

CT_EX 宏集手册

CTEX.ORG

2016/12/27 v2.4.7*

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
		8.1 字号与间距	24
第 2 节 简明教程	3	8.2 中文数字转换	25
2.1 CT _E X 宏集的组成	3	8.3 杂项	25
2.2 CT _E X 宏集的安装和更新	3	第 9 节 LuaL ^A T _E X 下的中文支持方式	25
2.3 使用 CT _E X 文档类	4	9.1 LuaL ^A T _E X 下替代字体的设置	26
2.4 使用 <code>ctex</code> 宏包	4	第 10 节 CT _E X 宏集的配置文	27
第 3 节 宏包选项与 <code>\ctexset</code> 命令	5	10.1 修改宏包默认选项	27
第 4 节 编译方式、编码与中文字库	5	10.2 宏包载入后的配置	28
4.1 编译方式	5	10.3 配置标题中文翻译	28
4.2 中文编码	6	10.4 自定义字体集	28
4.3 中文字库	6	第 11 节 对旧版本的兼容性	28
第 5 节 排版格式设定	8	11.1 CT _E X 0.8a 及以前的版本	28
5.1 文档默认字号	8	11.2 CT _E X 0.9–CT _E X 1.0d	29
5.2 章节标题风格	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	30
5.3 排版方案选项	9	11.4 CT _E X 2.2 之前的版本	31
第 6 节 文档汉化	11	11.5 CT _E X 2.4.1 和 2.4.2	31
6.1 日期汉化	11	第 12 节 宏集依赖情况与手工安装方法	31
6.2 文档标题汉化	11	第 13 节 开发人员	32
6.3 页面格式设置与汉化	12	第 14 节 代码实现	34
第 7 节 章节标题格式设置	14	版本历史	136
		代码索引	137

*`ctex-kit` rev. 6471e87.

第 1 节 介绍

历史

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

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

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

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

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 [GitHub](https://github.com/CTeX-org/ctex-kit)²。

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

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

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

关于宏集名字の説明

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

CT_EX 宏集是由 [CT_EX 社区](https://github.com/CTeX-org/ctex-kit) 发起并维护的 L^AT_EX 宏包和文档类的集合。社区另有发布名为 [CT_EX 套装](https://github.com/CTeX-org/ctex-kit) 的 T_EX 发行版，与本文档所述的 CT_EX 宏集并非是一事物。

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

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 C_TE_X 宏集组成

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

表 1 C_TE_X 宏集的组成

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

2.2 C_TE_X 宏集的安装和更新

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

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

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

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

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

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

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

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

⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

2.3 使用 CT_EX 文档类

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

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

例 1

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

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

以下是使用 `ctexbeamer` 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

`\ctexset {<键值列表>}`

New: 2014-03-18

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

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

例 4

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

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

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

表 2 CT_EX 宏集的中文支持方式

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

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

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

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

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

4.2 中文编码

GBK ☆
UTF8 ☆

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

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

4.3 中文字库

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

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

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

¹ Windows Vista 及以后的 Windows 操作系统。

² Windows XP 及以前的 Windows 操作系统。

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

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

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

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

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

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

zhmap ☆ zhmap = `<true|false|zhmCJK>`

Updated: 2014-03-08

指定字体映射机制。本选项只在使用 `pdfLATEX/LATEX` 编译时有意义。

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

fontset fontset = `<none|adobe|fandol|founder|mac|ubuntu|windows|windowsnew|windowsold|...>`

New: 2014-03-08

指定 C_T_EX 宏集加载的