@#1name }
2378   {
2379     \group_begin:
2380     \exp_not:c { CTEX@#1@nameformat }
2381     {
2382       \exp_not:c { CTEX@pre#1 }
2383       \exp_not:N \tl_if_empty:NTF
2384       \exp_not:c { CTEX@#1@numberformat }
2385       { \exp_not:c { CTEX@the#1 } }
2386       {
2387         \group_begin:
2388         \exp_not:c { CTEX@#1@numberformat }
2389         \exp_not:c { CTEX@the#1 }
2390         \group_end:
2391       }
2392       \exp_not:c { CTEX@post#1 }
2393     }
2394   \group_end:
2395 }
2396 }

```

```

__ctex_def_heading_keys:n 2397 \cs_new_protected_nopar:Npn __ctex_def_heading_keys:n #1
2398 {
2399   \tl_put_right:Nx \l__ctex_tmp_tl
2400   {
2401     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2402     #1 / name .code:n =
2403     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2404     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2405     #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1@beforekip } ,
2406     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2407     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2408     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2409     #1 / numbering .initial:n = true ,
2410     #1 / beforekip .initial:n = \c_zero_skip ,
2411     #1 / afterskip .initial:n = \c_zero_skip ,
2412     #1 / indent .initial:n = \c_zero_dim ,
2413     #1 / beforekip .value_required:n = true ,
2414     #1 / afterskip .value_required:n = true ,
2415     #1 / indent .value_required:n = true ,
2416     <*article|book|report>
2417     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2418     #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2419     #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2420     #1 / hang .initial:n = true ,
2421     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2422     #1 / tocline .code:n =
2423     {
2424       \cs_set:Npn \exp_not:c { CTEX@#1@tocline }
2425       \exp_not:n { #####1#####2 } { #####1 }
2426     } ,
2427     __ctex_plus_key_aux:nn {#1} { break } ,
2428     </article|book|report>

```

```

2429     \__ctex_plus_key_aux:nn {#1} { format } ,
2430     \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2431     \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2432     \__ctex_plus_key_aux:nn {#1} { titleformat } ,
2433     \__ctex_plus_key_aux:nn {#1} { aftername } ,
2434     \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2435   }
2436 }
2437 \cs_new_nopar:Npn \__ctex_plus_key_aux:nn #1#2
2438 {
2439   #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2440   #1 / #2 + .code:n =
2441     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2442   #1 / #2 ~ + .code:n =
2443     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2444 }

```

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

```

2445 \NewDocumentCommand \ctex_assign_heading_name:nn
2446 { m > { \SplitArgument { \c_one } { , } } +m }
2447 { \__ctex_assign_heading_name:nnn {#1} #2 }
2448 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2449 {
2450   \tl_set:cn { CTEX@pre#1 } {#2}
2451   \IfNoValueTF {#3}
2452     { \tl_clear:c { CTEX@post#1 } }
2453     { \tl_set:cn { CTEX@post#1 } {#3} }
2454 }

```

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

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2455 \group_begin:
2456 <*book|report>
2457 \tl_set:Nn \l__ctex_tmp_tl
2458 {
2459   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2460   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2461   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2462   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2463   chapter / lofskip .initial:n = \c_zero_skip ,
2464   chapter / lotskip .initial:n = \c_zero_skip ,
2465   chapter / lofskip .value_required:n = true ,
2466   chapter / lotskip .value_required:n = true ,
2467 }
2468 </book|report>
2469 <*article|beamer>
2470 \tl_clear:N \l__ctex_tmp_tl
2471 </article|beamer>

```

定义标题键值选项。

```

2472 \seq_map_inline:Nn \c__ctex_headings_seq
2473 {
2474   \__ctex_initial_heading:n {#1}
2475   \__ctex_def_heading_keys:n {#1}
2476 }
2477 \use:x
2478 {
2479   \group_end:
2480   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2481 }

```

2482 `<*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。

```
2483 <*book|report>
2484 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2485 {
2486   \CTEX@fixheadingskip
2487   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2488     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2489 }
2490 </book|report>
```

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

```
2491 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2492 {
2493   \par
2494   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2495   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2496 }
2497 \skip_new:N \l__ctex_heading_skip
2498 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2499 { \skip_set:Nn \l__ctex_heading_skip }
2500 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip
```

`\partmark` 提供 `\partmark`。

```
2501 \ProvideDocumentCommand \partmark { m }
2502 { \markboth { } { } }
```

`\CTEXifname` 用于判断当前标题是否有编号。

```
\CTEX@ifnametrue
\CTEX@ifnamefalse
2503 \cs_new_eq:NN \CTEXifname \use_ii:nn
2504 \cs_new_protected_nopar:Npn \CTEX@ifnametrue
2505 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2506 \cs_new_protected_nopar:Npn \CTEX@ifnamefalse
2507 { \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。

```
2508 <*book|report>
2509 \cs_new_protected_nopar:Npn \CTEX@addloflotskip #1
2510 {
2511   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2512   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2513   {
2514     \addtocontents { lof }
2515     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2516   }
2517   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2518   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2519   {
2520     \addtocontents { lot }
2521     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2522   }
2523 }
2524 </book|report>
```

```
\CTEX@addtocline 2525 \cs_new_protected:Npn \CTEX@addtocline #1#2
2526 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }
```

14.8.2.1 part 的标题

```
2527 <@@=>

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

\@part 2566 <*article>
2567 \def\@part[#1]#2{%
2568   \ifnum \c@secnumdepth >\m@ne
2569     \ifodd \CTEX@part@numbering
2570       \CTEX@ifnametrue
2571       \refstepcounter{part}%
2572       % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2573     \else
2574       \CTEX@ifnamefalse
2575       \CTEX@makeanchor{part*}%
2576       % \addcontentsline{toc}{part}{#1}%
2577     \fi
2578   \else
2579     \CTEX@ifnamefalse
2580     \CTEX@makeanchor{part*}%
2581     % \addcontentsline{toc}{part}{#1}%
2582   \fi
2583   \CTEX@addtocline{part}{#1}%
2584   {\interlinepenalty \@M
2585   % \normalfont \parindent \z@ \raggedright
```

```

2586 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2587 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2588 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2589 \CTEX@partname \CTEX@part@aftername
2590 \fi \fi
2591 % \huge\bfseries #2%
2592 \CTEX@part@titleformat{#2}%
2593 % \markboth{}{}%
2594 \partmark{#1}%
2595 \CTEX@part@aftertitle}%
2596 \nobreak
2597 % \vskip 3ex
2598 \CTEX@setheadingskip \CTEX@part@afterskip
2599 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2600 \vskip \CTEX@headingskip
2601 \@afterheading}
2602 </article>
2603 <*book|report>
2604 \def\@part[#1]#2{%
2605 \ifnum \c@secnumdepth >-2\relax
2606 \ifodd \CTEX@part@numbering
2607 \CTEX@ifnametrue
2608 \refstepcounter{part}%
2609 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2610 \else
2611 \CTEX@ifnamefalse
2612 \CTEX@makeanchor{part*}%
2613 % \addcontentsline{toc}{part}{#1}%
2614 \fi
2615 \else
2616 \CTEX@ifnamefalse
2617 \CTEX@makeanchor{part*}%
2618 % \addcontentsline{toc}{part}{#1}%
2619 \fi
2620 \CTEX@addtocline{part}{#1}%
2621 % \markboth{}{}%
2622 \partmark{#1}%
2623 {\interlinepenalty \@M
2624 % \normalfont \centering
2625 \normalfont \CTEX@part@format
2626 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2627 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2628 \CTEX@partname \CTEX@part@aftername
2629 \fi \fi
2630 % \Huge\bfseries #2\par}%
2631 \CTEX@part@titleformat{#2}%
2632 \CTEX@part@aftertitle}%
2633 \@endpart}
2634 </book|report>

\@spart 2635 <*article>
2636 \def\@spart#1{%
2637 \CTEX@ifnamefalse
2638 \CTEX@makeanchor@spart{part*}%
2639 {\interlinepenalty \@M
2640 % \normalfont \parindent \z@ \raggedright
2641 \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2642 \CTEX@part@format
2643 % \huge \bfseries #1\par}%
2644 \CTEX@part@titleformat{#1}%
2645 \CTEX@part@aftertitle}%
2646 \nobreak
2647 % \vskip 3ex
2648 \CTEX@setheadingskip \CTEX@part@afterskip
2649 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2650 \vskip \CTEX@headingskip
2651 \@afterheading}

```

```

2652 </article>
2653 <*book|report>
2654 \def\@spart#1{%
2655     \CTEX@ifnamefalse
2656     \CTEX@makeanchor@spart{part*}%
2657     {\interlinepenalty \@M
2658      \normalfont \centering
2659      \normalfont \CTEX@part@format
2660      \Huge \bfseries #1\par}%
2661     \CTEX@part@titleformat{#1}%
2662     \CTEX@part@aftertitle}%
2663     \@endpart}
2664 </book|report>

\@endpart 2665 <*book|report>
2666 \def\@endpart{%
2667     %
2668     \CTEX@setheadingskip \CTEX@part@afterskip
2669     \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2670     \vskip \CTEX@headingskip
2671     \newpage
2672     \if@twoside
2673     \if@openright
2674     \null
2675     \thispagestyle{empty}%
2676     \newpage
2677     \fi
2678     \fi
2679     \if@tempswa
2680     \twocolumn
2681     \fi}
2682 </book|report>

```

14.8.2.2 chapter 的标题

```

2683 <*book|report>

\chapter 2684 \renewcommand\chapter{%
2685     %
2686     %
2687     \CTEX@chapter@break
2688     \thispagestyle{\CTEX@chapter@pagestyle}%
2689     \global\@topnum\z@
2690     %
2691     \ifodd \CTEX@chapter@afterindent
2692     \@afterindenttrue
2693     \else
2694     \@afterindentfalse
2695     \fi
2696     \secdef\@chapter\@schapter}

\@chapter 2697 \def\@chapter[#1]#2{%
2698     \ifnum \c@secnumdepth > \m@ne
2699     <*book>
2700     \if@mainmatter
2701     </book>
2702     \ifodd \CTEX@chapter@numbering
2703     \CTEX@ifnametrue
2704     \refstepcounter{chapter}%
2705     %
2706     \typeout{\@chapapp\space\thechapter.}%
2707     \typeout{\CTEXthechapter}%
2707     %
2708     \addcontentsline{toc}{chapter}
2708     %
2709     {\protect\numberline{\thechapter}#1}%
2709     \else
2710     \CTEX@ifnamefalse
2711     \CTEX@makeanchor{\Hy@chapapp*}%
2712     %
2713     \addcontentsline{toc}{chapter}{#1}%
2713     \fi

```

```

2714 <*book>
2715     \else
2716         \CTEX@ifnamefalse
2717         \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2718 %         \addcontentsline{toc}{chapter}{#1}%
2719     \fi
2720 </book>
2721     \else
2722         \CTEX@ifnamefalse
2723         \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2724 %         \addcontentsline{toc}{chapter}{#1}%
2725     \fi
2726     \CTEX@addtocline{chapter}{#1}%
2727     \chaptermark{#1}%
2728 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2729 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2730     \CTEX@addloflotskip{chapter}%
2731     \if@twocolumn
2732         \topnewpage[\@makechapterhead{#2}]%
2733     \else
2734         \@makechapterhead{#2}%
2735     \afterheading
2736     \fi}

\@schapter 2737 \def\@schapter#1{%
2738     \CTEX@ifnamefalse
2739     \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2740     \if@twocolumn
2741         \topnewpage[\@makeschapterhead{#1}]%
2742     \else
2743         \@makeschapterhead{#1}%
2744         \afterheading
2745     \fi}

\@makechapterhead 2746 \def\@makechapterhead#1{%
2747 % \vspace*{50\p@}%
2748     \CTEX@setheadingskip \CTEX@chapter@beforeskip
2749     \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2750     \vspace*{\CTEX@headingskip}%
2751 % {\normalfont \parindent \z@ \raggedright
2752 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2753 \CTEX@chapter@format
2754 % \ifnum \c@secnumdepth >\m@ne
2755 %     \if@mainmatter
2756 %         \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2757 %     \fi
2758 % \fi
2759 \CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}%
2760 \interlinepenalty\@M
2761 % \Huge \bfseries #1\par\nobreak
2762 \CTEX@chapter@titleformat{#1}%
2763 \CTEX@chapter@aftertitle
2764 \nobreak
2765 % \vskip 40\p@
2766 \CTEX@setheadingskip \CTEX@chapter@afterskip
2767 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2768 \vskip \CTEX@headingskip
2769 }}

\@makeschapterhead 2770 \def\@makeschapterhead#1{%
2771 % \vspace*{50\p@}%
2772     \CTEX@setheadingskip \CTEX@chapter@beforeskip
2773     \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2774     \vspace*{\CTEX@headingskip}%
2775 % {\normalfont \parindent \z@ \raggedright
2776 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2777 \CTEX@chapter@format

```

```

2778 \interlinepenalty\@M
2779 % \Huge \bfseries #1\par\nobreak
2780 \CTEX@chapter@titleformat{#1}
2781 \CTEX@chapter@aftertitle
2782 \nobreak
2783 % \vskip 40\p@
2784 \CTEX@setheadingskip \CTEX@chapter@afterskip
2785 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2786 \vskip \CTEX@headingskip
2787 }}
2788 </book|report>

```

14.8.2.3 section 类的标题

```

\startsection 2789 \def\startsection#1#2#3#4#5#6{%
2790 \if@noskipsec \leavevmode \fi
2791 \par
2792 % \@tempskipa #4\relax
2793 % \@afterindenttrue
2794 % \ifdim \@tempskipa <\z@
2795 % \@tempskipa -\@tempskipa \@afterindentfalse
2796 % \fi
2797 \CTEX@update@sectionformat@n{#1}%
2798 \ifodd \CTEX@afterindent
2799 \@afterindenttrue
2800 \else
2801 \@afterindentfalse
2802 \fi
2803 \if@nobreak
2804 \everypar{}%
2805 \else
2806 % \addpenalty\@secpenalty\addvspace\@tempskipa
2807 \csname CTEX@#1@break\endcsname
2808 \CTEX@setheadingskip{#4}%
2809 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2810 \addvspace \CTEX@headingskip
2811 \fi
2812 \ifstar
2813 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2814 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

\@secntformat 2815 \def\@secntformat#1{%
2816 % \csname the#1\endcsname\quad}%
2817 \csname CTEX@#1name\endcsname
2818 \csname CTEX@#1@aftername\endcsname}

\@sect 2819 \def\@sect#1#2#3#4#5#6[#7]#8{%
2820 \ifnum #2>\c@secnumdepth
2821 \CTEX@ifnamefalse
2822 \CTEX@makeanchor@sect{#1*}%
2823 \let\@svsec\@empty
2824 \else
2825 \ifodd \csname CTEX@#1@numbering\endcsname
2826 \CTEX@ifnametrue
2827 \refstepcounter{#1}%
2828 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2829 \else
2830 \CTEX@ifnamefalse
2831 \CTEX@makeanchor{#1*}%
2832 \let\@svsec\@empty
2833 \fi
2834 \fi
2835 % \@tempskipa #5\relax
2836 % \ifdim \@tempskipa>\z@
2837 \unless \ifodd \CTEX@runin
2838 \begingroup
2839 #6{%

```

```

2840         \CTEX@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2841 %         \interlinepenalty \OM #8\@par}%
2842         \interlinepenalty \OM
2843         \csname CTEX@#1@titleformat\endcsname{#8}%
2844         \csname CTEX@#1@aftertitle\endcsname}%
2845     \endgroup
2846     \csname #1mark\endcsname{#7}%
2847 %     \addcontentsline{toc}{#1}{%
2848 %         \ifnum #2>\c@secnumdepth \else
2849 %             \protect\numberline{\csname the#1\endcsname}%
2850 %         \fi
2851 %         #7}%
2852     \CTEX@addtocline{#1}{#7}%
2853 \else
2854     \def\@svsechd{%
2855         #6{\hskip\glueexpr #3\relax
2856 %         \@svsec #8}%
2857         \@svsec
2858         \csname CTEX@#1@titleformat\endcsname{#8}%
2859         \csname CTEX@#1@aftertitle\endcsname}%
2860     \csname #1mark\endcsname{#7}%
2861 %     \addcontentsline{toc}{#1}{%
2862 %         \ifnum #2>\c@secnumdepth \else
2863 %             \protect\numberline{\csname the#1\endcsname}%
2864 %         \fi
2865 %         #7}%
2866     \CTEX@addtocline{#1}{#7}%
2867 \fi
2868 \@xsect{#5}}

\@ssect 2869 \def\@ssect#1#2#3#4#5{%
2870     \CTEX@ifnamefalse
2871 %     \@tempskipa #3\relax
2872 %     \ifdim \@tempskipa>\z@
2873     \unless \ifodd \CTEX@runin
2874     \begingroup
2875         #4{%
2876             \CTEX@hangfrom{\hskip\glueexpr #1\relax}%
2877 %             \interlinepenalty \OM #5\@par}%
2878             \interlinepenalty \OM
2879             \CTEX@titleformatn{#5}%
2880             \CTEX@aftertitle}%
2881     \endgroup
2882 \else
2883 %     \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2884     \def\@svsechd{#4{\hskip\glueexpr #1\relax
2885         \CTEX@titleformatn{#5}\CTEX@aftertitle}}%
2886 \fi
2887 \@xsect{#3}}

\@xsect 2888 \def\@xsect#1{%
2889 %     \@tempskipa #1\relax
2890 %     \ifdim \@tempskipa>\z@
2891     \unless \ifodd \CTEX@runin
2892     \par \nobreak
2893 %     \vskip \@tempskipa
2894     \CTEX@setheadingskip{#1}%
2895     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2896     \vskip \CTEX@headingskip
2897     \@afterheading
2898 \else
2899     \@nobreakfalse
2900     \global\@noskipsectrue
2901     \everypar{%
2902         \if@noskipsec
2903             \global\@noskipsecfalse
2904             {\setbox\z@\lastbox}%

```

```

2905         \clubpenalty \@M
2906         \begingroup \@svsechd \endgroup
2907         \unskip
2908         \@tempskipa #1\relax
2909         \hskip -\@tempskipa
2910         \hskip\glueexpr #1\relax
2911     \else
2912         \clubpenalty \@clubpenalty
2913         \everypar{}%
2914     \fi}%
2915 \fi
2916 \ignorespaces}

2917 <@@-ctex>

```

\CTEX@hangfrom hang 选项控制是否采用悬挂缩进。

```

2918 \cs_new_protected_nopar:Npn \CTEX@hangfrom
2919 {
2920     \bool_if:NTF \CTEX@hang
2921     { \@hangfrom }
2922     { \noindent \use:n }
2923 }

```

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

```

2924 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2925 {
2926     \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2927     \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2928     \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2929     \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2930     \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2931     \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2932 }
2933 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2934 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2935 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2936 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2937 \cs_new_eq:NN \CTEX@hang \c_true_bool
2938 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

```

\CTEX@part@tocline 2939 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 2940 {
2941     \CTEXifname
2942     { \CTEXthepart \hspace { 1em } }
2943     { }
2944     #2
2945 }
2946 <*book|report>
2947 \cs_new:Npn \CTEX@chapter@tocline #1#2
2948 {
2949     \CTEXifname
2950     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2951     { }
2952     #2
2953 }
2954 </book|report>

```

```

\CTEXnumberline 2955 \cs_new_nopar:Npn \CTEXnumberline #1
2956 {
2957     \CTEXifname
2958     { \protect \numberline { \use:c { CTEXthe #1 } } }
2959     { }
2960 }

2961 \int_zero:N \l__ctex_tmp_int
2962 \seq_map_inline:Nn \c__ctex_section_headings_seq

```

```

2963 {
2964   \int_incr:N \l__ctex_tmp_int
2965   \cs_gset_protected_nopar:cpx {#1}
2966   {
2967     \exp_not:N \@startsection {#1}
2968     { \int_use:N \l__ctex_tmp_int }
2969     { \exp_not:c { CTEX@#1@indent } }
2970     { \exp_not:c { CTEX@#1@beforeskip } }
2971     { \exp_not:c { CTEX@#1@afterskip } }
2972     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2973   }
2974   \cs_new:cpn { CTEX@#1@tocline } ##1##2
2975   { \CTEXnumberline { ##1 } ##2 }
2976 }

```

14.8.2.4 附录标题

```

appendix/name 2977 \keys_define:nn { ctex }
appendix/number 2978 {
appendix/numbering 2979   appendix .meta:nn = { ctex / appendix } {#1} ,
2980   appendix / name .code:n =
2981   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2982   appendix / number .tl_set:N = \CTEX@appendix@number ,
2983   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2984   appendix / numbering .initial:n = true
2985 }
2986 \tl_new:N \CTEX@preappendix
2987 \tl_new:N \CTEX@postappendix

\appendix 2988 \cs_new_eq:NN \CTEX@save@appendix \appendix
2989 \cs_gset_protected_nopar:Npn \appendix
2990 {
2991   \CTEX@save@appendix
2992   <*article>
2993   \gdef \CTEX@presection { \CTEX@preappendix }
2994   \gdef \CTEX@thesection { \CTEX@appendix@number }
2995   \gdef \CTEX@postsection { \CTEX@postappendix }
2996   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2997   </article>
2998   <*book|report>
2999   \gdef \CTEX@prechapter { \CTEX@preappendix }
3000   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3001   \gdef \CTEX@postchapter { \CTEX@postappendix }
3002   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3003   </book|report>
3004 }

```

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。

```

3005 \cs_new_protected_nopar:Npn \CTEX@makeanchor #1
3006 { }

```

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

```

3007 \seq_new:N \c__ctex_headings_cs_seq
3008 \seq_gset_from_clist:Nn \c__ctex_headings_cs_seq
<article> 3009 { part , spart , sect , ssect }
<book|report> 3010 { part , spart , chapter , schapter , sect , ssect }
3011 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3012 {
3013   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3014   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3015 }

```

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

```

3016 \cs_new_protected_nopar:Npn \CTEX@hyperheadinghook
3017 {
3018   \group_begin:
3019   \ifHy@implicit
3020     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3021     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3022     {
3023       \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3024       {
3025         \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3026         \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3027     }
3028   }
3029   \else:
3030     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3031     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3032   \fi:
3033   \group_end:
3034 }

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

```

14.8.2.6 兼容 `titlesec` 宏包

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

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

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

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

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

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

其中 `<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 `\ttlhf@<shape>` 和 `<afterindent>`。注意，由 `\ttl@extract` 得的 `<before>` 和 `<after>` 的值总是非负的，而 `CTEX` 的 `beforeskip` 和 `afterskip` 是可以取负值的，但我们不打算调整它们了。如果使用了 `titlesec` 的 `indentafter` 等选项，也不需要调整 `\ttls@<section>`。

```
3048 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
```

```

3049 {
3050   \@ifpackagewith { titlesec } { explicit }
3051   {
3052     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3053       \__ctex_titlesec_format_explicit:Nn
3054   }
3055   { }
3056   \clist_map_inline:nn
3057     { indentafter , noindentafter , indentfirst , nonindentfirst }
3058     {
3059       \@ifpackagewith { titlesec } { ##1 }
3060       {
3061         \clist_map_break:n
3062           { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3063       }
3064       { }
3065     }
3066   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3067 }
3068 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
3069 {
3070   \__ctex_titlesec_format:n {#1}
3071   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3072 }
3073 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
3074 {
3075   \cs_if_free:cF { ttlf@#1 }
3076   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3077 }
3078 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
3079 {
3080   \tl_set:Nx #1
3081   {
3082     \bool_if:cTF { CTEX@#2@runin }
3083     { \exp_not:N \ttlh@runin }
3084     { \exp_not:N \ttlh@hang }
3085     \tl_tail:N #1
3086   }
3087 }
3088 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3089 {
3090   \cs_set_nopar:Npx #1 ##1
3091   {
3092     \bool_if:cTF { CTEX@#2@runin }
3093     { \exp_not:N \ttlh@runin }
3094     { \exp_not:N \ttlh@hang }
3095     \exp_args:No \tl_tail:n { #1 { } }
3096   }
3097 }
3098 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
3099 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3100 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3101 {
3102   \exp_not:n { {#1} {#2} {#3} {#4} }
3103   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3104 }
3105 \@ifpackageloaded { titlesec }
3106 { }
3107 {
3108   \ctex_at_end_package:nn { titlesec }
3109   {
3110     \@ifpackagewith { titlesec } { loadonly }
3111     { }
3112     { \ctex_titlesec_hook: }
3113   }
3114 }

```

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

```

3115 \ctex_at_end_package:nn { titlesec }
3116 {
3117   <*book|report>
3118   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3119   </book|report>
3120   \cs_if_free:NF \ttl@tocpart
3121   {
3122     \cs_set_protected_nopar:Npn \ttl@tocpart
3123     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3124   }
3125   \seq_map_inline:Nn \c__ctex_headings_seq
3126   {
3127     \cs_if_exist:cF { ttl@toc #1 }
3128     {
3129       \cs_new_protected_nopar:cpx { ttl@toc #1 }
3130       {
3131         \tl_set:Nn \exp_not:N \ttl@a
3132         {
3133           \exp_not:N \protect
3134           \exp_not:N \numberline { \exp_not:c { CTExthe #1 } }
3135         }
3136       }
3137     }
3138   }
3139 }

```

14.8.2.7 兼容 titleps 宏包

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

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

```

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

```

`\CTEX@titlepslabel@set`
`\CTEX@titlepslabel@clear`

这两个函数要在随后被 `\xdef` 展开来获得 `\CTEXtheXXX` 的内容, 不应该用 `\protected` 来定

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

义。

```

3164 \cs_new_nopar:Npn \CTEX@titleslabel@set #1
3165 {
3166   \cs_if_free:cF { CTEXthe#1 }
3167   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3168 }
3169 \cs_new_nopar:Npn \CTEX@titleslabel@clear #1
3170 {
3171   \cs_if_free:cF { CTEXthe#1 }
3172   { \protect \@namedef { CTEXthe#1 } { } }
3173 }

```

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

```

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

```

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

```

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

```

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

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

```

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

```

14.8.3 目录标签的宽度

```

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

```

```

3213         \ctex_preto_cmd:NnnTF \numberline
3214         { \char_set_catcode_letter:n { 64 } }
3215         { \CTEX@toc@width@n {#1} }
3216         { }
3217         { \ctex_patch_failure:N \numberline }
3218     }
3219 }
3220 }

```

14.8.4 页眉信息的修改

```

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

```

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

```

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

```

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

```

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

```

3270 `</article|book|report>`

14.8.5 beamer 标题页模板的修改

3271 `<*beamer>`

3272 `\ExplSyntaxOff`

对应 `\partpage`。

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

对应 `\sectionpage`。

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

对应 `\subsectionpage`。

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

```

3326     \end{beamercolorbox}
3327     \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3328     \endgroup
3329 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3330 \defbeamer-templatealias{part page}{default}{CTEX}
3331 \defbeamer-templatealias{section page}{default}{CTEX}
3332 \defbeamer-templatealias{subsection page}{default}{CTEX}
3333 \ExplSyntaxOn
3334 </beamer>

```

14.8.6 标签引用数字的汉化

`\refstepcounter` 对标题进行引用时,设置标签为通过 `number` 选项设置的形式。

```

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

```

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

```

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

```

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

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

它的第二个参数必须放在花括号内,否则将会被作为宏的定界符号。

```

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

```

如果 `varioref` 已经被载入,则使用它来设置。

```

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

```

14.8.7 载入 $\langle scheme \rangle$ 文件

```
3365 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3366 </class|heading>
```

14.8.8 标题格式的 scheme 定义

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

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

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

```

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

3450 <@@= >

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

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

```

```

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

3517 <*article|book|report>

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

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

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

```

```

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

```

处理 sub3section 与 sub4section 的格式。

```

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

```

处理附录的格式。

```

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

```

14.8.9 ctex.sty 的 heading 选项

3611 `<*ctex|ctexheading>`

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

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

若标准文档类被引入, 则载入对应的标题定义文件。否则视 `\chapter` 是否有定义来引入 `book` 或者 `article`。

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

14.8.10 标题配置文件

3643 `<*name>`

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

3662 `</name>`

14.9 chinese 方案的其他设置

3663 `<*scheme&chinese>`

chinese 在标准文档类下的页面格式总采用 headings。

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

日期格式。

3665 `\keys_set:nn { ctex } { today = small }`

若用户未设置宏包选项 autoindent, 则自动调整首行缩进。

3666 `\ctex_if_autoindent_touched:F`

3667 `{ \keys_set:nn { ctex } { autoindent = true } }`

使用标题定义时的设置。首先是命题名字汉化。beamer 需要汉化定理名称。

3668 `<!*generic>`

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

3670 `<*beamer>`

3671 `{`

3672 `\uselanguage { ChineseGBK }`

3673 `\languagealias { chinese } { ChineseGBK }`

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

3675 `}`

3676 `{`

3677 `\uselanguage { ChineseUTF8 }`

3678 `\languagealias { chinese } { ChineseUTF8 }`

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

3680 `}`

让 translator 包优先查找中文翻译。

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

3682 `</beamer>`

3683 `<!*beamer>`

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

3685 `{ \ctex_file_input:n { ctex-name-utf8.cfg } }`

对 beamer 以外的文档类, 若用户未设置宏包选项 zihao, 则设置 \normalsize 为五号字。beamer 不调整默认字体大小。

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

3687 `{ \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }`

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

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

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

3690 `</!beamer>`

3691 `</!generic>`

不使用标题定义时的通用设置。

3692 `<*generic>`

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

3694 `\tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl`

3695 `{`

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

3697 `{`

3698 `\uselanguage { ChineseGBK }`

3699 `\languagealias { chinese } { ChineseGBK }`

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

3701 `}`

3702 `{`

3703 `\uselanguage { ChineseUTF8 }`

3704 `\languagealias { chinese } { ChineseUTF8 }`

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

3706 `}`

```

3707 \clist_put_left:Nn \trans@languagepath { chinese }
3708 }
3709 {
3710 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3711 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3712 { \ctex_file_input:n { ctex-name-utf8.cfg } }
3713 \int_compare:nNnF \g__ctex_font_size_flag > { -1 }
3714 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3715 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3716 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }

```

若 `ctex` 宏包与标准文档类及其衍生文档类联用, 则将载入 `indentfirst` 宏包, 实现章节标题后首个段落的段首缩进。

```

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

```

14.10 中文字号

```

3722 <*class|ctex|ctexsize>

```

```

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

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

```

14.10.1 定义中文字号

```

\c__ctex_font_size_prop
\__ctex_save_font_size:nn

```

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

```

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

```

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

```

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

```

```

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

```

```

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

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

```

14.10.2 修改默认字号大小

```

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

3815 \if_case:w \g__ctex_font_size_flag

```

```

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

```

```

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

```

14.11 更新行距

`\l__ctex_line_spread_fp` 被设置了才有必要更新行距和 `\footnotesep`。

```

3901 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3902 {
3903 \exp_args:Nx \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

`\footnotesep` 我们调整了行距，可能导致脚注的间距与行距不协调，需要调整 `\footnotesep`。标准文档类对 `\footnotesep` 的设置是，字体大小为 `\footnotesize` 时 `\strutbox` 的高度（默认值是 `.7\baselineskip`）。我们沿用这个设置方法，只需要更新具体的大小。

```

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

```

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

```

3907 \normalsize

```

14.12 其它功能

`\CTeX` `ctex-faq.sty` 中的定义是

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

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

```

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

```

`captiondelimit` 过时选项。

```

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

```

3920 `</class|ctex>`

14.12.1 列表环境的缩进

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

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

14.13 载入中文字体

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

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

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

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

`fontset` 在导言区通过 `\ctexset` 载入中文字库的选项。

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

```

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

```

载入中文字库。

```

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

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

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

```

14.14.2 ctexopts.cfg

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

```

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

```

14.15 字体定义文件

14.15.1 传统定义方式

```

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

```

CJK 宏包使用的字体族。

```

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

```

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

```

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

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

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

```

```

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

```

14.15.2 ctex-fontset-windows.def 等

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

```

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

```

旧的 Windows 字体设置使用黑体作为无衬线体，楷体和仿宋是 GB2312 编码；新的 Windows 字体设置使用微软雅黑作为无衬线体，楷体和仿宋是大字库。Windows 8 以后，微软雅黑由原来的 .ttf 后缀改为 .ttc 后缀，需要加以区分。

```

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

```

```

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

```

```

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

```

14.15.3 ctex-fontset-adobe.def

```

4223 <*adobe>

```

\pdfmapline 不支持 OpenType 字体，因而 adobe 字体集在 pdf 模式下就没有定义。fandol 的情况类似。

```

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

```

```

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

```

14.15.4 ctex-fontset-fandol.def

```

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

```

```

4317         [
4318             cmap = UniGB-UTF16-H ,
4319             BoldFont = FandolHei-Bold.otf
4320         ] { FandolHei-Regular.otf }
4321     \setCJKfamilyfont { zhfs }
4322     [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4323     \setCJKfamilyfont { zhkai }
4324     [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4325     \ctex_punct_set:n { fandol }
4326     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4327     \ctex_punct_map_family:nn { \JKsfdefault } { zhhei }
4328     \ctex_punct_map_family:nn { \JKttdefault } { zhfs }
4329     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4330     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4331     \ctex_punct_map_bfseries:nn { \JKsfdefault , zhhei } { zhheib }
4332 }
4333 {
4334     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4335     \ctex_punct_set:n { fandol }
4336     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4337     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4338     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4339 }
4340 { \ctex_fontset_error:n { fandol } }
4341 }
4342 }
4343 {
4344     \sys_if_engine_uptex:TF
4345     {
4346         \ctex_set_upfonts:nnnnnn
4347         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4348         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4349         {FandolFang-Regular.otf}
4350         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4351         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4352         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4353         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4354     }
4355     {
4356         \setCJKmainfont
4357         [
4358             Extension = .otf ,
4359             BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular
4360         ]
4361         { FandolSong-Regular }
4362         \setCJKsansfont
4363         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4364         \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4365         \setCJKfamilyfont { zhsong }
4366         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4367         \setCJKfamilyfont { zhhei }
4368         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4369         \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4370         \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4371     }
4372 }
4373 </fandol>
4374 <*mac>

```

14.15.5 ctex-fontset-mac.def

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

/Library/Fonts/Baoli.ttc: 报隶-简,Baoli SC:style=Regular
 /Library/Fonts/Hannotate.ttc: 手札体-简,Hannotate SC:style=Bold

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

```

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

```

在 DVIPDFM_x 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}

```

DVIPDFM_x 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²³。华文细黑和华文仿宋的情况类似。这确认为 DVIPDFM_x 的 bug²⁴。

```

4375 \sys_if_engine_pdftex:TF
4376 { \ctex_fontset_error:n { mac } }
4377 {
4378   \sys_if_engine_uptex:TF
4379   { \ctex_fontset_error:n { mac } }
4380   {
4381     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4382     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }

```

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

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

```

4383     \setCJKmonofont { STFangsong }
4384     \setCJKfamilyfont { zhsong } { STSong }
4385     \setCJKfamilyfont { zhhei } { STHeiti }
4386     \setCJKfamilyfont { zhfs } { STFangsong }
4387     \setCJKfamilyfont { zhkai } { STKaiti }
4388   }
4389 }

4390 </mac>
4391 <*founder>

```

14.15.6 ctex-fontset-founder.def

```

4392 \sys_if_engine_pdftex:TF
4393 {
4394   \ctex_zhmap_case:nnn
4395   {
4396     \setCJKmainfont
4397       [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4398     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4399     \setCJKmonofont { FZFSK.TTF }
4400     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4401     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4402     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4403     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4404     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4405     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4406     \ctex_punct_set:n { founder }
4407     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4408     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4409     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4410     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4411     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4412     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4413     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4414   }
4415   {
4416     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4417     \ctex_punct_set:n { founder }
4418     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4419     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4420     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4421   }
4422   { \ctex_fontset_error:n { founder } }
4423 }
4424 {
4425   \sys_if_engine_uptex:TF
4426   {
4427     \ctex_set_upfonts:nnnnnn
4428       { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4429       { FZXH1K.TTF } { FZHTK.TTF }
4430       { FZFSK.TTF }
4431     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4432     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4433     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4434     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4435     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4436     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4437     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4438     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4439   }
4440   {
4441     \setCJKmainfont
4442       [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }

```

在 WPS For Linux 附带的 5.00 版和家庭版 5.20 版的“方正细黑一_GBK”的字体名称是

FZXiHeiI-Z08。但在网上发现不少文档和资料都是 FZXiHeiI-Z08,而在官方资料《2013 方正字库字体样张》中对应的英文名字是 FZXiHeiI。可以用 Postscript 名字 FZXH1K--GBK1-0 来统一。经测试时发现 (WPS 中的字体), \LaTeX 找该字体时会出现明显的延迟,而用字体文件名 FZXH1K.TTF 又可能会出现大小写问题,遂采用汉字名称。由于 luaotfload 不支持汉字名称,故使用 Postscript 名字,速度不受影响。

```

4443     \setCJKsansfont [ BoldFont = FZHei-B01 ]
4444     { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4445     \setCJKmonofont { FZFangSong-Z02 }
4446     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4447     \setCJKfamilyfont { zhhei } { FZHei-B01 }
4448     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4449     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4450     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4451     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4452   }
4453 }
4454 </founder>
4455 <*ubuntu>

```

14.15.7 ctex-fontset-ubuntu.def

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

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

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

```

4456 \sys_if_engine_pdftex:TF
4457 {
4458   \ctex_zhmap_case:nnn
4459   {
4460     \setCJKmainfont
4461     [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4462     \setCJKsansfont { wqy-zenhei.ttc }
4463     \setCJKmonofont { uming.ttc }
4464     \setCJKfamilyfont { zhsong } { uming.ttc }
4465     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4466     \setCJKfamilyfont { zhkai } { ukai.ttc }
4467     \ctex_punct_set:n { ubuntu }
4468     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4469     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4470     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4471     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4472     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4473   }
4474   {
4475     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4476     \ctex_punct_set:n { ubuntu }
4477     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4478     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4479     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4480   }
4481   { \ctex_fontset_error:n { ubuntu } }
4482 }
4483 {
4484   \sys_if_engine_uptex:TF
4485   {
4486     \ctex_set_upfonts:nnnnnn
4487     { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4488     { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4489     { uming.ttc }

```

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

```

4490     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4491     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4492     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4493   }
4494   {
4495     \setCJKmainfont
4496     [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4497     \setCJKsansfont { WenQuanYi~Zen~Hei }
4498     \setCJKmonofont { AR~PL~UMing~CN }
4499     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4500     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4501     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4502   }
4503 }
4504 </ubuntu>

```

14.15.8 中文字体命令

```

4505 <!*windows>
4506 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4507 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
</ubuntu> 4508 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4509 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4510 <*windowsnew|windowsold|founder>
4511 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4512 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4513 </windowsnew|windowsold|founder>
<windowsnew> 4514 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4515 </!windows>
4516 </fontset>
4517 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4518 \begingroup\catcode61\catcode48\catcode32=10\relax%
4519 \catcode 35=6 % #
4520 \catcode123=1 % {
4521 \catcode125=2 % }
4522 \toks0{\endlinechar=\the\endlinechar\relax}%
4523 \toks2{\endlinechar=13}%
4524 \def\x#1 #2 {%
4525   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4526   \toks2\expandafter{\the\toks2 \catcode#1=#2}}%
4527 \x 13 5 % carriage return
4528 \x 32 10 % space
4529 \x 35 6 % #
4530 \x 40 12 % (
4531 \x 41 12 % )
4532 \x 45 12 % -
4533 \x 46 12 % .
4534 \x 47 12 % /
4535 \x 58 12 % :
4536 \x 60 12 % <
4537 \x 61 12 % =
4538 \x 64 11 % @
4539 \x 91 12 % [
4540 \x 93 12 % ]
4541 \x 123 1 % {
4542 \x 125 2 % }
4543 \edef\x#1{\endgroup%
4544   \edef\noexpand#1{%
4545     \the\toks0 %
4546     \let\noexpand\noexpand\noexpand#1%

```

```

4547 \noexpand\noexpand\noexpand\undefined%
4548 \noexpand\noexpand\noexpand\endinput}%
4549 \the\toks2}%
4550 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4551 \begingroup\expandafter\endgroup
4552 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4553 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4554 \expandafter\ifx\csname pdfoutput\endcsname\relax
4555 \iffalse\else\ifnum\pdfoutput < 1 \iffalse\else \iftrue\fi\fi
4556 \else ifctexpdf\fi
4557 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4558 \begingroup
4559 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4560 \long\def\x#1\ProvidesFile#2[#3]{%
4561 #1%
4562 \immediate\write-1{File: #2 #3}%
4563 \expandafter\xdef\csname ver@#2\endcsname{#3}}
4564 \expandafter\x%
4565 \fi
4566 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4567 <*windows>
4568 \ProvidesFile{zhwindowsfonts.tex}%
4569 [2017/11/22 v2.4.11 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4570
4571 \ifzhmappdf
4572 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4573 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4574 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4575 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4576 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4577 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4578 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4579
4580 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4581 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4582 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4583 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4584 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4585 \pdfmapline{=unili@Unicode@ <simli.ttf}
4586 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4587
4588 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4589 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4590 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4591 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4592 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4593 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4594
4595 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4596 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4597 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4598 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4599 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4600 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4601
4602 \else
4603 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4604 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4605 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4606 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}

```

```

4607 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4608 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4609 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4610
4611 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4612 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4613 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4614 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4615 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4616 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4617 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4618
4619 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4620 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4621 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4622 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4623 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4624 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4625
4626 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4627 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4628 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4629 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4630 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4631 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4632
4633 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4634 <*adobe>
4635 \ProvidesFile{zhadobefonts.tex}%
4636 [2017/11/22 v2.4.11 Adobe font map loader for DVIPDFMx (CTEX)]
4637
4638 \ifzhmappdf
4639 %% pdfTeX does not support OTF fonts
4640
4641 \else
4642 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4643 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4644 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4645 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4646 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4647
4648 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4649 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4650 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4651 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4652 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4653
4654 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4655 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4656 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4657 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4658
4659 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4660 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4661 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4662 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4663
4664 </adobe>

```

14.15.9.3 zhfandolfonts.tex

```

4665 <*fandol>
4666 \ProvidesFile{zhfandolfonts.tex}%
4667 [2017/11/22 v2.4.11 Fandol font map loader for DVIPDFMx (CTEX)]
4668

```

```

4669 \ifzhmappdf
4670 %% pdfTeX does not support OTF fonts
4671
4672 \else
4673 \special{pdf:mapline gbk@UGBK@      UniGB-UTF16-H FandolSong-Regular.otf}
4674 \special{pdf:mapline gbksong@UGBK@  UniGB-UTF16-H FandolSong-Regular.otf}
4675 \special{pdf:mapline gbkkai@UGBK@   UniGB-UTF16-H FandolKai-Regular.otf}
4676 \special{pdf:mapline gbkhei@UGBK@   UniGB-UTF16-H FandolHei-Regular.otf}
4677 \special{pdf:mapline gbkfs@UGBK@    UniGB-UTF16-H FandolFang-Regular.otf}
4678
4679 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4680 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4681 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4682 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4683 \special{pdf:mapline unifs@Unicode@  UniGB-UTF16-H FandolFang-Regular.otf}
4684
4685 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4686 \special{pdf:mapline gbkkaisl@UGBK@  UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4687 \special{pdf:mapline gbkheisl@UGBK@   UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4688 \special{pdf:mapline gbkfssl@UGBK@    UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4689
4690 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4691 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4692 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4693 \special{pdf:mapline unifssl@Unicode@  UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4694
4695 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4696 <*founder>
4697 \ProvidesFile{zhfounderfonts.tex}%
4698 [2017/11/22 v2.4.11 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4699
4700 \ifzhmappdf
4701 \pdfmapline{=gbk@UGBK@      <FZSSK.TTF>}
4702 \pdfmapline{=gbksong@UGBK@  <FZSSK.TTF>}
4703 \pdfmapline{=gbkkai@UGBK@   <FZKTK.TTF>}
4704 \pdfmapline{=gbkhei@UGBK@   <FZHTK.TTF>}
4705 \pdfmapline{=gbkfs@UGBK@    <FZFSK.TTF>}
4706 \pdfmapline{=gbkli@UGBK@    <FZLSK.TTF>}
4707 \pdfmapline{=gbkyou@UGBK@   <FZY1K.TTF>}
4708
4709 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF>}
4710 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF>}
4711 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF>}
4712 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF>}
4713 \pdfmapline{=unifs@Unicode@  <FZFSK.TTF>}
4714 \pdfmapline{=unili@Unicode@  <FZLSK.TTF>}
4715 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF>}
4716
4717 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF>}
4718 \pdfmapline{=gbkkaisl@UGBK@  <FZKTK.TTF>}
4719 \pdfmapline{=gbkheisl@UGBK@   <FZHTK.TTF>}
4720 \pdfmapline{=gbkfssl@UGBK@    <FZFSK.TTF>}
4721 \pdfmapline{=gbklisl@UGBK@    <FZLSK.TTF>}
4722 \pdfmapline{=gbkyousl@UGBK@   <FZY1K.TTF>}
4723
4724 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF>}
4725 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF>}
4726 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF>}
4727 \pdfmapline{=unifssl@Unicode@  <FZFSK.TTF>}
4728 \pdfmapline{=unilisl@Unicode@  <FZLSK.TTF>}
4729 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF>}
4730
4731 \else
4732 \special{pdf:mapline gbk@UGBK@      unicode FZSSK.TTF}
4733 \special{pdf:mapline gbksong@UGBK@  unicode FZSSK.TTF}

```

```

4734 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4735 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4736 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4737 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4738 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4739
4740 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4741 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4742 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4743 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4744 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4745 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4746 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF}
4747
4748 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4749 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4750 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4751 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4752 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4753 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4754
4755 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4756 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4757 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4758 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4759 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4760 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4761
4762 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4763 <*ubuntu>
4764 \ProvidesFile{zhubuntufonts.tex}%
4765 [2017/11/22 v2.4.11 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4766
4767 \ifzhmappdf
4768 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4769 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4770 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4771 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4772 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4773 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4774
4775 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4776 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4777 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4778 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4779 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4780 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4781
4782 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4783 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4784 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4785 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4786 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4787
4788 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4789 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4790 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4791 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4792 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4793
4794 \else
4795 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4796 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4797 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4798 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}

```

```

4799 \special{pdf:mapline gbksf@UGBK@ unicode :0:uming.ttc}
4800
4801 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4802 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4803 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4804 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4805 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4806
4807 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4808 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4809 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4810 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4811
4812 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4813 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4814 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4815 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4816
4817 </ubuntu>
4818 \fi
4819
4820 \ctex@zhmap@endinput
4821 </zhmap>

```

14.15.10 制作 spa 文件

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

```

4822 <*spa>
4823 <*macro>
4824 \input expl3-generic %
4825 \ExplSyntaxOn
4826 \sys_if_engine_xetex:F
4827 {
4828   \msg_new:nn { ctex } { xetex }
4829   { XeTeX~is~required~to~compile~this~document! }
4830   \msg_fatal:nn { ctex } { xetex }
4831 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

4832 \seq_new:N \c__ctex_punct_seq
4833 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4834 {
4835   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4836   "3008 , "300A , "3016 , "3010 ,
4837   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4838   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4839   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4840 }

```

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

```

4841 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4842 {
4843   \iow_open:Nn \g__ctex_spa_iow {#1}
4844   \clist_map_inline:nn {#2}
4845   { \__ctex_write_family:nn ##1 }
4846   \iow_close:N \g__ctex_spa_iow
4847 }
4848 \iow_new:N \g__ctex_spa_iow

```

```

4849 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn
4850 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4851 {
4852   \group_begin:
4853   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4854   \l__ctex_punct_font
4855   \clist_clear:N \l__ctex_punct_bounds_clist
4856   \seq_map_inline:Nn \c__ctex_punct_seq
4857   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4858   \iow_now:Nx \g__ctex_spa_iow
4859   {
4860     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4861     { \l__ctex_punct_bounds_clist , , , }
4862   }
4863   \group_end:
4864 }
4865 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4866 {
4867   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4868   {
4869     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4870     \__ctex_calc_bounds:nn { \c_three } {#1}
4871   }
4872 }
4873 \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。

```

4874 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4875 {
4876   \fp_eval:n
4877   {
4878     round
4879     (
4880       \dim_to_decimal_in_unit:nn
4881       { 100 \xetex_glyphbounds:D #1 ~ #2 }
4882       { 1 em }
4883     )
4884   }
4885 }
4886 \ExplSyntaxOff
4887 </macro>

```

下面是 CTeX 定义的一些字体。

```

4888 <*make>
4889 \input ctexspamacro %
4890
4891 \MAKESPA {ctexpunct.spa}
4892 {
4893   {adobezhsong}      {AdobeSongStd-Light} ,
4894   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4895   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4896   {adobezhfs}        {AdobeFangsongStd-Regular} ,
4897   {fandolzhsong}     {FandolSong} ,
4898   {fandolzhsongb}    {FandolSong-Bold} ,
4899   {fandolzhhei}      {FandolHei} ,
4900   {fandolzhheib}     {FandolHei-Bold} ,
4901   {fandolzhkai}      {FandolKai} ,
4902   {fandolzhfs}       {FandolFang} ,
4903   {founderzhsong}    {FZShuSong-Z01} ,
4904   {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
4905   {founderzhhei}     {FZHei-B01} ,
4906   {founderzhheil}    {FZXiHeiI-Z08} ,
4907   {founderzhkai}     {FZKai-Z03} ,

```

```

4908 {founderzhfs} {FZFangSong-Z02} ,
4909 {founderzhli} {FZLiShu-S01} ,
4910 {founderzhyou} {FZXiYuan-M01} ,
4911 {founderzhyoub} {FZZhunYuan-M02} ,
4912 {ubuntuzhsong} {AR PL Uming CN} ,
4913 {ubuntuzhhei} {WenQuanYi Zen Hei} ,
4914 {ubuntuzhkai} {AR PL UKai CN} ,
4915 {windowszhsong} {SimSun} ,
4916 {windowszhhei} {SimHei} ,
4917 {windowszhkai} {KaiTi} ,
4918 {windowszhfs} {FangSong} ,
4919 {windowszhli} {LiSu} ,
4920 {windowszhyou} {YouYuan} ,
4921 {windowszhyhei} {Microsoft YaHei} ,
4922 {windowszhyheib} {Microsoft YaHei Bold}
4923 }
4924
4925 \primitive\end
4926 </make>
4927 </spa>

```

14.16 translator 宏包的中文字典

```
4928 <*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。

```

4929 <*theorem>
4930 \providetranslation{Comments}{评论}
4931 \providetranslation{comments}{评论}
4932 \providetranslation{Comment}{评论}
4933 \providetranslation{comment}{评论}
4934 \providetranslation{Corollaries}{推论}
4935 \providetranslation{corollaries}{推论}
4936 \providetranslation{Corollary}{推论}
4937 \providetranslation{corollary}{推论}
4938 \providetranslation{Definitions}{定义}
4939 \providetranslation{definitions}{定义}
4940 \providetranslation{Definition}{定义}
4941 \providetranslation{definition}{定义}
4942 \providetranslation{Examples}{例}
4943 \providetranslation{examples}{例}
4944 \providetranslation{Example}{例}
4945 \providetranslation{example}{例}
4946 \providetranslation{Exercises}{练习}
4947 \providetranslation{exercises}{练习}
4948 \providetranslation{Exercise}{练习}
4949 \providetranslation{exercise}{练习}
4950 \providetranslation{Facts}{事实}
4951 \providetranslation{facts}{事实}
4952 \providetranslation{Fact}{事实}
4953 \providetranslation{fact}{事实}
4954 \providetranslation{Key Lemmas}{关键引理}
4955 \providetranslation{key lemmas}{关键引理}
4956 \providetranslation{Key Lemma}{关键引理}
4957 \providetranslation{key lemma}{关键引理}
4958 \providetranslation{Key Observations}{关键观察}
4959 \providetranslation{key observations}{关键观察}
4960 \providetranslation{Key Observation}{关键观察}
4961 \providetranslation{key observation}{关键观察}
4962 \providetranslation{Lemmas}{引理}
4963 \providetranslation{lemmas}{引理}
4964 \providetranslation{Lemma}{引理}
4965 \providetranslation{lemma}{引理}
4966 \providetranslation{Main Theorems}{主要定理}

```

```

4967 \providetranslation{main theorems}{主要定理}
4968 \providetranslation{Main Theorem}{主要定理}
4969 \providetranslation{main theorem}{主要定理}
4970 \providetranslation{Observations}{观察}
4971 \providetranslation{observations}{观察}
4972 \providetranslation{Observation}{观察}
4973 \providetranslation{observation}{观察}
4974 \providetranslation{Problems}{问题}
4975 \providetranslation{problems}{问题}
4976 \providetranslation{Problem}{问题}
4977 \providetranslation{problem}{问题}
4978 \providetranslation{Proofs}{证明}
4979 \providetranslation{proofs}{证明}
4980 \providetranslation{Proof}{证明}
4981 \providetranslation{proof}{证明}
4982 \providetranslation{Proof Sketch}{证明提要}
4983 \providetranslation{Proof sketch}{证明提要}
4984 \providetranslation{proof sketch}{证明提要}
4985 \providetranslation{Proof Sketches}{证明提要}
4986 \providetranslation{Proof sketches}{证明提要}
4987 \providetranslation{proof sketches}{证明提要}
4988 \providetranslation{Sketch of Proof}{证明提要}
4989 \providetranslation{Sketch of Proofs}{证明提要}
4990 \providetranslation{Sketch of proof}{证明提要}
4991 \providetranslation{Sketch of proofs}{证明提要}
4992 \providetranslation{sketch of proof}{证明提要}
4993 \providetranslation{sketch of proofs}{证明提要}
4994 \providetranslation{Propositions}{命题}
4995 \providetranslation{propositions}{命题}
4996 \providetranslation{Proposition}{命题}
4997 \providetranslation{proposition}{命题}
4998 \providetranslation{Remarks}{注}
4999 \providetranslation{remarks}{注}
5000 \providetranslation{Remark}{注}
5001 \providetranslation{remark}{注}
5002 \providetranslation{Solutions}{解}
5003 \providetranslation{solutions}{解}
5004 \providetranslation{Solution}{解}
5005 \providetranslation{solution}{解}
5006 \providetranslation{Theorems}{定理}
5007 \providetranslation{theorems}{定理}
5008 \providetranslation{Theorem}{定理}
5009 \providetranslation{theorem}{定理}
5010 </theorem>
5011 </dict>

```

14.17 ctexcap 宏包

```
5012 <*ctexcap>
```

ctexcap 是过时宏包。

```

5013 \clist_new:N \l__ctex_ctexcap_options_clist
5014 \clist_set:Nx \l__ctex_ctexcap_options_clist
5015 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
5016 \msg_new:nnn { ctexcap } { deprecated }
5017 {
5018   Package~ctexcap'~is~deprecated.\\
5019   Please~use~package~ctex'~with~option~`#1'~instead: \\\
5020   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5021 }
5022 \msg_warning:nnx { ctexcap } { deprecated }
5023 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

5024 \PassOptionsToPackage { heading = true } { ctexcap }
5025 \RequirePackageWithOptions { ctex }

```

5026 </ctexcap>

14.18 ctexhook 宏包

5027 <*ctexhook>

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

5028 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
5029 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
5030 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
5031 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
5032 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
5033 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5034 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
5035 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5036 \cs_set_nopar:Npx \document
5037 {
5038   \CTEX@document@left@hook
5039   \exp_not:o { \document }
5040   \CTEX@document@right@hook
5041 }
5042 \tl_new:N \g__ctex_end_preamble_hook_tl
5043 \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。

```

5044 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
5045 {
5046   \@ifpackageloaded {#1}
5047   {#2}
5048   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
5049 }

```

给钩子附加内容。

```

5050 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
5051 {
5052   \tl_if_exist:NF #1 { \tl_new:N #1 }
5053   \tl_gput_right:Nn #1 {#2}
5054 }
5055 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

宏包末尾钩子, 只执行一次, 用后清除。

```

5056 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
5057 {
5058   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
5059   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
5060 }
5061 \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。

```

5062 \tl_put_left:Nn \@popfilename
5063 {
5064   \cs_if_eq:NNT \@currentx \pkgextension
5065   { \ctex_package_end_hook:o { \@currname } }
5066 }

```

5067 </ctexhook>

14.19 ctexpatch 宏包

5068 <*ctexpatch>

```
5069 \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。

```
5070 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
5071 {
5072   \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnnw #1
5073   { once } {#2} { \use_i:nn } { \use_ii:nn }
5074 }
```

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

```
5075 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5076 {
5077   \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnnw #1
5078   { all } {#2} { \use_i:nn } { \use_ii:nn }
5079 }
```

`\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。

```
5080 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5081 {
5082   \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnnw #1
5083   { once }
5084   {
5085     \ExplSyntaxOff
5086     \char_set_catcode_letter:n { 64 }
5087   }
5088   { }
5089   { \ctex_patch_failure:N #1 }
5090 }
5091 \cs_new_protected:Npn \ctex_patch_failure:N #1
5092 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5093 \msg_new:nnn { ctex } { patch-failure }
5094 { Oops!~Command~`#1'~is~NOT~patchable.\\ }
```

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

```
5095 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5096 {
5097   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
5098   { left } {#2} { \use_i:nn } { \use_ii:nn }
5099 }
```

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

```
5100 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5101 {
5102   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
5103   { right } {#2} { \use_i:nn } { \use_ii:nn }
5104 }
```

`\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。

```
5105 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
5106 {
5107   \tl_set:Nn \__ctex_patch_true:w {#5}
5108   \tl_set:Nn \__ctex_patch_false:w {#6}
5109   \group_begin:
5110     \char_set_catcode_other:n { 35 }
5111     \ctex_parse_name:NN #1 #2 {#3} {#4}
5112   }
5113   \tl_new:N \__ctex_patch_true:w
5114   \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。

```

5115 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5116 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5117 \group_begin:
5118 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5119 {
5120   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5121   {
5122     \bool_lazy_or:nnTF
5123     { \cs_if_exist_p:c { ##3 ~ } }
5124     { \cs_if_exist_p:c { #1##3 } }
5125     {
5126       \group_begin:
5127       \use:x
5128       {
5129         \group_end:
5130         \__ctex_parse_name:nnNNn
5131         { \token_get_replacement_spec:N ##2 }
5132         \exp_not:N ##2
5133         \exp_not:c { ##3 ~ }
5134         \exp_not:c { #1##3 }
5135       } {##3} ##1
5136     }
5137     { ##1##2 }
5138   }
5139   \cs_new_protected:Npn \__ctex_parse_name:nnNNn ##1##2##3##4##5##6
5140   {
5141     \exp_args:Nc ##6
5142     {
5143       \str_case:nnTF {##1}
5144       {
5145         { \protect ##3 } { }
5146         { \x@protect ##2 \protect ##3 } { }
5147       }
5148       {
5149         \str_if_eq_x:nnTF
5150         { \exp_not:n { #1@protected@ ##3 #1##3 } }
5151         {
5152           \exp_last_unbraced:Nf \__ctex_parse_name:w
5153           \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
5154         }
5155         { #1##5 ~ } { ##5 ~ }
5156       }
5157     }
5158     \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5159     {
5160       { #1@protected@ ##2 ##4 } { }
5161       { #1@ ##4 } { }
5162     }
5163     { #1##5 } {##5}

```

```

5164     }
5165   }
5166 }
5167 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5168 }
5169 \use:x
5170 {
5171   \__ctex_tmp:w
5172   { \c_backslash_str }
5173   { \c_left_brace_str }
5174   { \tl_to_str:n { testopt } }
5175 }
5176 \group_end:
5177 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

```

分别保存宏的 `\meaning` 中的前缀、参数文本和替换文本。

```

5178 \str_new:N \l__ctex_prefix_str
5179 \str_new:N \l__ctex_parameter_str
5180 \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。

```

5181 \group_begin:
5182 \cs_set_protected:Npn \__ctex_tmp:w #1
5183 {
5184   \prg_new_protected_conditional:Npnn
5185   \ctex_get_macro_meaning:N ##1 { TF }
5186   {
5187     \exp_after:wN \__ctex_get_macro_meaning:w
5188     \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5189   }
5190   \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5191   ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5192   {
5193     \tl_if_empty:nTF { ##4 }
5194     { \prg_return_false: }
5195     {
5196       \str_set:Nn \l__ctex_prefix_str { ##1 }
5197       \str_set:Nn \l__ctex_parameter_str { ##2 }
5198       \str_set:Nn \l__ctex_replacement_str { ##3 }
5199       \prg_return_true:
5200     }
5201   }
5202 }
5203 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5204 \group_end:

```

```

\ctex_if_rescanable:NnTF

```

检查宏是否可以重建。

```

5205 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5206 {
5207   \ctex_get_macro_meaning:NTF #1
5208   {
5209     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5210     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5211   }
5212   {#4}
5213 }
5214 \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。

```

5215 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5216 {
5217   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
5218   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}

```

```

5219 \__ctex_patch_rescan:NnN \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5220 \use:x
5221 {
5222   \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5223   \exp_not:o { \l__ctex_parameter_tl }
5224   { \exp_not:o { \l__ctex_replacement_tl } }
5225 }
5226 }
5227 \cs_new_protected:Npn \__ctex_patch_rescan:NnN #1#2#3
5228 {
5229   \str_if_empty:NTF #2
5230   { \tl_clear:N #1 }
5231   { \tl_set_rescan:Nno #1 {#3} {#2} }
5232 }
5233 \tl_new:N \l__ctex_prefix_tl
5234 \tl_new:N \l__ctex_parameter_tl
5235 \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。

```

5236 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5237 {
5238   \group_end:
5239   \ctex_if_rescanable:NnTF #1 {#3}
5240   {
5241     \use:x
5242     {
5243       \__ctex_patch_replace:nnnTF {#2}
5244       { \tl_to_str:n {#4} }
5245       { \tl_to_str:n {#5} }
5246     }
5247     {
5248       \__ctex_patch_rebuild:Nn #1 {#3}
5249       \__ctex_patch_true:w
5250     }
5251     { \__ctex_patch_false:w }
5252   }
5253   { \__ctex_patch_false:w }
5254 }

```

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

```

5255 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5256 {
5257   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5258   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5259 }

```

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

5260 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5261 {
5262   \group_end:
5263   \ctex_get_macro_meaning:NTF #1
5264   {
5265     \str_if_empty:NTF \l__ctex_parameter_str
5266     { \__ctex_hookto_cmd_parameterless:Nnnnw }
5267     { \__ctex_hookto_cmd_parameter:Nnnnw }
5268     #1 {#2} {#3} {#4}
5269   }
5270   { \__ctex_patch_false:w }
5271 }

```

__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。

```

5272 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5273 {

```

```

5274 \str_if_empty:NF \l__ctex_prefix_str
5275 { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5276 \tex_edef:D #1
5277 {
5278   \use:c { __ctex_ #2 _hook_aux:nn }
5279   { \exp_not:o {#1} }
5280   { \exp_not:n {#4} }
5281 }
5282 \__ctex_patch_true:w
5283 }
5284 \cs_generate_variant:Nn \tl_rescan:nn { no }
5285 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5286 \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。

```

5287 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5288 {
5289   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5290   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5291   {
5292     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5293     \__ctex_patch_rebuild:Nn #1 {#3}
5294     \__ctex_patch_true:w
5295   }
5296   { \__ctex_patch_false:w }
5297 }
5298 </ctexpatch>

```

版本历史

v2.0	(2014/03/06 – 2015/05/06)	将章节标题设置功能提取到可以独立使用的宏包 <code>ctexheading</code> 中。 1
General: <code>c5size</code> , <code>cs4size</code> 是过时选项。 37		新的标题格式选项 <code>aftertitle</code> 。 83
<code>captiondelimiter</code> 是过时选项。 110		修复 <code>ctexbook</code> 和 <code>ctexrep</code> 类的中文 <code>part/number</code> 选项初值为空的错误。 101
<code>fancyhdr</code> 成为过时选项, 原选项功能总是打开。 40		<code>\ctex_if_macosx:TF</code> : 改用 <code>/Library/Fonts/Songti.ttc</code> 为特征文件。 70
<code>fntef</code> 成为过时选项, 原选项功能总是打开。 40		<code>hyperref</code> : 补充定义 <code>\hypersetup</code> 。 40
<code>hyperref</code> 成为过时选项, 原选项功能总是打开。 40		v2.2 (2015/06/21 – 2015/06/30)
<code>indent</code> , <code>noindent</code> 是过时选项。 38		General: <code>before skip</code> 和 <code>after skip</code> 选项的符号不再有特殊意义。 83
<code>nofonts</code> , <code>adobe fonts</code> , <code>win fonts</code> 是过时选项。 38		<code>before skip</code> , <code>after skip</code> 和 <code>indent</code> 选项支持表达式。 86
<code>nopunct</code> 是过时选项。 39		不再依赖 <code>etoolbox</code> 宏包。 1
<code>nospace</code> 是过时选项。 40		非 <code>ctexart</code> 类的 <code>part/before skip</code> 和 <code>part/after skip</code> 选项有意义。 86
<code>nozhmap</code> 是过时选项。 39		给 <code>enumitem</code> 宏包注册 <code>\chinese</code> 和 <code>\zhnum</code> 。 81
<code>punct</code> 选项可以设置标点格式。 39		将文档开头和宏包末尾钩子提取到 <code>ctexhook</code> 宏包中。 131
<code>ctex</code> 宏包新增 <code>heading</code> 选项。 40		将中文版式下的 <code>part</code> 和 <code>chapter</code> 标题的 <code>nameformat</code> 和 <code>titleformat</code> 选项的初值合并到 <code>format</code> 中。 101
<code>\CTEXindent</code> , <code>\CTEXnoindent</code> 是过时命令。 80		删去 <code>etoolbox</code> 与 <code>breqn</code> 的兼容补丁。 111
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。 14, 77		新的标题格式选项 <code>afterindent</code> 。 83
<code>\CTEXunderdot</code> , <code>\CTEXunderline</code> , <code>\CTEXunderdblline</code> , <code>\CTEXunderwave</code> , <code>\CTEXsout</code> , <code>\CTEXxout</code> 是过时命令; <code>CTEXfilltwosides</code> 是过时环境。 72		新的标题格式选项 <code>numbering</code> 。 83
标题设置新增 <code>pagestyle</code> 选项。 84		新的标题格式选项 <code>runin</code> 。 83
调整 <code>\footnotesep</code> 的大小, 以适合行距的变化。 110		新增子宏包 <code>ctexpatch</code> 实现给宏打补丁功能。 131
兼容 <code>extsizes</code> 宏包、 <code>beamer</code> 、 <code>memoir</code> 等提供的更多字号选项。 41		v2.3 (2015/09/17 – 2016/01/05)
将标题汉化功能加入 <code>ctex.sty</code> 。 81		General: <code>.value_required:</code> 和 <code>.value_forbidden:</code> 已过时。 37
将中文字号功能提取到可以独立使用的 <code>ctexsize</code> 。 107		代码实现避免使用 <code>\lowercase</code> 技巧 (Joseph Wright)。 36
解决 <code>etoolbox</code> 与 <code>breqn</code> 关于 <code>\end</code> 的冲突。 111		更新 <code>LuaTeX-j</code> 支持 (20150922.0)。 48
默认关闭 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 的彩色设置。 72		更新 <code>unicode-math</code> 宏包补丁。 49
删除 <code>c19gsn.fd</code> 和 <code>c19gkai.fd</code> 。 1		兼容 <code>titleps</code> 宏包。 96
通过 <code>LuaTeX-j</code> 宏包支持 <code>Lua\LaTeX</code> 。 48		修复 <code>nameformat</code> 作用域问题。 83
新增 <code>autoindent</code> 选项。 38		与 <code>LaTeX3</code> (2015/12/20) 同步。 34
新增 <code>fontset</code> 选项。 38		v2.4 (2015/02/19 – 2016/04/25)
新增 <code>linespread</code> 选项。 38		General: 初步支持 <code>up\LaTeX</code> 。 67
新增 <code>linestretch</code> 选项。 78		加强 <code>beamer</code> 宏包支持。 1
新增 <code>scheme</code> 选项, 并将 <code>cap</code> 和 <code>nocap</code> 列为过时选项。 40		提供 <code>translator</code> 宏包的中文定理名称翻译。 129
新增 <code>zhmCJK</code> 支持选项。 39		提供 <code>up\LaTeX</code> 的 NFSS 字体定义。 113
新增 <code>zihao</code> 选项。 37		正确更新 <code>CJK</code> 包的 <code>\CJKfamilydefault</code> 。 69
新增统一设置接口 <code>\ctexset</code> 。 77		正确设置 <code>up\TeX</code> 下字体命令。 68
应用 <code>LaTeX3</code> 重新整理代码。 1		<code>\ctex_parse_name:NN</code> : 修复宏名解析错误。 133
中文字号不再采用近似值。 107		v2.4.1 (2016/04/26 – 2016/05/14)
自动检测操作系统, 载入对应的字体配置。 69		General: <code>beamer</code> 不调整默认行距。 106
<code>\CJK@surr</code> : 解决与 <code>\nouppercase</code> 的冲突。 45		<code>beamer</code> 不调整默认字体大小。 106
v2.0.1 (2015/05/15)		使用 <code>bootfont.bin</code> 判断 Windows XP 以避免权限问题。 114
General: 修复 10pt、11pt 等选项无效的问题。 41		随字体更新 <code>up\TeX</code> 的 <code>\xkanjiskip</code> 。 75
v2.0.2 (2015/05/16)		新的标题格式选项 <code>part/fixbefore skip</code> 和 <code>chapter/fixbefore skip</code> 。 84
General: 修复加载 <code>ctex</code> 宏包后章节标题后第一段无段首缩进的问题。 107		正确更新 <code>up\LaTeX</code> 的 <code>\CJKfamilydefault</code> 。 69
v2.1 (2015/05/18 – 2015/06/19)		<code>\ccwd</code> : 正确设置 <code>up\TeX</code> 下的 <code>\ccwd</code> 。 73
General: <code>format+</code> , <code>nameformat+</code> 等带加号的选项, 加号与前面的文字之间可以有可选的空格。 83		<code>\chinese</code> : 支持 <code>\pagenumbering</code> 。 81
<code>nameformat</code> 可以接受章节名字为参数。 83		<code>zihao</code> : 不允许无参 <code>zihao</code> 选项。 37
不依赖 <code>ifpdf</code> 宏包。 36		
不再设置 <code>hyperref</code> 宏包的 <code>colorlinks</code> 选项。 71		
给 <code>pdf\LaTeX</code> 下的非 UTF8 编码 CJK 字体族加上 <code>CMap</code> 。 42		

v2.4.2	(2016/05/15)	v2.4.6	(2016/10/31 – 2016/11/20)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	49	General: <code>\CTEXifname</code> 初始为假。	85
<code>\em</code> : 兼容 <code>upL^AT_EX 2016/05/07u00</code> 的定义。	68	支持字体属性可选项在后的新语法。	59
v2.4.3	(2016/06/03 – 2016/08/26)	重新初始化 <code>\ifthechapter</code> 等。	97
General: 更新 <code>unicode-math</code> 补丁。	54	v2.4.7	(2016/12/23 – 2016/12/27)
简化 <code>fontspec</code> 补丁。	49	General: 依赖 <code>pxeverysel</code> 宏包。	69
确保 <code>\proofname</code> 非空。	81	<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	98
删除选项 <code>part/fixbeforeskip</code> 和 <code>chapter/fixbeforeskip</code> 。	84	v2.4.8	(2017/02/23)
新的标题格式选项 <code>fixskip</code> 。	83	<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。	60
v2.4.4	(2016/09/09 – 2016/09/19)	v2.4.9	(2017/02/27)
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	48	General: 调整 <code>unicode-math</code> 补丁的代码顺序。	54
改进 <code>hyperref</code> 宏包的标题锚点设置。	93	v2.4.10	(2017/07/19 – 2017/07/23)
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	44	General: 常数 <code>\c_minus_one</code> 已过时。	34
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	95	定义 <code>\cht</code> , <code>\cdp</code> 和 <code>\c wd</code> 。	76
提供 <code>\CTEXifname</code> 。	85	使用 <code>lazy</code> 函数对 <code>Boolean</code> 表达式进行最小化运算 (<code>L^AT_EX3 2017/07/19</code>)。	34
提供 <code>\partmark</code> 。	85	v2.4.11	(2017/08/17 – 2017/11/21)
新的标题格式选项 <code>break</code> 。	83	General: 不把 <code>Enclosed Alphanumerics</code> 设置为 <code>J_Achar</code> 。	48
v2.4.5	(2016/10/01 – 2016/10/25)	不把希腊和西里尔字母设置为 <code>J_Achar</code> 。	48
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	84	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。	83
新的标题格式选项 <code>hang</code> 。	83	<code>\ps@headings</code> : 补充页眉空格。	98
新的标题格式选项 <code>tocline</code> 。	83		
<code>\ps@headings</code> : 修复补丁失败。	98		

代码索引

意大利体的数字表示描述对应索引项的页码; 带下划线的数字表示定义对应索引项的代码行号; 罗马字体的数字表示使用对应索引项的代码行号。

Symbols	
<code>_</code>	509, 3240, 3247, 3263, 3266
<code>\#</code>	475, 3141, 3200
<code>.../afterindent</code>	20
<code>.../aftername</code>	18
<code>.../aftername+</code>	18
<code>.../afterskip</code>	20
<code>.../aftertitle</code>	19
<code>.../aftertitle+</code>	19
<code>.../beforeskip</code>	20
<code>.../break</code>	22
<code>.../break+</code>	22
<code>.../fixskip</code>	20
<code>.../format</code>	16
<code>.../format+</code>	16
<code>.../indent</code>	21
<code>.../name</code>	15
<code>.../nameformat</code>	17
<code>.../nameformat+</code>	17
<code>.../number</code>	15
<code>.../numberformat</code>	17
<code>.../numberformat+</code>	17
<code>.../titleformat</code>	18
<code>.../titleformat+</code>	18
<code>.../tocline</code>	22
<code>\/</code>	1644
<code>\</code>	15, 33, 34, 102, 145, 147, 149, 206, 216, 673, 1102, 1184, 2324, 3620, 3621, 3734, 3735, 3955, 3985, 5018, 5019, 5020, 5094
<code>\{</code>	1113, 1114, 5020
<code>\}</code>	1113, 1114, 5020
10pt	8, 367
11pt	8, 367
12pt	8, 367
A	
<code>\abovedisplayshortskip</code>	3825, 3833, 3844, 3864, 3872, 3883
<code>\abovedisplayskip</code>	3824, 3827, 3832, 3839, 3843, 3850, 3863, 3866, 3871, 3878, 3882, 3889
<code>\abstractname</code>	2308
<code>abstractname</code>	12
<code>\addCJKfontfeature</code>	1184, 1228
<code>\addCJKfontfeatures</code>	1201
<code>\addcontentsline</code>	2526, 2572, 2576, 2581, 2609, 2613, 2618, 2707, 2712, 2718, 2724, 2847, 2861

- \AddEnumerateCounter 2263, 2265, 2266, 2267
 - \addpenalty 2806, 3460, 3493, 3527, 3541, 3555
 - \addtocontents 2514, 2520, 2728, 2729
 - \addvspace 2515, 2521, 2533, 2536,
2728, 2729, 2806, 2810, 3279, 3289, 3298, 3308, 3317, 3327
 - adobefonts 29
 - \algorithmname 2315, 2331
 - algorithmname 12
 - AlternateFont 26, 1340
 - \appendix 2988
 - appendix/name 23, 2977
 - appendix/number 23, 2977
 - appendix/numbering 23, 2977
 - \appendixname 2310, 3605
 - appendixname 12
 - \arabic 3507
 - \AtBeginDocument 576, 1633
 - \AtBeginDvi 433
 - \AtBeginShipoutFirst 435, 1836, 1851
 - \AtBeginUTFCommand 717
 - \AtEndOfClass 135
 - \AtEndOfPackage 136
 - \AtEndUTFCommand 723
 - autoindent 10, 185, 2118
- B**
- \baselinestretch 2173
 - \begin 3285, 3304, 3323
 - \begingroup 2838, 2874, 2906, 3274, 3293, 3312, 4518, 4551, 4558
 - \belowdisplayshortskip . 3826, 3834, 3845, 3865, 3873, 3884
 - \belowdisplayskip 3827, 3839, 3850, 3866, 3878, 3889
 - \bfdefault 964, 970, 1330, 1337
 - \bfseries 2588, 2591, 2627, 2630, 2643,
2660, 2756, 2761, 2779, 3388, 3390, 3394, 3407, 3408,
3411, 3436, 3438, 3444, 3462, 3466, 3487, 3521, 3538, 3552
 - \bibname 2313, 2316, 2324, 2332, 2338, 2339
 - bibname 12
 - bool commands:
 - \bool_gset_false:N 266, 271
 - \bool_gset_true:N 261
 - \bool_if:NTF 70, 1142, 2920, 3082, 3092, 3103, 3624
 - \bool_lazy_or:nnTF 5122
 - \bool_new:N 131, 1179
 - \bool_set_false:N 1154
 - \bool_set_true:N 1172
 - \c_false_bool 2936, 2938
 - \c_true_bool 2935, 2937
 - box commands:
 - \box_ht:N 3905
 - \box_new:N 60
 - \box_wd:N 1922, 3196
- C**
- c5size 29
 - cap 29
 - captiondelimiter 30, 3912
 - \catcode 4518, 4519, 4520, 4521, 4525, 4526
 - CCTF 29
 - CCTFont 29
 - \ccwd 25, 115, 189, 1918, 1968,
1969, 1970, 2124, 2162, 2181, 2182, 2189, 2216, 2217,
2218, 2221, 2223, 2224, 2235, 2246, 2247, 3922, 3923, 3924
 - \cdp 2034
 - \centering 2624, 2658, 3275,
3294, 3313, 3394, 3406, 3411, 3415, 3444, 3466, 3474, 3509
 - \chapter 2684, 3630
 - chapter/beforeskip 31
 - chapter/fixbeforeskip 31
 - chapter/lofskip 21, 2455
 - chapter/lotskip 21, 2455
 - chapter/numbering 15
 - chapter/pagestyle 21, 2455
 - \chaptermark 2727
 - \chaptername 3433
 - char commands:
 - \char_set_catcode:nn 73, 513, 514, 549, 550
 - \char_set_catcode_active:N 1365
 - \char_set_catcode_comment:n 508
 - \char_set_catcode_escape:n 509
 - \char_set_catcode_letter:n 67, 1636, 3214, 5086
 - \char_set_catcode_math_superscript:n 544
 - \char_set_catcode_other:N . 475, 872, 1348, 3141, 3200
 - \char_set_catcode_other:n 543, 5110
 - \char_set_catcode_space:n 2040
 - \char_value_catcode:n 73, 513, 514, 549, 550
 - CharRange 26, 1340
 - \Chinese 30, 2259
 - \chinese 25, 2257, 2267, 3381, 3443
 - \cht 2034
 - \CJK 1887
 - \CJKbold
 - 4035, 4036, 4046, 4047, 4061, 4062, 4064, 4065, 4067,
4068, 4072, 4073, 4075, 4076, 4078, 4079, 4091, 4092,
4094, 4095, 4097, 4098, 4102, 4103, 4105, 4106, 4108, 4109
 - \CJKfamily 523,
527, 531, 535, 1036, 1059, 1276, 1283, 1284, 1285,
1286, 1434, 1668, 4506, 4507, 4508, 4509, 4511, 4512, 4514
 - \CJKfamilydefault 517,
518, 535, 591, 1263, 1264, 1276, 1286, 1294, 1296, 1300,
1307, 1320, 1651, 1665, 1673, 1726, 1730, 1738, 1751, 1753
 - \CJKfilltwosides 1895
 - \CJKfontspec 1201
 - \CJKglue 1921, 1935, 1952, 1965
 - \CJKhook 606
 - \CJKnormal 4028, 4029, 4030, 4031, 4032, 4033, 4034,
4039, 4040, 4041, 4042, 4043, 4044, 4045, 4060, 4063,
4066, 4071, 4074, 4077, 4090, 4093, 4096, 4101, 4104, 4107
 - \CJKpunctmapfamily
 - . 639, 640, 641, 642, 643, 644, 648, 649, 650, 651, 652, 653
 - \CJKrmdefault 498,
518, 519, 523, 1109, 1120, 1232, 1264, 1265, 1283, 1296,
1299, 1300, 1648, 1651, 1653, 1734, 4158, 4160, 4161,
4166, 4167, 4168, 4171, 4248, 4251, 4252, 4257, 4258,

- 4259, 4326, 4329, 4330, 4336, 4337, 4338, 4407, 4410,
4411, 4418, 4419, 4420, 4468, 4471, 4472, 4477, 4478, 4479
- \CJKsfdefault 499, 520,
527, 1110, 1121, 1239, 1266, 1284, 1649, 1657, 1735,
4141, 4148, 4149, 4172, 4249, 4327, 4331, 4408, 4412, 4469
- \CJKtilde 592
- \CJKttdefault 500, 521, 531, 1111, 1122, 1245, 1267,
1285, 1650, 1661, 1736, 4159, 4173, 4250, 4328, 4409, 4470
- clearalternatefont 27, 1526
- \cleardoublepage 2547, 2685, 3404, 3431
- \clearpage 2549, 2685, 3404, 3431
- clist commands:
- \clist_clear:N 985, 4855
- \clist_concat:NNN 1005
- \clist_const:Nn 619
- \clist_gput_right:Nn 383, 395, 405, 407
- \clist_gset:Nn 1255
- \clist_if_empty:NTF 1353, 1372, 1612
- \clist_map_break:n 3061, 3615
- \clist_map_function:nN 1529, 1531, 1563, 1577
- \clist_map_inline:Nn 609, 1465
- \clist_map_inline:nn
. 368, 631, 1570, 1879, 1905, 3056, 3612, 3752, 3787, 4844
- \clist_new:N 138, 1002, 1256, 1407, 4873, 5013
- \clist_put_left:Nn 1006, 3681, 3707
- \clist_put_right:Nn 1171, 1453, 4867
- \clist_set:Nn 989, 1390, 1391, 5014
- \clist_use:Nn 5023
- \clubpenalty 2905, 2912
- \contentsname 2303
- contentsname 11
- continuation 12
- cs commands:
- \cs:w 3352
- \cs_end: 3352
- \cs_generate_variant:Nn 93,
451, 463, 949, 1071, 1085, 1086, 1087, 1160, 1180,
1181, 1386, 1423, 1525, 1563, 1571, 5055, 5061, 5177, 5284
- \cs_gset_eq:NN
..... 162, 163, 164, 262, 267, 272, 310, 312, 378,
382, 483, 767, 771, 1147, 1150, 3025, 3026, 3031, 3687, 3714
- \cs_gset_nopar:Npx 768
- \cs_gset_protected_nopar:Npn 2989, 3037
- \cs_gset_protected_nopar:Npx
..... 806, 1032, 1427, 1521, 2965
- \cs_if_eq:NNTF 908, 3023, 5064, 5210, 5290
- \cs_if_exist:NTF 44, 359, 460, 762, 774, 804,
842, 927, 1426, 1518, 1998, 3127, 3339, 3630, 3632, 5069
- \cs_if_exist_p:N 5123, 5124
- \cs_if_exist_use:N 1055, 1056
- \cs_if_exist_use:NTF 4, 745, 1077, 1540, 5058
- \cs_if_free:NTF 445, 455, 611, 626,
1144, 1329, 2137, 2151, 2263, 3075, 3120, 3166, 3171, 3175
- \cs_new:Npn 110, 117, 124,
863, 869, 873, 981, 1058, 2939, 2947, 2974, 3100, 5167, 5285
- \cs_new_eq:NN . 6, 7, 8, 9, 20, 23, 95, 106, 107, 108, 133,
134, 360, 452, 596, 657, 783, 808, 884, 915, 1228, 1235,
1747, 1900, 1901, 1902, 1914, 1965, 1976, 1982, 2258,
2259, 2260, 2500, 2503, 2933, 2934, 2935, 2936, 2937,
2938, 2988, 3013, 3014, 3357, 3628, 3633, 4849, 5214, 5286
- \cs_new_nopar:Npn
823, 1050, 1105, 1116, 2257, 2437, 2955, 3164, 3169, 4874
- \cs_new_protected:Npn 62,
917, 982, 1187, 1351, 1442, 1456, 1607, 1971, 2448,
2525, 3190, 3807, 3813, 4850, 5028, 5030, 5044, 5050,
5070, 5075, 5080, 5091, 5095, 5100, 5105, 5115, 5120,
5139, 5190, 5205, 5215, 5227, 5236, 5255, 5260, 5272, 5287
- \cs_new_protected_nopar:Npn
..... 77, 79, 135, 136, 431, 443, 453, 458, 464,
496, 504, 539, 556, 598, 604, 607, 624, 629, 637, 646,
655, 743, 752, 765, 770, 772, 784, 798, 825, 840, 853,
877, 886, 903, 942, 959, 1003, 1011, 1030, 1051, 1061,
1088, 1126, 1138, 1140, 1162, 1164, 1273, 1279, 1290,
1315, 1324, 1370, 1381, 1408, 1417, 1424, 1430, 1450,
1463, 1482, 1496, 1503, 1515, 1535, 1551, 1569, 1572,
1574, 1587, 1599, 1641, 1686, 1699, 1713, 1724, 1756,
1774, 1789, 1805, 1816, 1820, 1857, 1864, 1868, 1891,
1897, 1918, 1932, 1967, 1983, 2013, 2058, 2136, 2138,
2163, 2198, 2208, 2213, 2367, 2397, 2484, 2491, 2498,
2504, 2506, 2509, 2918, 2924, 3005, 3016, 3048, 3068,
3073, 3078, 3088, 3098, 3142, 3335, 3344, 3349, 3725,
3740, 3772, 3777, 3927, 3931, 4841, 4865, 5032, 5034, 5056
- \cs_new_protected_nopar:Npx ... 468, 1211, 1884, 3129
- \cs_set:Npn 1349, 1979, 2424
- \cs_set_eq:NN 109, 190, 195, 200, 536, 554,
574, 601, 605, 613, 627, 777, 778, 885, 916, 934, 979,
1036, 1037, 1426, 1434, 1444, 1445, 1446, 1518, 1519,
1520, 1644, 1666, 1729, 2274, 2279, 2282, 2287, 2505,
2507, 2926, 2927, 2928, 2929, 2930, 2931, 3020, 3052, 3062
- \cs_set_nopar:Npn 1685, 3181
- \cs_set_nopar:Npx 1915, 3090, 5036
- \cs_set_protected:Npn 724, 920, 5118, 5182
- \cs_set_protected_nopar:Npn
..... 600, 935, 1935, 3122, 3179, 3814
- \cs_set_protected_nopar:Npx 1973
- \cs_to_str:N 53, 1210, 5116
- \cs_undefine:N . 1015, 1016, 1020, 1021, 1447, 1987, 5059
- cs4size 29
- \csname 2807, 2816, 2817, 2818, 2825, 2843, 2844, 2846, 2849,
2858, 2859, 2860, 2863, 4550, 4552, 4553, 4554, 4559, 4563
- \CTeX 25, 3908
- ctex commands:
- \ctex_add_cmap:n 448, 453
- \ctex_add_cmap:Nn 456, 458, 463
- \ctex_add_to_selectfont:n 1971, 2004, 2032, 2060, 2150
- \ctex_after_end_preamble:n 36, 109, 5028
- \ctex_appto_cmd:NnnTF 479, 1635, 5100
- \ctex_assign_heading_name:nn 2403, 2445, 2981
- \ctex_at_end:n 135, 294, 357, 1965, 3994
- \ctex_at_end_package:nn
131, 434, 482, 732, 925, 1830, 1832, 1850, 2261, 3035,

- 3108, 3115, 3174, 3176, 3177, 3211, 3251, 3363, 3910, 5044
 \ctex_at_end_preamble:n 362, 581, 588, 1289, 1674, 5028
 \ctex_auto_ignorespaces: 595, 605, 2070
 \ctex_CJK_input:n 539
 \ctex_declare_math_sizes:nnnn 3772, 3806
 \ctex_default_pt:n 113, 117
 \ctex_detected_platform: 1755, 3935, 3944
 \ctex_family_cmap:nn 443, 452
 \ctex_file_input:n 77, 398,
 658, 2100, 3639, 3674, 3679, 3684, 3685, 3700, 3705,
 3711, 3712, 3816, 3818, 3936, 3940, 3948, 3994, 4119, 4120
 \ctex_file_wrapper:nnn 62, 78, 81, 506, 541
 \ctex_fix_varioref_label:n 3347, 3349
 \ctex_fontset_error:n
 ... 3927, 4227, 4261, 4295, 4340, 4376, 4379, 4422, 4481
 \ctex_gadd_hook:Nn 606, 5048, 5050
 \ctex_get_macro_meaning:N 5185
 \ctex_get_macro_meaning:NTF 5181, 5207, 5263
 \ctex_hypersetup:n
 71, 360, 1813, 1825, 1828, 1843, 1846, 1849
 \ctex_if_autoindent_touched:TF 133, 190, 195, 200, 3666
 \ctex_if_ccglue_touched:TF 1949, 2169
 \ctex_if_ccglue_touched_p: 1949
 \ctex_if_macosx:TF 1796, 1797, 1799, 1805
 \ctex_if_preamble:TF 108, 3963
 \ctex_if_rescanable:NnTF 5205, 5239
 \ctex_ignorespaces_case:N 598, 2067, 2070, 2073
 \ctex_load_fontset: 3931, 3971, 3993
 \ctex_load_zhmap:nnnn 496, 4164, 4255, 4334, 4416, 4475
 \ctex_ltj_add_font_features:n 1162, 1225
 \ctex_ltj_add_font_features:nn 1162, 1163
 \ctex_ltj_char_range_key:nn 1604, 1607
 \ctex_ltj_clear_alternate_font:n 1529, 1535
 \ctex_ltj_declare_alternate_shape:nnnnnn 1436, 1456
 \ctex_ltj_declare_char_range:n 1566, 1569, 1571
 \ctex_ltj_declare_char_range:nn 1573, 1574
 \ctex_ltj_def_char_range_key:n 1579, 1599
 \ctex_ltj_ensure_default_family: 1290, 1743
 \ctex_ltj_extract_font: 777, 784
 \ctex_ltj_family_if_exist:nNTF
 ... 1063, 1072, 1085, 1086, 1294, 1299, 1317, 1320, 1538
 \l_ctex_ltj_family_tl
 1065, 1070, 1129, 1130, 1145, 1149, 1152, 1163, 1532, 1533
 \ctex_ltj_fontspec:nn 1126, 1138, 1173, 1220
 \ctex_ltj_get_and_define_fonts:nN 877
 \ctex_ltj_get_and_define_fonts_al:nN 881, 884
 \ctex_ltj_get_and_define_fonts_ja:nN 880, 886
 \ctex_ltj_if_alternate_shape_exist:nTF
 787, 800, 818, 827
 \ctex_ltj_if_jfont:nTF 54, 863, 874, 879
 \ctex_ltj_if_jfont_math:NTF 869, 910
 \ctex_ltj_math_group_hook: 53, 54, 909, 915, 935
 \ctex_ltj_pickup_font: ... 758, 772, 847, 889, 891, 893
 \ctex_ltj_reset_alternate_font:n 1531, 1551
 \ctex_ltj_save_alternate_family:Nnnn 1401, 1421, 1424
 \ctex_ltj_save_alternate_seq:Nn 1376, 1381
 \ctex_ltj_save_alternate_seq:Nnnwn 1381
 \ctex_ltj_save_alternate_seq:Nnnwnw 1384, 1387
 \ctex_ltj_save_char_range:n 1577, 1584
 \ctex_ltj_save_char_range:nn 1586, 1587
 \ctex_ltj_select_alternate_font: 825, 2007
 \ctex_ltj_select_font: 743, 2006
 \ctex_ltj_set_alternate_family:nn 1379, 1408
 \ctex_ltj_set_alternate_family:nnn . 1420, 1428, 1430
 \ctex_ltj_set_alternate_family:Nnnnn 1412, 1417, 1423
 \ctex_ltj_set_alternate_family:nnTF 990, 1370
 \ctex_ltj_set_alternate_seq:n 1342, 1347
 \ctex_ltj_set_alternate_shape:n ... 1484, 1496, 1507
 \ctex_ltj_set_alternate_shape:nnN 1469, 1503
 \ctex_ltj_set_alternate_shape:nnn 1472, 1479
 \ctex_ltj_set_alternate_shape:nnnn 1481, 1482
 \ctex_ltj_set_alternate_shape:Nnnnnnn . 1459, 1463
 \ctex_ltj_set_family:nnn
 982, 1157, 1204, 1214, 1232, 1239, 1245, 1251
 \ctex_ltj_set_math_letter:NN 917, 929, 937
 \ctex_ltj_subst_font: 778, 798
 \ctex_ltj_swap_cs:NN 1435, 1442
 \ctex_ltj_switch_family:n
 1060, 1061, 1071, 1130, 1158, 1212
 \ctex_ltj_update_mathfont: 1312, 1315
 \ctex_ltj_update_mathfont:n 1318, 1321, 1324
 \ctex_ltj_use_math_group:Nn 903, 934
 \ctex_make_spa:nn 4841
 \ctex_mono_jfm:n 942, 950, 2090
 \ctex_package_end_hook:n 5056, 5065
 \ctex_parse_name:NN 1990, 5111, 5115
 \ctex_parse_name:NNn 5116, 5120, 5177
 \ctex_patch_boot:NNnnTF
 5072, 5077, 5082, 5097, 5102, 5105
 \ctex_patch_cmd:Nnn
 1681, 3144, 3150, 3156, 3223, 3226, 3230,
 3236, 3246, 3254, 3257, 3262, 3265, 3922, 3923, 3924, 5080
 \ctex_patch_cmd_all:NnnnTF 5075
 \ctex_patch_cmd_once:NnnnTF 1675, 5070
 \ctex_patch_failure:N 485, 525, 529, 533, 537,
 1639, 1655, 1659, 1663, 1667, 1995, 3207, 3217, 5089, 5091
 \ctex_plane_to_utfxvibe:Nn 555
 \ctex_preto_cmd:NnnTF 522, 526,
 530, 534, 1652, 1656, 1660, 1664, 1992, 3204, 3213, 5095
 \ctex_punct_map_bfseries:nn
 629, 4149, 4161, 4167, 4252,
 4258, 4330, 4331, 4337, 4411, 4412, 4413, 4419, 4472, 4478
 \ctex_punct_map_family:nn 624, 4141, 4148,
 4158, 4159, 4166, 4248, 4249, 4250, 4257, 4326, 4327,
 4328, 4336, 4407, 4408, 4409, 4418, 4468, 4469, 4470, 4477
 \ctex_punct_map_itshape:nn 646,
 4160, 4168, 4251, 4259, 4329, 4338, 4410, 4420, 4471, 4479
 \ctex_punct_map_series:nnn 633, 634, 637
 \ctex_punct_set:n 607,
 4136, 4165, 4247, 4256, 4325, 4335, 4406, 4417, 4467, 4476
 \ctex_punct_space:nn 655
 \ctex_scheme_input:n 79, 3365, 3641

- \ctex_select_size: .. 2125, 2132, 2135, 2136, 2157, 2211
- \ctex_set_default_ccwd:Nn 110, 199, 2131, 2156
- \ctex_set_font_size:Nnn 3807, 3822, 3830, 3841, 3852, 3853, 3854, 3855, 3856, 3857, 3858, 3861, 3869, 3880, 3891, 3892, 3893, 3894, 3895, 3896, 3897
- \ctex_set_ignorespaces: 598, 606
- \ctex_set_upfamily:nnn 1686, 4184, 4185, 4186, 4187, 4188, 4189, 4191, 4271, 4272, 4273, 4274, 4350, 4351, 4352, 4353, 4431, 4432, 4433, 4434, 4435, 4437, 4490, 4491, 4492
- \ctex_set_upfonts:nnnnnn 1713, 4179, 4267, 4346, 4427, 4486
- \ctex_set_upmap:nnn 1699, 1715, 1716, 1717, 1718, 4190, 4192, 4436, 4438
- \ctex_set_zhmap:n 430, 501, 1701
- \ctex_titleps_hook: 3140, 3175, 3176
- \ctex_titlesec_hook: 3048, 3112
- \ctex_update_ccglue: 1932, 2176, 2195, 2239
- \ctex_update_ccwd: 1918, 2170, 2247
- \ctex_update_default_family: 46, 581, 1289, 1674, 1723
- \ctex_update_em_unit: 1967, 2165, 2215
- \ctex_update_kanjisize: 2058, 2060
- \ctex_update_parindent: 2143, 2198, 2240
- \ctex_update_size: 2137, 2138
- \ctex_update_stretch: 2142, 2163
- \ctex_update_xkanjiskip: 2012, 2032
- \ctex_update_ziju: 2145, 2213
- \ctex_varioref_hook: 3344, 3355, 3363
- \ctex_zhmap_case:nnn 134, 262, 267, 272, 4134, 4229, 4297, 4394, 4458
- \ctex_zhmap_input:n 501, 504
- \ctex_zihao:n 3724, 3725
- \ctex_ziju:n 2207, 2208
- ctex internal commands:
 - \g_ctex_after_end_preamble_hook_tl 5031, 5035, 5043
 - _ctex_assign_heading_name:nnn 2445
 - \l_ctex_autoindent_tl 132, 189, 194, 199, 2124, 2128, 2131, 2200, 2203
 - _ctex_calc_bounds:nn 4869, 4870, 4874
 - _ctex_ccglue: 1952, 1965
 - \l_ctex_ccglue_skip 1936, 1940, 1943, 1946, 1948, 1956, 1960, 2174, 2185, 2194, 2230, 2238
 - \c_ctex_class_tl 84, 86, 410, 415, 420, 425, 3628, 3634, 3636, 3637, 3639
 - _ctex_clear_fntef_color:n ... 1857, 1864, 1868, 1883
 - \c_ctex_cmap_encoding_seq 440, 447
 - \l_ctex_ctexcap_options_clist 5013, 5014, 5023
 - _ctex_def_heading_keys:n 2397, 2475
 - _ctex_default_pt:w 119, 124
 - \l_ctex_encoding_tl 128, 222, 223, 489, 492, 583, 591, 1826, 2096, 2255, 3669, 3696, 3710
 - \g_ctex_end_preamble_hook_tl 5029, 5033, 5042
 - \c_ctex_engine_file_str 43, 2100
 - \c_ctex_engine_str 43
 - \l_ctex_family_default_init_tl 1726, 1746
 - _ctex_family_default_wrap:n 1729, 1747, 1750
 - _ctex_fix_varioref_label:n 3349
 - \g_ctex_font_size_flag 151, 162, 163, 164, 378, 382, 404, 3686, 3687, 3713, 3714, 3815
 - \c_ctex_font_size_prop 3727, 3738, 3782, 3809
 - \c_ctex_font_size_seq 3736, 3739, 3750
 - \l_ctex_font_size_tl 3727, 3728, 3774, 3775, 3809, 3810
 - \g_ctex_fontset_tl 226, 1758, 1776, 1801, 1808, 1809, 3933, 3936, 3939, 3940, 3946, 3948, 3968, 3970, 3975, 3992
 - _ctex_get_font_sizes:Nn 3774, 3777
 - _ctex_get_macro_meaning:w 5181
 - \l_ctex_heading_bool 305, 3624
 - \l_ctex_heading_skip 2488, 2495, 2497, 2499, 2500, 2511, 2512, 2515, 2517, 2518, 2521
 - \c_ctex_headings_cs_seq 3007, 3021, 3030
 - \c_ctex_headings_seq ... 2357, 2472, 3125, 3188, 3346
 - _ctex_hookto_cmd:Nnnnw 5097, 5102, 5260
 - _ctex_hookto_cmd_parameter:Nnnnw 5267, 5287
 - _ctex_hookto_cmd_parameterless:Nnnnw . 5266, 5272
 - _ctex_initial_heading:n 2367, 2474
 - \g_ctex_kanjisize_func 2038, 2048, 2059
 - _ctex_left_hook_aux:nn 5285
 - \l_ctex_line_spread_fp 110, 182, 3688, 3689, 3715, 3716, 3901, 3903
 - \l_ctex_line_stretch_tl 2156, 2161, 2162, 2166
 - \l_ctex_ltj_add_alternate_bool 1142, 1154, 1172, 1179
 - _ctex_ltj_alternate_cs:n 1016, 1019, 1020, 1021, 1055, 1056, 1058, 1145, 1148, 1149, 1151, 1152, 1376, 1413, 1475, 1500, 1540, 1557
 - \l_ctex_ltj_alternate_family_tl 1438, 1441
 - \l_ctex_ltj_alternate_options_clist 1391, 1392, 1403, 1407
 - \l_ctex_ltj_alternate_seq 986, 1360, 1361, 1369, 1375, 1383, 1631
 - \l_ctex_ltj_base_CJKfamily_tl 987, 1001, 1053, 1475, 1500
 - \l_ctex_ltj_base_family_tl 64, 1017, 1048, 1054, 1419, 1460, 1519, 1538, 1543, 1544, 1556
 - _ctex_ltj_change_encoding: 951, 1035, 1433
 - \l_ctex_ltj_char_range_clist 985, 1344, 1353, 1372, 1390, 1402, 1414, 1437, 1459, 1610, 1612
 - _ctex_ltj_char_range_parse_feature:w . 1615, 1619
 - \g_ctex_ltj_char_range_prop 1467, 1574
 - \l_ctex_ltj_char_range_tl 1467, 1470, 1576, 1578, 1580, 1582, 1589
 - _ctex_ltj_check_family:n 1011, 1374
 - \l_ctex_ltj_current_font_tl 50, 745, 749, 750, 762, 804, 809
 - \l_ctex_ltj_current_shape_tl 827, 832, 837, 838
 - _ctex_ltj_declare_char_range:nn 1570, 1572
 - \g_ctex_ltj_default_features_clist 1005, 1255, 1256
 - _ctex_ltj_family_csname:n ... 1015, 1032, 1050, 1077
 - \g_ctex_ltj_family_font_name_prop 992, 1008, 1013, 1090, 1166, 1292, 1304
 - \g_ctex_ltj_family_font_options_prop 993, 1008, 1169
 - \g_ctex_ltj_family_int . 1132, 1134, 1452, 1453, 1455

`\g_ctex_ltj_family_name_prop` 1008, 1017, 1040, 1074, 1419
`_ctex_ltj_family_unknown_warning:n` 1068, 1088, 1548
`\l_ctex_ltj_font_options_clist` 989, 994, 995, 996, 1002, 1039, 1170, 1171, 1174, 1410, 1411, 1415
`_ctex_ltj_fontspec:nnn` 1133, 1140, 1160
`\g_ctex_ltj_fontspec_family_tl` 1038, 1041, 1042, 1043, 1049, 1054, 1079
`\g_ctex_ltj_fontspec_prop` 1128, 1156, 1161
`_ctex_ltj_gset_family_cs:nn` 997, 1030
`_ctex_ltj_if_jfont_math:w` 870, 873
`\l_ctex_ltj_jfm_tl` 942, 963, 965, 1006
`\c_ctex_ltj_math_fam_int` 971, 972
`\c_ctex_ltj_math_family_tl` 1326, 1328, 1330, 1333, 1337
`\c_ctex_ltj_math_tl` 966, 967, 969, 971, 1251, 1317, 1327, 1332, 1336
`_ctex_ltj_msg_def_family_map:n` 1103, 1105
`_ctex_ltj_msg_family_map:n` 1029, 1102, 1116
`_ctex_ltj_noembed_wrap:n` 979, 981
`_ctex_ltj_pass_args:nnnn` 1187, 1203, 1213, 1219, 1231, 1238, 1244, 1250
`_ctex_ltj_patch_external_font:w` 790, 823
`_ctex_ltj_pop_fontname:` 761, 770, 849
`_ctex_ltj_post_arg:w` 1190, 1196
`_ctex_ltj_push_fontname:n` 757, 765, 846
`\g_ctex_ltj_reset_alternate_prop` 1022, 1542, 1554, 1562
`_ctex_ltj_save_alternate_shape:Nnn` 1474, 1499, 1515
`_ctex_ltj_save_fontname:w` 767, 771
`_ctex_ltj_select_font_aux:` 746, 752
`_ctex_ltj_set_alternate_family:n` 1044, 1051
`_ctex_ltj_tmp:w` 1349, 1366, 1444, 1446, 1447
`\l_ctex_ltj_tmp_seq` 1357, 1358, 1361, 1368
`\l_ctex_ltj_tmp_tl` 988, 989, 1013, 1024, 1027, 1167, 1175, 1209, 1212, 1214, 1294, 1299, 1317, 1318, 1320, 1321, 1355, 1356, 1357, 1394, 1396, 1397, 1404, 1614, 1622, 1631
`\g_ctex_ltj_unknown_family_seq` ... 1092, 1094, 1099
`_ctex_ltj_update_family_aux:` 1277, 1279
`_ctex_ltj_update_family_uid:N` 995, 1410, 1449
`_ctex_ltj_use_global_options:N` 996, 1003, 1392, 1411
`\c_ctex_macosx_file_str` 1807, 1811
`\l_ctex_msyh_suffix_tl` 4123, 4124, 4128, 4145, 4147, 4181
`\l_ctex_parameter_str` 5178, 5197, 5218, 5265
`\l_ctex_parameter_tl` 5218, 5223, 5234
`_ctex_parse_name:nnNNnN` 5130, 5139
`_ctex_parse_name:w` 5152, 5158, 5167
`_ctex_patch_cmd:Nnnnnw` 5072, 5077, 5082, 5236
`_ctex_patch_false:w` 5108, 5114, 5251, 5253, 5270, 5296
`_ctex_patch_rebuild:Nn` 5209, 5215, 5248, 5289, 5293
`_ctex_patch_replace:nnnTF` 5243, 5255
`_ctex_patch_rescan:Nn` 5217, 5218, 5219, 5227
`_ctex_patch_true:w` 5107, 5113, 5249, 5282, 5294
`_ctex_plus_key_aux:nn` 2427, 2429, 2430, 2431, 2432, 2433, 2434, 2437
`\l_ctex_prefix_str` 5178, 5196, 5217, 5274, 5275
`\l_ctex_prefix_tl` 5217, 5222, 5233
`\l_ctex_punct_bounds_clist` ... 4855, 4861, 4867, 4873
`\c_ctex_punct_family_clist` 609, 619
`\l_ctex_punct_font` 4853, 4854
`\c_ctex_punct_seq` 4832, 4833, 4856
`\l_ctex_punct_tl` 281, 578, 579, 665, 950, 2087, 2088, 2089, 2090
`_ctex_rebuild_cmd:w` ... 5209, 5210, 5214, 5289, 5290
`\l_ctex_replacement_str` 5178, 5198, 5219, 5257, 5258, 5292
`\l_ctex_replacement_tl` 5219, 5224, 5235
`_ctex_restore_selectfont:` 1983, 1987, 1994
`_ctex_right_hook_aux:nn` 5286
`_ctex_save_bounds:n` 4857, 4865
`_ctex_save_cmap:Nn` 460, 464
`_ctex_save_font_size:nn` 3738
`\l_ctex_scheme_tl` 316, 3365, 3641
`\g_ctex_section_depth_flag` 94, 310, 312, 3563, 3579, 3595
`\c_ctex_section_headings_seq` . 2352, 2359, 2962, 3066
`_ctex_set_font_size:nnNn` 3810, 3813
`\g_ctex_spa_iow` 4843, 4846, 4848, 4858
`\c_ctex_std_class_tl` ... 3612, 3627, 3628, 3694, 3717
`\g_ctex_std_options_clist` 137, 383, 395, 405, 407, 411, 416, 421, 426
`_ctex_titlesec_format:n` 3062, 3070, 3073
`_ctex_titlesec_format:Nn` 3052, 3076, 3078
`_ctex_titlesec_format_explicit:Nn` ... 3053, 3088
`_ctex_titlesec_hook:n` 3062, 3066, 3068
`_ctex_titlesec_spacing:Nn` 3071, 3098
`_ctex_titlesec_spacing:nnnnnn` 3099, 3100
`_ctex_tmp:N` 1979, 1990
`_ctex_tmp:w` 5118, 5171, 5182, 5203
`\l_ctex_tmp_box` 58, 1921, 1922, 3192, 3196
`\l_ctex_tmp_dim` 58, 2166, 2167, 2172, 2175, 2181, 2220, 2222, 2224, 2233, 2235
`\l_ctex_tmp_int` 58, 559, 561, 562, 564, 568, 570, 2180, 2182, 2183, 2189, 2190, 2222, 2223, 2224, 2227, 2228, 2233, 2235, 2961, 2964, 2968
`\l_ctex_tmp_tl` 58, 367, 374, 389, 390, 466, 467, 558, 560, 1981, 1992, 2399, 2457, 2470, 2480, 3693, 3694, 3782, 3783
`\c_ctex_upper_case_file_str` 1795, 1803
`\c_ctex_version_tl` 3, 6, 7, 8, 9, 20, 23
`_ctex_write_family:nn` 4845, 4850
`\l_ctex_xkanjiskip_skip` 2012
`\l_ctex_xkanjiskip_tl` 2019, 2024, 2025
`\c_ctex_zero_tl` 2140, 2147, 2149
`\g_ctex_zhmCJK_bool` 131, 261, 266, 271, 488, 582
`\l_ctex_ziju_dim` 2216, 2217, 2221, 2232, 2234, 2238, 2242
`\l_ctex_ziju_tl` 2140, 2148, 2149, 2210, 2216
`\ctexbibname` 2325, 2345
`\CTEXcounter` 2260

`\CTEXdigits` 25, 2270
`\CTEXfilltwosides` 30, 1891, 1901
`\CTEXifname` 16, 2503, 2759, 2941, 2949, 2957,
 3183, 3225, 3228, 3232, 3244, 3248, 3256, 3259, 3264, 3267
`\CTEXindent` 30, 2243
`\CTEXnoindent` 30, 2243
`\CTEXnumber` 25, 2270
`\CTEXnumberline` 2955, 2975, 3440
`\CTEXoptions` 30, 2104
`\ctexset` 5, 2101, 2108, 2114
`\CTEXsetfont` 30, 2135
`\CTEXsetup` 30, 2104
`\CTEXsout` 30
`\ctexspadef` 655, 4860
`\CTEXthechapter` 16, 2706, 2950, 3118, 3244, 3264
`\CTEXtheparagraph` 16
`\CTEXthepart` 16, 2942, 3123
`\CTEXthesection` 16, 3225, 3232, 3248, 3256, 3267
`\CTEXthesubparagraph` 16
`\CTEXthesubsection` 16, 3228, 3259
`\CTEXthesubsubsection` 16
`\CTEXunderdblline` 30
`\CTEXunderdot` 30
`\CTEXunderline` 30
`\CTEXunderwave` 30
`\CTEXxout` 30
`\CurrentOption` 395
`\cwd` 2034

D

`declarecharrange` 26, 1564
`\DeclareFontEncoding` 952
`\DeclareFontFamily`
 474, 961, 4012, 4013, 4014, 4015, 4016, 4017
`\DeclareFontShape`
 962, 964, 1690, 1691, 1694, 1695, 4028, 4029,
 4030, 4031, 4032, 4033, 4034, 4035, 4036, 4039, 4040,
 4041, 4042, 4043, 4044, 4045, 4046, 4047, 4051, 4052,
 4053, 4054, 4055, 4056, 4060, 4061, 4062, 4063, 4064,
 4065, 4066, 4067, 4068, 4071, 4072, 4073, 4074, 4075,
 4076, 4077, 4078, 4079, 4083, 4084, 4085, 4086, 4090,
 4091, 4092, 4093, 4094, 4095, 4096, 4097, 4098, 4101,
 4102, 4103, 4104, 4105, 4106, 4107, 4108, 4109, 4113, 4114
`\DeclareFontSubstitution` 955
`\DeclareKanjifamily`
 1688, 1689, 4019, 4020, 4021, 4022, 4023, 4024
`\DeclareMathSizes` 3775
`\DeclareSymbolFont` 967, 1327
`\def` 2567, 2604, 2636, 2654,
 2666, 2697, 2737, 2746, 2770, 2789, 2815, 2819, 2854,
 2869, 2883, 2884, 2888, 3835, 3846, 3874, 3885, 4524, 4560
`\defaultCJKfontfeatures` 1229
`\defbeamertemplate` 3273, 3292, 3311
`\defbeamertemplatealias` 3330, 3331, 3332
dim commands:
`\dim_abs:n` 2234

`\dim_add:Nn` 2217
`\dim_compare:nNnTF` ... 112, 2167, 2202, 2218, 2225, 2487
`\dim_eval:n` 2187, 2233
`\dim_max:nn` 3195
`\dim_min:nn` 2234
`\dim_new:N` 61, 1931, 2035, 2036, 2037, 2242
`\dim_set:Nn` 1922, 1968, 1969,
 1970, 2166, 2172, 2203, 2216, 2220, 2247, 2494, 3193, 3905
`\dim_sub:Nn` 2224
`\dim_to_decimal:n` 36, 3746, 3747, 3784
`\dim_to_decimal_in_unit:nn` 4880
`\dim_use:N` 120, 3905
`\dim_zero:N` 2253
`\c_max_dim` 2167, 2487
`\c_zero_dim` 2175, 2187, 2202, 2218, 2225, 2412, 3596
`\dimexpr` 2586, 2641, 2752, 2776, 3280, 3299, 3318
`\directlua` 703, 704
`\document` 5036, 5039

E

`\edef` 4543, 4544
`\else` . 1677, 1678, 1682, 1683, 2539, 2548, 2557, 2573, 2578,
 2610, 2615, 2685, 2693, 2709, 2715, 2721, 2733, 2742,
 2800, 2805, 2824, 2829, 2848, 2853, 2862, 2882, 2898,
 2911, 3404, 3431, 4555, 4556, 4602, 4641, 4672, 4731, 4794
else commands:
`\else:` 491, 728, 821, 867, 1953, 3029, 3229, 3909
`\em` 1675
`\eminnershape` 1677, 1678, 1683, 1685
`\end` 3288, 3307, 3326, 4925
`\endCJKfilltwosides` 1897
`\endcsname` 2807,
 2816, 2817, 2818, 2825, 2843, 2844, 2846, 2849, 2858,
 2859, 2860, 2863, 4550, 4552, 4553, 4554, 4557, 4559, 4563
`\endCTEXfilltwosides` 1897, 1902
`\endgroup` 2845, 2881, 2906, 3290, 3309, 3328, 4543, 4551, 4566
`\endinput` 4548
`\endlinechar` 4522, 4523
etex commands:
`\etex_dimexpr:D` 120, 2181
`\etex_glueexpr:D` 2173
`\etex_iffontchar:D` 726
`\everypar` 2804, 2901, 2913
`\EverySelectfont` 1998, 2001
exp commands:
`\exp_after:wN`
 119, 774, 790, 792, 867, 870, 3099, 3728, 3775, 3810, 5187
`\exp_args:cc` 3340
`\exp_args:Nc` 3071, 3076, 5141
`\exp_args:Nco` 3181
`\exp_args:Nnc` 100
`\exp_args:NNf` 1622
`\exp_args:NNNo` 3904
`\exp_args:Nno` 3352
`\exp_args:No` 560, 1804, 3095, 4857, 5203
`\exp_args:Nx` 294, 588, 662, 2207, 3724, 3903

- `\exp_last_unbraced:Nn` 5152
`\exp_last_unbraced:NNn` 1991
`\exp_not:N` 124, 125,
470, 590, 591, 592, 736, 737, 739, 809, 845, 955, 1040,
1041, 1042, 1043, 1401, 1522, 1734, 1735, 1736, 1750,
1886, 1887, 1888, 1915, 2373, 2374, 2375, 2380, 2382,
2383, 2384, 2385, 2388, 2389, 2392, 2404, 2405, 2406,
2407, 2408, 2417, 2418, 2419, 2421, 2424, 2439, 2441,
2443, 2967, 2969, 2970, 2971, 2972, 3083, 3084, 3093,
3094, 3131, 3133, 3134, 3744, 3946, 5132, 5133, 5134, 5222
`\exp_not:n` 69,
389, 738, 740, 1036, 1037, 1038, 1039, 1174, 1175, 1402,
1403, 1404, 1428, 1522, 1729, 1751, 1974, 2425, 2480,
3102, 3341, 5015, 5039, 5150, 5222, 5223, 5224, 5279, 5280
`\exp_stop_f:` 726, 727
`\expandafter`
4525, 4526, 4550, 4551, 4552, 4553, 4554, 4559, 4563, 4564
`\ExplSyntaxOff` 66, 72, 479, 522, 526, 530, 534, 685,
1652, 1656, 1660, 1664, 1676, 1992, 3204, 3272, 4886, 5085
`\ExplSyntaxOn` 71, 714, 3333, 4825
- ## F
- `\familydefault` 1275, 1286, 1732
`fancyhdr` 29, 342
`\fangsong` 4508
`\fi` .. 2530, 2535, 2541, 2550, 2559, 2562, 2577, 2582, 2590,
2599, 2614, 2619, 2629, 2649, 2669, 2677, 2678, 2681,
2685, 2695, 2713, 2719, 2725, 2736, 2745, 2749, 2757,
2758, 2767, 2773, 2785, 2790, 2796, 2802, 2809, 2811,
2833, 2834, 2850, 2864, 2867, 2886, 2895, 2914, 2915,
3224, 3227, 3231, 3241, 3242, 3247, 3255, 3258, 3263,
3266, 3284, 3303, 3322, 3404, 3431, 4555, 4556, 4565, 4818
fi commands:
`\fi:` 408, 538, 575,
594, 730, 821, 867, 1953, 1996, 3032, 3233, 3249, 3819, 3909
`\figurename` 2306
`figurename` 11
file commands:
`\g_file_current_name_tl` 677, 1804
`\file_if_exist:nTF`
..... 48, 1791, 1793, 1795, 1807, 3939, 4118, 4125, 4127
`\file_if_exist_input:nTF` 86
`\file_input:n` 78, 87, 89, 511, 547
`fn` 30, 342
`\font` 4012, 4013, 4014, 4015, 4016, 4017
`\fontfamily` 1268
`fontset` 7, 226, 3959
`\fontsize` 3728
fontspec commands:
`\g_fontspec_encoding_tl` 960
`\l_fontspec_family_tl` 1461
`\fontspec_set_family:Nnn` 1038, 1438
`\fontspec_visible_space:` 724
`\fontspec_visible_space_fallback:` 729
fontspec internal commands:
`__fontspec_fontname_wrap:n` 55, 979
- `\footnotesep` 3904
`\footnotesize` 3841, 3880, 3904
fp commands:
`\fp_compare:nNnTF` 3688, 3715, 3901
`\fp_eval:n` 2210, 4876
`\fp_set:Nn` 3689, 3716
`\fp_use:N` 2147, 3903
`\c_nan_fp` 183
`\c_zero_fp` 2147, 3688, 3715, 3901
- ## G
- `GBK` 6, 222
`\gdef` 2993, 2994, 2995, 2996, 2999, 3000, 3001, 3002
`\global` 2689, 2900, 2903
`\globaljfont` 792
`\glueexpr` 2840, 2855,
2876, 2883, 2884, 2910, 3279, 3289, 3298, 3308, 3317, 3327
group commands:
`\group_begin:` 474,
719, 754, 776, 802, 844, 871, 919, 984, 1034, 1347, 1432,
1517, 1537, 1553, 1728, 1978, 2039, 2379, 2387, 2455,
3018, 3140, 3199, 3904, 4852, 5033, 5109, 5117, 5126, 5181
`\group_end:` 478, 487, 723, 759,
780, 815, 848, 875, 923, 999, 1045, 1367, 1439, 1523,
1549, 1560, 1740, 1991, 2057, 2390, 2394, 2479, 3033,
3163, 3203, 3904, 4863, 5033, 5129, 5176, 5204, 5238, 5262
`\group_insert_after:N` 812, 813
`\gtfamily` 1677, 1682
- ## H
- hbox** commands:
`\hbox_set:Nn` 1921, 3192
`heading` 9, 305
`\heiti` 4507
`\hskip` ... 2840, 2855, 2876, 2883, 2884, 2909, 2910, 3255, 3258
`\hspace` 2572, 2609, 2942, 2950, 3123
`\Huge` 2630, 2660, 2761, 2779, 3408, 3438, 3858, 3897
`\huge` 2591,
2627, 2643, 2756, 3390, 3407, 3411, 3436, 3444, 3857, 3896
`hyperref` 30, 342
`\hypersetup` 359, 360, 1817
`\hyphenchar` 4012, 4013, 4014, 4015, 4016, 4017
- ## I
- if** commands:
`\if_bool:N` 488, 582
`\if_case:w` 404, 3815
`\if_cs_exist:N` 555, 1977
`\if_false:` 107
`\if_meaning:w` 1952
`\if_true:` 106
`\ifctexpdf` 105
`\ifdim` 2794, 2836, 2872, 2890
`\ifincsname` 3909
`\IfNoValueTF`
... 1189, 1210, 1486, 1591, 1624, 1626, 2109, 2115, 2451

`\ifnum` 2568,
 2587, 2605, 2626, 2698, 2754, 2820, 2848, 2862, 3224,
 3227, 3231, 3238, 3239, 3247, 3255, 3258, 3263, 3266, 4555
`\ifodd` 2535, 2537, 2562, 2569, 2587, 2599, 2606,
 2626, 2649, 2669, 2691, 2702, 2749, 2767, 2773, 2785,
 2798, 2809, 2825, 2837, 2873, 2891, 2895, 3282, 3301, 3320
`\ifx` 4553, 4554, 4559
`\ifzhmappdf` 4571, 4638, 4669, 4700, 4767
`\ignorespaces` 2916
`\immediate` 4562
`indent` 29, 202
`\indexname` 2309
`indexname` 12
`\input` 4824, 4889
`\insertcontinuationtext` 2318, 2327, 2334
`\insertpart` 3286, 3287
`\insertromanpartnumber` 3277, 3377
`\insertsection` 3305, 3306
`\insertsectionnumber` 3296, 3475
`\insertsubsection` 3324, 3325
`\insertsubsectionnumber` 3315, 3504

int commands:

`\c_five` 3747
`\c_four` 312, 568, 570
`\int_compare:nNnTF`
 561, 2183, 2227, 3563, 3579, 3595, 3686, 3713
`\int_const:Nn` 971
`\int_div_truncate:nn` 79, 568
`\int_eval:n` 1487, 1489, 1490, 1592, 1594, 1595
`\int_from_hex:n` 560
`\int_gincr:N` 1132, 1452
`\int_incr:N` 2964
`\int_mod:nn` 570
`\int_new:N` 59, 152, 1455
`\int_set:Nn` 153, 545, 551, 559, 2180, 2222
`\int_sub:Nn` 564, 2228
`\int_to_Hex:n` 562, 567, 569
`\int_use:N` ... 471, 551, 859, 1134, 1453, 2048, 2968, 4857
`\int_zero:N` 2961
`\c_one` 163, 1480, 1585, 2235, 2446, 4869
`\c_six` 3747
`\c_three` 310, 2227, 3579, 4870
`\c_two` 95, 164, 378, 382, 2228, 3196, 3563, 3595
`\c_zero` 162, 2183, 3687, 3714
`\interlinepenalty`
 2584, 2623, 2639, 2657, 2760, 2778, 2841, 2842, 2877, 2878

iow commands:

`\iow_close:N` 4846
`\iow_indent:n` 5020
`\iow_new:N` 4848
`\iow_now:Nn` 4858
`\iow_open:Nn` 4843
`\itemsep` 3838, 3849, 3877, 3888
`\itshape` 1677, 1678, 1682, 1683

J

`\jfam` 911, 972

K

`\kaishu` 4509
`\kanjifamily` 1653, 1657, 1661, 1665, 1670
`\kanjifamilydefault` 1673
 kernel internal commands:
`\l_kernel_expl_bool` 70
 keys commands:
`\l_keys_choice_tl` 295
`\keys_define:nn` 156, 389, 392, 976,
 1340, 1449, 1526, 1564, 1603, 2062, 2083, 2118, 2152,
 2275, 2301, 2329, 2339, 2342, 2345, 2480, 2977, 3912, 3959
`\keys_if_exist:nnTF` 1601
`\l_keys_key_tl` 141, 143, 145
`\keys_set:nn` 28, 172,
 178, 234, 243, 252, 278, 288, 295, 302, 332, 338, 2102,
 2109, 2115, 3368, 3423, 3451, 3483, 3518, 3535, 3549,
 3565, 3573, 3581, 3589, 3596, 3597, 3599, 3665, 3667, 4001
`\keys_set_known:nn` 3644
`\keys_set_known:nnN` 988

L

`\labelformat` 3352
`\languagealias` 3673, 3678, 3699, 3704
`\LARGE` 3856, 3895
`\Large` 2588, 3388, 3394, 3462, 3466, 3855, 3894
`\large` 3487, 3854, 3893
`\lastbox` 2904
`\leavevmode` 2530, 2790
 left commands:
`\c_left_brace_str` 5173
`\leftmargin` 3835, 3846, 3874, 3885
`\leftmargini` 3835, 3846, 3874, 3885
`\let` 1986, 2823, 2832, 3828, 3867, 4546, 4552
`\linespread` 3903
`linespread` 10, 182
`linestretch` 10, 2152
`\linewidth` 2181, 2189, 2221
`\lishu` 4511
`\listfigurename` 2304
`listfigurename` 11
`\listtablename` 2305
`listtablename` 11
`\LoadClass` 412, 417, 422, 427
`\long` 4560
`\lstlistingname` 737, 738
`\lstlistlistingname` 739, 740
`\ltjalchar` 727
`\ltjdefcharrange` 686, 687, 688, 690, 694, 695, 698, 701
`LTJFONTUID` 1449
`\ltjgetparameter` 1926, 1956, 2016, 2030
`\ltjsetkanjiskip` 1943
`\ltjsetmathletter` 921
`\ltjsetparameter` 702, 708, 1642
`\ltjsetxkanjiskip` 2020

lua commands:

\lua_escape_x:n
 .. 820, 832, 859, 866, 1477, 1492, 1493, 1510, 1511, 1520
 \lua_now_x:n 720, 794,
 820, 824, 829, 834, 856, 865, 958, 1498, 1522, 1760, 2041

luatex commands:

\luatex_luafunction:D 2059

M

\MAKESPA 4849, 4891
 \markboth 2502, 2593, 2621
 \mathgroup 911
 \mcfamily 1682
 \mddefault 956, 962, 968, 1328, 1333

mode commands:

\mode_if_math:TF 905

msg commands:

\msg_critical:nnn 53, 677, 3928
 \msg_error:nn 3979
 \msg_error:nnn 19, 22, 38, 41, 668, 681, 684, 2291, 3729, 3811
 \msg_error:nnnn 3945, 3974
 \msg_fatal:nn 4830
 \msg_new:nn 4828
 \msg_new:nnn 140, 142, 144, 146, 148, 671,
 1028, 1100, 1182, 1873, 2322, 3618, 3929, 3988, 5016, 5093
 \msg_new:nnnn 12, 30, 45, 99, 2293, 3731, 3953, 3982
 \msg_warning:nn 1177, 1877, 2080, 2091, 2344
 \msg_warning:nnn
 170, 176, 204, 214, 229, 238, 247, 276, 286, 300,
 330, 336, 344, 349, 355, 1095, 3637, 3916, 3966, 5022, 5092
 \msg_warning:nnnn 1024, 1886, 1893, 2107, 2113, 2245, 2251

N

\newCJKfontfamily 1201
 \NewDocumentCommand
 1059, 1196, 1201, 1207, 1217, 1223, 1229,
 1236, 1242, 1248, 1254, 1387, 1479, 1584, 1619, 1668,
 2102, 2105, 2111, 2135, 2206, 2243, 2249, 2270, 2272,
 2445, 3723, 3908, 4506, 4507, 4508, 4509, 4511, 4512, 4514
 \newfontfeature 973, 974, 975
 \newluafunction 2038
 \newpage 2671, 2676
 \ngostype 44, 1778
 \nobreak 2588,
 2596, 2646, 2756, 2761, 2764, 2779, 2782, 2892, 3389, 3437
 \nobreakspace 2588, 2627
 nocap 29
 \noexpand 4544, 4546, 4547, 4548
 nofonts 29
 \noindent 2922
 noindent 29
 nopunct 29
 \normalem 1856
 \normalfont 534, 536, 537, 1233,
 1240, 1246, 1311, 1664, 1666, 1667, 2585, 2586, 2624,
 2625, 2640, 2641, 2658, 2659, 2751, 2752, 2775, 2776, 2972
 \normalsize 3521, 3538, 3552, 3822, 3861, 3899, 3907

nospace 29
 nozhmap 29
 \null 2560, 2674
 \numberline 2708, 2849, 2863, 2950, 2958, 3134, 3190

O

\onecolumn 2555
 or commands:
 \or: 406, 3817

P

\pagestyle 3664
 \par 2493, 2531, 2588, 2627, 2630, 2643,
 2660, 2756, 2761, 2779, 2791, 2892, 3278, 3279, 3286,
 3289, 3297, 3298, 3305, 3308, 3316, 3317, 3324, 3327,
 3370, 3389, 3400, 3418, 3426, 3437, 3478, 3480, 3512, 3514
 paragraph/afterskip 31
 paragraph/beforeskip 31
 paragraph/hang 21
 paragraph/numbering 15
 paragraph/runin 19
 \parindent
 2202, 2203, 2225, 2246, 2247, 2252, 2253, 2585, 2586,
 2640, 2641, 2751, 2752, 2775, 2776, 3280, 3299, 3318, 3597
 \parsep 3837, 3838, 3848, 3849, 3876, 3877, 3887, 3888
 \part 2528
 part/beforeskip 31
 part/fixbeforeskip 31
 part/numbering 15
 part/pagestyle 21, 2455
 \partmark 2501, 2594, 2622
 \partname 2588, 2627, 3277, 3372
 \PassOptionsToClass 411, 416, 421, 426
 \PassOptionsToPackage 489, 1821, 2255, 5024
 \pdfmapline 4572,
 4573, 4574, 4575, 4576, 4577, 4578, 4580, 4581, 4582,
 4583, 4584, 4585, 4586, 4588, 4589, 4590, 4591, 4592,
 4593, 4595, 4596, 4597, 4598, 4599, 4600, 4701, 4702,
 4703, 4704, 4705, 4706, 4707, 4709, 4710, 4711, 4712,
 4713, 4714, 4715, 4717, 4718, 4719, 4720, 4721, 4722,
 4724, 4725, 4726, 4727, 4728, 4729, 4768, 4769, 4770,
 4771, 4772, 4773, 4775, 4776, 4777, 4778, 4779, 4780,
 4782, 4783, 4784, 4785, 4786, 4788, 4789, 4790, 4791, 4792
 \pdfoutput 4555
 \pdfstringdefDisableCommands 3911
 pdftex commands:
 \pdftex_pdffontattr:D 470
 \pdftex_pdflastobj:D 471
 \pdftex_pdfobj:D 467
 prg commands:
 \prg_do_nothing:
 915, 1426, 1518, 1902, 1976, 2067, 2934, 5214
 \prg_new_conditional:Npnn 818, 1949
 \prg_new_protected_conditional:Npnn 1072, 5184
 \prg_return_false: ... 821, 1082, 1953, 1957, 1961, 5194
 \prg_return_true: 821, 1075, 1080, 1953, 1957, 1961, 5199
 \primitive 4925

\ProcessKeysOptions 401
 \proofname 2296, 2311
 proofname 12
 prop commands:
 \prop_get:NnN 1169
 \prop_get:NnNTF
 1074, 1128, 1166, 1419, 1467, 3727, 3782, 3809
 \prop_gpop:NnNTF 1013, 1017, 1554
 \prop_gput:Nnn ... 992, 993, 1040, 1156, 1542, 1578, 3744
 \prop_gremove:Nn 1022
 \prop_if_empty:NTF 1090, 1292
 \prop_map_break:n 1306
 \prop_map_inline:Nn 1304
 \prop_new:N 1008, 1009, 1010, 1161, 1562, 1583, 3738
 \protect 2515, 2521, 2708,
 2728, 2729, 2849, 2863, 2950, 2958, 3133, 3145, 3147,
 3151, 3153, 3157, 3159, 3167, 3172, 3184, 3185, 5145, 5146
 \ProvideDocumentCommand 2501
 \ProvidesFile 4560, 4568, 4635, 4666, 4697, 4764
 \providetranslation
 4930, 4931, 4932, 4933, 4934, 4935, 4936,
 4937, 4938, 4939, 4940, 4941, 4942, 4943, 4944, 4945,
 4946, 4947, 4948, 4949, 4950, 4951, 4952, 4953, 4954,
 4955, 4956, 4957, 4958, 4959, 4960, 4961, 4962, 4963,
 4964, 4965, 4966, 4967, 4968, 4969, 4970, 4971, 4972,
 4973, 4974, 4975, 4976, 4977, 4978, 4979, 4980, 4981,
 4982, 4983, 4984, 4985, 4986, 4987, 4988, 4989, 4990,
 4991, 4992, 4993, 4994, 4995, 4996, 4997, 4998, 4999,
 5000, 5001, 5002, 5003, 5004, 5005, 5006, 5007, 5008, 5009
 ptex commands:
 \ptex_kanjiskip:D 1929, 1946, 1960
 \ptex_xkanjiskip:D 2017, 2021, 2031
 punct 9, 29, 281, 2083
 \punctstyle 579, 2088

Q

\quad 2816,
 3224, 3225, 3227, 3228, 3231, 3232, 3244, 3248, 3256,
 3259, 3264, 3267, 3395, 3445, 3455, 3488, 3522, 3539, 3553
 quark commands:
 \q_mark 5188, 5191
 \q_stop 120, 125, 870,
 873, 1384, 1388, 1615, 1620, 5153, 5158, 5167, 5188, 5191
 \quotation 3921

R

\raggedright 2585, 2640, 2751, 2775, 3387, 3435
 \refname 2312, 2317, 2324, 2333, 2341, 2342
 refname 12
 \refstepcounter ... 2571, 2608, 2704, 2827, 3335, 3357, 3358
 \relax 2586, 2605, 2626, 2641, 2752, 2776, 2792,
 2828, 2835, 2840, 2855, 2871, 2876, 2883, 2884, 2889,
 2908, 2910, 3255, 3258, 3279, 3280, 3289, 3298, 3299,
 3308, 3317, 3318, 3327, 4518, 4522, 4525, 4553, 4554, 4559
 \renewcommand 2529, 2545, 2684
 \RenewDocumentCommand 1268, 3358

\RequirePackage 27,
 56, 57, 351, 362, 490, 493, 494, 495, 661, 679, 682,
 716, 736, 1721, 1830, 1839, 1855, 1861, 2256, 3718, 5069
 \RequirePackageWithOptions 5025
 resetalternatefont 27, 1526
 reverse commands:
 \reverse_if:N 582
 \rmdefault 1283, 1734
 \rmfamily 522, 525, 1652, 1655

S

scan commands:
 \scan_stop:
 ... 64, 120, 792, 911, 1519, 1520, 2173, 2181, 4853, 5035
 scheme 9, 316
 \scriptsize 3852, 3891
 \secdef 2542, 2564, 2696
 section/afterskip 31
 section/beforeskip 31
 section/hang 21
 section/numbering 15
 section/runin 19
 \sectionname 3296, 3472
 \selectfont 1066, 1545, 1558, 1671, 1990, 1995, 2009, 2137, 3728
 seq commands:
 \seq_clear:N 986
 \seq_concat:NNN 1360
 \seq_gput_left:Nn 2360, 2361
 \seq_gput_right:Nn 1094, 3750
 \seq_gset_eq:NN 2359
 \seq_gset_from_clist:Nn 2354, 2364, 3008, 4833
 \seq_if_empty:NTF 1375
 \seq_if_in:NnTF 447, 1092
 \seq_map_function:NN 3066, 3188
 \seq_map_inline:Nn
 ... 1383, 2472, 2962, 3011, 3021, 3030, 3125, 3346, 4856
 \seq_new:N
 ... 440, 1099, 1368, 1369, 2353, 2357, 3007, 3739, 4832
 \seq_put_right:Nn 1631
 \seq_set_filter:NNn 1358
 \seq_set_from_clist:Nn 441
 \seq_set_split:Nnn 1357
 \seq_use:Nnnn 3736
 \setbox 2904
 \setCJKfamilyfont 1113,
 1201, 4146, 4152, 4153, 4154, 4155, 4156, 4157, 4200,
 4201, 4209, 4210, 4212, 4213, 4214, 4215, 4217, 4239,
 4241, 4243, 4245, 4284, 4285, 4286, 4287, 4311, 4316,
 4321, 4323, 4365, 4367, 4369, 4370, 4384, 4385, 4386,
 4387, 4400, 4401, 4402, 4403, 4404, 4405, 4446, 4447,
 4448, 4449, 4450, 4451, 4464, 4465, 4466, 4499, 4500, 4501
 \setCJKmainfont 1109, 1229, 4137, 4196,
 4204, 4231, 4277, 4299, 4356, 4381, 4396, 4441, 4460, 4495
 \setCJKmathfont 1229
 \setCJKmonofont 1111, 1229, 4151, 4199,
 4208, 4238, 4283, 4310, 4364, 4383, 4399, 4445, 4463, 4498

`\setCJKromanfont` 1235, 1261
`\setCJKsansfont` 1110, 1229, 4140, 4144, 4198,
 4206, 4237, 4282, 4305, 4362, 4382, 4398, 4443, 4462, 4497
`\SetSymbolFont` 969, 1332, 1336
`\sfdefault` 1284, 1735
`\sffamily` 526, 529, 1656, 1659
skip commands:
`\skip_horizontal:N` 1936, 1940
`\skip_if_eq:nnTF` 1956, 1960, 2015, 2512, 2518
`\skip_new:N` 1948, 2028, 2497
`\skip_set:Nn` 1926,
 1929, 2019, 2029, 2174, 2185, 2230, 2238, 2499, 2511, 2517
`\skip_set_eq:NN` 1946, 2021
`\skip_sub:Nn` 2488, 2495
`\skip_use:N` 2515, 2521
`\skip_zero:N` 2194
`\c_zero_skip` 2410, 2411, 2463, 2464, 2512, 2518
`\small` 3830, 3869
`\songti` 4506
`\space` 2705, 2756, 3372, 3433, 3472, 3503, 3605
`space` 10, 29, 291, 2062
`\special` 1703, 1704, 1707,
 1708, 1837, 1851, 4603, 4604, 4605, 4606, 4607, 4608,
 4609, 4611, 4612, 4613, 4614, 4615, 4616, 4617, 4619,
 4620, 4621, 4622, 4623, 4624, 4626, 4627, 4628, 4629,
 4630, 4631, 4642, 4643, 4644, 4645, 4646, 4648, 4649,
 4650, 4651, 4652, 4654, 4655, 4656, 4657, 4659, 4660,
 4661, 4662, 4673, 4674, 4675, 4676, 4677, 4679, 4680,
 4681, 4682, 4683, 4685, 4686, 4687, 4688, 4690, 4691,
 4692, 4693, 4732, 4733, 4734, 4735, 4736, 4737, 4738,
 4740, 4741, 4742, 4743, 4744, 4745, 4746, 4748, 4749,
 4750, 4751, 4752, 4753, 4755, 4756, 4757, 4758, 4759,
 4760, 4795, 4796, 4797, 4798, 4799, 4801, 4802, 4803,
 4804, 4805, 4807, 4808, 4809, 4810, 4812, 4813, 4814, 4815
`\SplitArgument` 1480, 1585, 2446
str commands:
`\c_backslash_str` 5172
`\c_colon_str` 897, 898, 899
`\str_case:nnTF` 1732, 1778, 5143, 5158
`\str_case_x:nn` 1281
`\str_case_x:nnTF` 1107, 1118
`\str_const:Nn` 43, 50, 1803, 1811
`\str_if_empty:NTF` 5229, 5265, 5274
`\str_if_eq:nnTF` 492, 583,
 944, 1275, 1826, 1834, 3669, 3696, 3710, 3968, 3985, 3992
`\str_if_eq_x:nnTF` 578, 1296, 3965, 5149
`\str_lower_case:n` 466
`\str_new:N` 5069, 5178, 5179, 5180
`\str_set:Nn` 5196, 5197, 5198
`\str_upper_case:n` 1804
`\strutbox` 3905
`sub3section` 9, 308
`sub4section` 9, 308
`subparagraph/afterskip` 31
`subparagraph/beforeskip` 31
`subparagraph/hang` 21

`subparagraph/numbering` 15
`subparagraph/runin` 19
`subsection/afterskip` 31
`subsection/beforeskip` 31
`subsection/hang` 21
`subsection/numbering` 15
`subsection/runin` 19
`\subsectionname` 3315, 3503
`subsubsection/afterskip` 31
`subsubsection/beforeskip` 31
`subsubsection/hang` 21
`subsubsection/numbering` 15
`subsubsection/runin` 19
sys commands:
`\c_sys_engine_str` 44
`\sys_if_engine_pdftex:TF`
 130, 4132, 4224, 4292, 4375, 4392, 4456
`\sys_if_engine_uptex:TF`
 4177, 4265, 4344, 4378, 4425, 4484
`\sys_if_engine_xetex:TF` 4444, 4826
`\sys_if_output_pdf:TF` 105, 476, 1829, 4226, 4294

T

`\tablename` 2307
`tablename` 12
`\TeX` 3909
T_EX and L^AT_EX 2_ε commands:
`\@@italiccorr` 1644
`\@@par` 2841, 2877, 3456, 3489, 3523, 3567, 3583
`\@afterheading` 2601, 2651, 2735, 2744, 2897
`\@afterindentfalse` 2540, 2690, 2694, 2795, 2801
`\@afterindenttrue` 2538, 2692, 2793, 2799
`\@Alph` 3601, 3606
`\@chapapp` 2705, 2756, 3240, 3263
`\@chapter` 2696, 2697
`\@chinese` 2258, 2267
`\@clubpenalty` 2912
`\@currenthref` 3042
`\@currentlabel` 3337, 3341
`\@currentx` 4, 5015, 5064
`\@currname` 4, 5015, 5065
`\@dblarg` 2814
`\@defaultunits` 36
`\@empty` 100, 2823, 2832
`\@endpart` 2633, 2663, 2665
`\@EverySelectfont@Init` 1977, 1985
`\@firstoftwo` 3184
`\@hangfrom` 2921
`\@ifclassloaded` 3614
`\@ifpackagelater` 37, 40, 667, 680, 683, 1862
`\@ifpackageloaded` 18, 21, 676, 1814, 3105, 3208, 3354, 5046
`\@ifpackagewith` 3050, 3059, 3110
`\@ifstar` 2812
`\@listI` 3828, 3867
`\@listi` 3828, 3835, 3846, 3867, 3874, 3885

\@M	2584,	\AtEndPreamble	131
2623, 2639, 2657, 2760, 2778, 2841, 2842, 2877, 2878, 2905		\baselineskip	10, 11, 75, 79
\@makechapterhead	2732, 2734, 2746	\bfseries	47
\@makeschapterhead	2741, 2743, 2770	\bibname	12
\@minus	3457, 3490, 3524, 3540, 3554, 3824,	\c@chapter	3606
3826, 3832, 3834, 3836, 3837, 3843, 3845, 3847, 3848,		\c@secnumdepth	
3863, 3865, 3871, 3873, 3875, 3876, 3882, 3884, 3886, 3887		2568, 2587, 2605, 2626, 2698, 2754, 2820, 2848, 2862,	
\@namedef . 3145, 3147, 3151, 3153, 3157, 3159, 3167, 3172		3224, 3227, 3231, 3238, 3239, 3247, 3255, 3258, 3263, 3266	
\@nameuse	3145, 3147, 3157, 3159, 3167	\c@section	3601
\@ne	3103, 3227, 3258	\caption	30
\@nil	845	\catcode	35, 44, 44, 122, 132
\@nobreakfalse	2899	\ccwd . 10, 10, 11, 24, 25, 30, 30, 36, 73, 74, 74, 78, 78, 79, 79	
\@noskipsecfalse	2903	\chapter	9, 15, 16, 21, 23, 23, 85, 105
\@noskipsectrue	2900	\chaptermark	13
\@onlypreamble	36, 437, 503,	\chaptername	15
1257, 1258, 1259, 1260, 1261, 1262, 1698, 1712, 1720, 3952		\char	49
\@part	2542, 2564, 2566	\CheckCommand	74
\@pkgextension	6, 7, 8, 9, 20, 23, 678, 5064	\Chinese	30
\@plus	3401,	\chinese	25, 30, 81
3402, 3457, 3458, 3490, 3491, 3524, 3525, 3540, 3554,		\CJK@ignorespaces	46, 596
3568, 3584, 3824, 3825, 3826, 3832, 3833, 3834, 3836,		\CJK@encoding	54, 56, 63, 751, 755, 839,
3837, 3843, 3844, 3845, 3847, 3848, 3863, 3864, 3865,		951, 1327, 1330, 1332, 1336, 1477, 1492, 1493, 1510, 1511	
3871, 3872, 3873, 3875, 3876, 3882, 3883, 3884, 3886, 3887		\CJK@envStart	45, 45, 46, 590
\@popfilename	131, 5062	\CJK@family	49,
\@schapter	2696, 2737	743, 756, 803, 839, 1063, 1544, 1554, 1556, 2009, 2010	
\@secntformat	2815, 2828	\CJK@ignorespaces	46, 46, 601, 605
\@secondoftwo	3185	\CJK@input	45, 539
\@secpenalty	2806, 3460, 3493, 3527, 3541, 3555	\CJK@loadBinding	44, 45, 586
\@sect	2814, 2819	\CJK@loadEncoding	44
\@setfontsize	3814	\CJK@makeActive	45, 585
\@spart	2542, 2564, 2635	\CJK@plane	45, 75, 455, 456, 466, 1998
\@ssect	2813, 2869	\CJK@surr	45, 555
\@startsection	92, 94, 101, 2789, 2967	\CJK@upperReset	46
\@svsec	2823, 2828, 2832, 2840, 2856, 2857	\CJKfamily	56
\@svsechd	2854, 2883, 2884, 2906	\CJKfamilydefault	45, 46, 60, 60, 68, 69, 69, 69, 69
\@tempdima	3193, 3195	\CJKglue	79, 110
\@tempskipa	2792, 2794, 2795,	\CJKhook	46, 46
2806, 2835, 2836, 2871, 2872, 2889, 2890, 2893, 2908, 2909		\CJKrmdefault	43
\@tempswafalse	2558	\CJKsymbol	75
\@tempswatrue	2556	\CJKunderdotbasesep	30
\@topnewpage	2732, 2741	\CJKunderline	30
\@topnum	2689	\clearpage	46
\@xsect	2868, 2887, 2888	\contentsname	11, 11
\@zhdig	2266	\CS	50
\@zhnum	2258, 2265	\csname	110
\abstractname	12	\CTEX@addloflotskip	2508, 2730
\addCJKfontfeature	64	\CTEX@addtocline ... 2525, 2583, 2620, 2726, 2852, 2866	
\AfterEndPreamble	131	\CTEX@afterindent	2798, 2928, 2935
\algorithmname	12	\CTEX@aftertitle	2880, 2885, 2927, 2934
\appendixname	12, 23	\CTEX@appendix@number	2982, 2994, 3000
\AtBeginDocument	36, 45	\CTEX@appendix@numbering	2983, 2996, 3002
\AtBeginDvi	42	\CTEX@chapter@afterindent	2691
\AtBeginShipoutFirst	42	\CTEX@chapter@aftername	2759
\AtEndOfClass	37	\CTEX@chapter@afterskip	2766, 2784
\AtEndOfPackage	37	\CTEX@chapter@aftertitle	2763, 2781
\AtEndOfPackageFile*	131	\CTEX@chapter@beforeskip	2748, 2772

<code>\CTEX@chapter@break</code>	2687	<code>\CTEX@save@refstepcounter</code>	3357, 3360
<code>\CTEX@chapter@fixskip</code>	2749, 2767, 2773, 2785	<code>\CTEX@section@aftername</code>	3302
<code>\CTEX@chapter@format</code>	2753, 2777	<code>\CTEX@section@afterskip</code>	3308
<code>\CTEX@chapter@indent</code>	2752, 2776	<code>\CTEX@section@aftertitle</code>	3306
<code>\CTEX@chapter@lofskip</code>	2461	<code>\CTEX@section@beforeskip</code>	3298
<code>\CTEX@chapter@lotskip</code>	2462	<code>\CTEX@section@format</code>	3300
<code>\CTEX@chapter@numbering</code>	2702, 3002	<code>\CTEX@section@indent</code>	3299
<code>\CTEX@chapter@pagestyle</code>	2460, 2688	<code>\CTEX@section@numbering</code>	2996, 3301
<code>\CTEX@chapter@titleformat</code>	2762, 2780	<code>\CTEX@section@titleformat</code>	3306
<code>\CTEX@chapter@tocline</code>	2939	<code>\CTEX@sectionname</code>	3302
<code>\CTEX@chaptername</code>	2759	<code>\CTEX@selectfont@hook</code>	1971, 1998, 2001
<code>\CTEX@document@left@hook</code>	5032, 5038	<code>\CTEX@selectfont@save</code>	1982, 1986
<code>\CTEX@document@right@hook</code>	5034, 5040	<code>\CTEX@setcurrentlabel@n</code>	3335, 3361
<code>\CTEX@Family@CMap</code>	452, 480, 483	<code>\CTEX@setheadingskip</code>	2498, 2534, 2561, 2598, 2648, 2668, 2748, 2766, 2772, 2784, 2808, 2894
<code>\CTEX@fixheadingskip</code>	2486, 2491, 2535, 2599, 2649, 2669, 2767, 2785, 2809, 2895	<code>\CTEX@subsection@aftername</code>	3321
<code>\CTEX@fixskip</code>	2809, 2895, 2929, 2936	<code>\CTEX@subsection@afterskip</code>	3327
<code>\CTEX@fixtopskip</code>	2483, 2562, 2749, 2773	<code>\CTEX@subsection@aftertitle</code>	3325
<code>\CTEX@fontfamily</code>	1037, 1268	<code>\CTEX@subsection@beforeskip</code>	3317
<code>\CTEX@hang</code>	2920, 2930, 2937	<code>\CTEX@subsection@format</code>	3319
<code>\CTEX@hangfrom</code>	2840, 2876, 2918	<code>\CTEX@subsection@indent</code>	3318
<code>\CTEX@headingskip</code>	2500, 2536, 2563, 2600, 2650, 2670, 2750, 2768, 2774, 2786, 2810, 2896	<code>\CTEX@subsection@numbering</code>	3320
<code>\CTEX@hyperheadinghook</code>	3016, 3046	<code>\CTEX@subsection@titleformat</code>	3325
<code>\CTEX@ifnamefalse</code>	2503, 2574, 2579, 2611, 2616, 2637, 2655, 2710, 2716, 2722, 2738, 2821, 2830, 2870	<code>\CTEX@subsectionname</code>	3321
<code>\CTEX@ifnametrue</code>	2503, 2570, 2607, 2703, 2826	<code>\CTEX@thechapter</code>	3000
<code>\CTEX@makeanchor</code>	2575, 2580, 2612, 2617, 2711, 2831, 3005, 3014, 3026, 3031, 3037	<code>\CTEX@thesection</code>	2994
<code>\CTEX@makeanchor@chapter</code>	2717, 2723	<code>\CTEX@titleformat@n</code>	92, 2879, 2885, 2926, 2933
<code>\CTEX@makeanchor@schapter</code>	2739	<code>\CTEX@titleslabel@clear</code>	3154, 3164
<code>\CTEX@makeanchor@sect</code>	2822	<code>\CTEX@titleslabel@set</code>	3148, 3160, 3164
<code>\CTEX@makeanchor@spart</code>	2638, 2656	<code>\CTEX@toc@width@n</code>	3190, 3205, 3215
<code>\CTEX@makeanchor@ssect</code>	2813	<code>\CTEX@todayold</code>	2274, 2279
<code>\CTEX@part@afterindent</code>	2537	<code>\CTEX@update@sectionformat@n</code>	2797, 2924
<code>\CTEX@part@aftername</code>	2589, 2628, 3283	<code>\CTEX@verbatim@font@hook</code>	1637, 1641
<code>\CTEX@part@afterskip</code>	2598, 2648, 2668, 3289	<code>\ctex@zhmap@endinput</code>	4820
<code>\CTEX@part@aftertitle</code>	2595, 2632, 2645, 2662, 3287	<code>\CTEXdigits</code>	25
<code>\CTEX@part@beforeskip</code>	2534, 2561, 3279	<code>\CTEXifname</code>	16
<code>\CTEX@part@break</code>	2532, 2551	<code>\CTEXnumber</code>	25, 25
<code>\CTEX@part@fixskip</code>	2535, 2562, 2599, 2649, 2669	<code>\CTEXnumberline</code>	22
<code>\CTEX@part@format</code>	2586, 2625, 2642, 2659, 3281	<code>\CTEXoptions</code>	30, 30, 77
<code>\CTEX@part@indent</code>	2586, 2641, 3280	<code>\ctexset</code>	5, 5, 7, 11, 14, 26, 27, 30, 30, 111
<code>\CTEX@part@numbering</code>	2569, 2587, 2606, 2626, 3282	<code>\CTEXsetup</code>	30
<code>\CTEX@part@pagestyle</code>	2459, 2553	<code>\CTEXthechapter</code>	12, 13, 16, 96
<code>\CTEX@part@titleformat</code>	2592, 2631, 2644, 2661, 3287	<code>\CTEXthesection</code>	12, 13, 16, 98
<code>\CTEX@part@tocline</code>	2939	<code>\CTEXunderdotbasesep</code>	30
<code>\CTEX@partname</code>	2589, 2628, 3283	<code>\CTEXunderline</code>	30
<code>\CTEX@postappendix</code>	2987, 2995, 3001	<code>\curr@fontshape</code>	757, 787, 796, 800, 846
<code>\CTEX@postchapter</code>	3001	<code>\DeclareAlternateKanjiFont</code>	63
<code>\CTEX@postsection</code>	2995	<code>\DeclareFontEncoding</code>	53
<code>\CTEX@preappendix</code>	2986, 2993, 2999	<code>\DeclareFontFamily</code>	42, 51
<code>\CTEX@prechapter</code>	2999	<code>\DeclareFontShape</code>	51
<code>\CTEX@presection</code>	2993	<code>\DeclareFontShape@</code>	63, 1436
<code>\CTEX@runin</code>	2837, 2873, 2891, 2931, 2938	<code>\DeclareRobustCommand</code>	133
<code>\CTEX@save@appendix</code>	2988, 2991	<code>\define@newfont</code>	50, 779
		<code>\do@subst@correction</code>	50, 51, 778
		<code>\em</code>	68
		<code>\escapechar</code>	53

\EverySelectfont	74	\linewidth	79
\external@font	789, 790, 792	\lishu	8
\extract@font	50, 777	\listfigurename	11
\f@baselineskip	2173	\listtablename	11
\f@encoding	755, 795, 908	\ltj@does@alt@set	51
\f@family	50, 756, 795, 803, 1270, 1281	\ltj@getjfontnumber	52, 855
\f@series	751, 839	\ltj@set@stackfont	897, 898, 899
\f@shape	751, 839	\ltj@allalchar	721
\f@size	751, 757, 834, 842, 845, 846, 1922, 1968, 3196	\ltj@curjfont	50, 75
\fangsong	8	\ltj@pickup@altfont@auxy	840
\figurename	11	\ltj@pickup@altfont@copy	853
\font	51	\ltj@setpar@global	896
\font@name	50, 51, 760, 767, 768, 771, 774, 792, 793, 810, 888, 889, 890, 891, 892, 899	\ltj@tempcntc	52, 859
\fontcharwd	74	\ltjalchar	49
\fontfamily	60	\m@ne	2568, 2587, 2698, 2754, 3231, 3238, 3239, 3263, 4012, 4013, 4014, 4015, 4016, 4017
\footnotesep	110, 110	\markboth	13
\footnotesize	110	\markright	13
\get@external@font	786	\math@bgroup	907
\getanddefine@fonts	877	\math@egroup	912
\globaljfont	50	\math@fonts	894
\H@old@chapter	3020	\mathbb	110
\hbox	85	\maxdimen	11, 78
\heiti	8	\meaning	133, 134, 134, 134
\hrule	85	\newcommand	133
\Hy@chapapp	2711, 2717, 2723, 2739	\newrobustcmd	133
\Hy@driver	1834	\newtitlemark	97
\Hy@MakeCurrentHrefAuto	3039	\nfss@catcodes	54
\Hy@org@chapter	3020	\ngostype	70
\Hy@raisedlink	3040	\normalsize	8, 106
\Hy@unicodetrue	71	\nouppercase	45
\hyper@anchorend	3043	\p@	1922, 1968, 2627, 2728, 2729, 2747, 2756, 2765, 2771, 2783, 3196, 3400, 3427, 3428, 3429, 3430, 3437, 3824, 3825, 3826, 3832, 3833, 3834, 3836, 3837, 3843, 3844, 3845, 3847, 3848, 3863, 3864, 3865, 3871, 3872, 3873, 3875, 3876, 3882, 3883, 3884, 3886, 3887
\hyper@anchorstart	3042	\pagenumbering	25
\HyPsd@ConvertToUnicode	71	\pagestyle	9
\HyPsd@LoadUnicode	72	\paragraph	9, 19, 35
\HyPsd@pdfencoding	72	\parindent	10, 30, 78, 80, 110
\HyPsd@ToBigChars	72	\parskip	85
\if@mainmatter	2700, 2755, 3238, 3632, 3633	\part	14, 20, 21, 85
\if@nobreak	2803	\partmark	85
\if@noskipsec	2530, 2790, 2902	\partname	15
\if@openright	2546, 2673, 2685, 3404, 3431	\partpage	14, 99
\if@tempswa	2679	\PassOptionsToClass	42
\if@twocolumn	2554, 2731, 2740	\PassOptionsToPackage	71
\if@twoside	2672, 3222, 3245	\pdfmapline	116, 119
\iffalse	53	\pdfstringdef	71
\ifHy@implicit	3019	\pickup@font	50
\ifin@	53, 867	\pickup@jfont	783
\iftrue	53	\ProcessKeysOptions	28
\indexname	12	\proofname	12, 81
\insertcontinuationtext	12	\protected	96
\itdefault	51	\protected@edef	2271, 2273, 2828, 3337
\itshape	47	\ProvidesFile	123
\kaishu	8		
\kanjiskip	75		
\labelformat	100		
\leftmark	13		

<code>\ps@fancy</code>	3251	<code>\wrong@fontshape</code>	50
<code>\ps@headings</code>	3221	<code>\x@protect</code>	5146
<code>\refname</code>	12, 12	<code>\xdef</code>	96
<code>\refstepcounter</code>	100	<code>\XeTeXglyphbounds</code>	127
<code>\relax</code>	45, 78	<code>\xkanjiskip</code>	75
<code>\reset@font</code>	536, 1666	<code>\yahei</code>	8
<code>\rightmark</code>	13	<code>\youyuan</code>	8
<code>\rmfamily</code>	48	<code>\z@</code>	708, 711, 2585, 2640, 2689, 2751, 2775, 2794, 2836, 2872, 2890, 2904, 3103, 3224, 3247, 3255, 3266, 3825, 3833, 3844, 3864, 3872, 3883
<code>\roman</code>	25	<code>\zhdig</code>	81
<code>\scantokens</code>	133	<code>\zhdigits</code>	25
<code>\scriptfont@name</code>	891, 898	<code>\zhnum</code>	81
<code>\section</code> . 9, 14, 15, 19, 20, 20, 21, 21, 22, 23, 23, 31, 82, 98		<code>\zhnumber</code>	25, 25
<code>\section*</code>	15	<code>\zihao</code>	24
<code>\sectionmark</code>	13	<code>\ziju</code>	24
<code>\sectionname</code>	15	<code>\zw</code>	74
<code>\sectionpage</code>	14, 99	tex commands:	
<code>\selectfont</code>	52, 69, 74, 75, 75, 78	<code>\tex_def:D</code>	5222
<code>\setCJKfamilyfont</code>	26, 26	<code>\tex_divide:D</code>	79, 2182, 2223
<code>\setCJKmainfont</code>	26	<code>\tex_edef:D</code>	5276
<code>\sf@size</code>	890	<code>\tex_endlinechar:D</code>	545, 551
<code>\sffamily</code>	48	<code>\tex_font:D</code>	470, 726, 4853
<code>\size@update</code>	78, 79, 2137, 2151	<code>\tex_iftrue:D</code>	3633
<code>\sldefault</code>	51	<code>\tex_ignorespaces:D</code>	1060, 1221, 1226, 2073, 2207, 3724, 5035
<code>\songti</code>	7	<code>\tex_immediate:D</code>	467
<code>\special</code>	7, 71	<code>\tex_pagegoal:D</code>	2487
<code>\split@name</code>	845	<code>\tex_parskip:D</code>	2495
<code>\SplitArgument</code>	84	<code>\tex_prevdepth:D</code>	2494
<code>\ssf@size</code>	892	<code>\tex_topskip:D</code>	2488
<code>\strutbox</code>	110	<code>\the</code>	4522, 4525, 4526, 4545, 4549
<code>\subparagraph</code>	9, 19, 35	<code>\thechapter</code>	2705, 2708, 2756, 3240, 3263, 3434
<code>\subsectionname</code>	15	<code>\theparagraph</code>	3537
<code>\subsectionpage</code>	14, 99	<code>\thepart</code>	2572, 2588, 2609, 2627, 3374
<code>\subst@correction</code>	806, 813	<code>\thesection</code>	3224, 3231, 3247, 3255, 3266, 3454
<code>\tablename</code>	12	<code>\thesubparagraph</code>	3551
<code>\textfont@name</code>	889, 897	<code>\thesubsection</code>	3227, 3258, 3486
<code>\tf@size</code>	888	<code>\thesubsubsection</code>	3520
<code>\thechapter</code>	13, 16, 23	<code>\thispagestyle</code>	2552, 2553, 2675, 2686, 2688
<code>\thesection</code>	13, 15, 16, 23, 98	<code>\tiny</code>	3853, 3892
<code>\titleformat</code>	94	tl commands:	
<code>\titlespacing</code>	94	<code>\c_space_tl</code>	471
<code>\today</code>	11, 11	<code>\tl_clear:N</code>	194, 390, 1576, 1580, 1858, 1869, 2010, 2128, 2452, 2470, 3779, 5230
<code>\trans@languagepath</code>	3681, 3707	<code>\tl_clear_new:N</code>	58, 367
<code>\ttfamily</code>	48	<code>\tl_const:Nn</code>	3, 410, 415, 420, 425, 518, 656, 678, 951, 966, 1264, 1265, 1266, 1267, 1326, 2147, 2371, 2377, 3615, 3634, 3636
<code>\ttl@a</code>	3123, 3131	<code>\tl_gput_right:Nn</code>	448, 5029, 5031, 5053
<code>\ttl@chapterout</code>	3118	<code>\tl_gset:Nn</code>	562, 565, 888, 890, 892, 1300, 1730, 1758, 1776, 1801, 1808, 1809, 3970
<code>\ttl@extract</code>	94	<code>\tl_gset_eq:NN</code>	1042, 1753
<code>\ttl@setifthe</code>	3177	<code>\tl_gset_rescan:Nnn</code>	1307
<code>\ttl@setsubmark</code>	96, 3150, 3156	<code>\tl_head:N</code>	3783
<code>\ttl@settopmark</code>	96, 3144, 3175	<code>\tl_if_blank:nTF</code>	1393, 1489, 1490, 1594, 1595, 1609, 1627
<code>\ttl@tocpart</code>	3120, 3122		
<code>\ttlh@hang</code>	3084, 3094		
<code>\ttlh@runin</code>	3083, 3093		
<code>\uppercase</code>	45		
<code>\use@mathgroup</code>	903, 934		
<code>\usepackage</code>	29, 30		
<code>\verbatim@font</code>	1635, 1639		

- \tl_if_blank_p:n 1359
 - \tl_if_empty:NTF 746, 2200, 2383, 3351, 3933
 - \tl_if_empty:nTF 1692, 1705, 5193
 - \tl_if_eq:NNTF 1726, 2140, 3694
 - \tl_if_exist:NTF 84, 517, 519, 520, 521, 1263,
1265, 1266, 1267, 2296, 2327, 2338, 2341, 3627, 3717, 5052
 - \tl_if_in:NnTF 5257
 - \tl_map_inline:Nn 1505
 - \tl_map_inline:nn 3780
 - \tl_new:N 128, 132, 519, 520, 521,
748, 749, 837, 948, 1001, 1027, 1048, 1049, 1070, 1441,
1582, 1746, 2024, 2148, 2161, 2298, 2369, 2370, 2986,
2987, 4123, 5042, 5043, 5052, 5113, 5114, 5233, 5234, 5235
 - \tl_put_left:Nn 1985, 5062
 - \tl_put_right:Nn
..... 374, 894, 1589, 1622, 2399, 2441, 2443, 3783, 3784
 - \tl_replace_all:Nnn 1356, 1397
 - \tl_rescan:nn 134, 5275, 5284
 - \tl_set:Nn 114, 115, 129,
189, 222, 223, 466, 498, 499, 500, 558, 737, 739, 750,
789, 838, 945, 946, 987, 1053, 1065, 1209, 1270, 1355,
1394, 1396, 1437, 1610, 1614, 1648, 1649, 1650, 1651,
1673, 1748, 1981, 2009, 2025, 2087, 2096, 2124, 2162,
2210, 2299, 2450, 2453, 2457, 3080, 3099, 3118, 3123,
3131, 3693, 3911, 4124, 4128, 4171, 4172, 4173, 5107, 5108
 - \tl_set_eq:NN
755, 756, 803, 889, 891, 960, 1054, 1079, 1544, 1556, 2149
 - \tl_set_rescan:Nnn 5231
 - \tl_tail:N 3085
 - \tl_tail:n 3095
 - \tl_to_str:n 125, 126, 5174, 5203, 5244, 5245
 - \tl_trim_spaces:n 1573
 - \today 2274, 2279, 2282, 2287
 - today 11, 2274
 - token commands:
 - \token_get_replacement_spec:N 5131, 5153
 - \token_to_meaning:N 5188
 - \token_to_str:N
.. 53, 870, 879, 888, 890, 892, 1109, 1110, 1111, 1113,
1120, 1121, 1122, 1184, 2324, 2325, 3732, 4860, 5020, 5092
 - \toks 4522, 4523, 4525, 4526, 4545, 4549
 - \topsep 3836, 3847, 3875, 3886
 - \TrimSpaces 2105
 - \ttdefault 1285, 1736
 - \ttfamily 530, 533, 1660, 1663
 - two commands:
 - \c_two_hundred_fifty_six 561, 564
 - \twocolumn 2680
 - \typeout 2705, 2706, 3118
- ## U
- um commands:
 - \um_input_math_symbol_table: 927, 931
 - \um_sym:nnn 930
 - um internal commands:
 - _um_input_math_symbol_table: 939
 - _um_switchto_literal: 936
 - _um_sym:nnn 938
 - \undefined 4547
 - \unless 2837, 2873, 2891
 - \unskip 2907
 - \updefault ... 956, 962, 964, 968, 970, 1328, 1330, 1333, 1337
 - \upshape 1682, 1685
 - use commands:
 - \use:N 757, 795, 796, 846, 888, 890, 892,
971, 1019, 1557, 2511, 2517, 2526, 2958, 5258, 5278, 5292
 - \use:n 64, 122, 133, 389, 734, 845, 953, 1139,
1192, 1198, 1297, 1301, 1399, 1573, 1747, 1900, 2477,
2922, 2933, 3201, 3625, 3742, 3942, 5127, 5169, 5220, 5241
 - \use:nn 5286
 - \use_i:nn 108, 867, 2505, 5073, 5078, 5098, 5103
 - \use_i:nnn 262
 - \use_ii:nn ... 109, 867, 2503, 2507, 5073, 5078, 5098, 5103
 - \use_ii:nnn 134, 267
 - \use_iii:nnn 272
 - \use_none:n 190, 195, 200, 1036, 1037, 1300, 1434, 1901, 2260
 - \use_none:nn 483
 - \usebeamercolor 3277, 3296, 3315, 3417, 3477, 3511
 - \usebeamerfont 3276, 3286,
3295, 3305, 3314, 3324, 3416, 3419, 3476, 3479, 3510, 3513
 - \uselanguage 3672, 3677, 3698, 3703
 - \usepackage 5020
 - UTF8 6, 222
- ## V
- \verse 3921
 - \vfil 2560, 2667
 - \vskip 2597, 2600,
2627, 2647, 2650, 2670, 2756, 2765, 2768, 2783, 2786,
2893, 2896, 3278, 3297, 3316, 3400, 3418, 3437, 3478, 3512
 - \vspace 2563, 2747, 2750, 2771, 2774
- ## W
- winfonts 29
 - \write 4562
- ## X
- \x 4524, 4527,
4528, 4529, 4530, 4531, 4532, 4533, 4534, 4535, 4536,
4537, 4538, 4539, 4540, 4541, 4542, 4543, 4550, 4560, 4564
 - \xdef 4563
 - \xeCJKsetup 662, 1865, 1940, 2068, 2071, 2074, 2089
 - xetex commands:
 - \xetex_charglyph:D 4857
 - \xetex_glyphbounds:D 4881
- ## Y
- \yahei 4514
 - \youyuan 4512
- ## Z
- \zhdig 2266
 - \zhdigits 25, 2273

zhmap	7, 29, <u>257</u>	\zhtoday	2282, 2287
\zhnum	2265	\zihao	24, <u>3723</u> , 3732
zhnum commands:			
\zhnum_counter:n	2257	zihao	8, <u>159</u>
\zhnumber	25, 2271	\ziju	24, <u>2206</u>
\zhnumsetup	2283, 2288	\zw	709, 1926, 1969, 2026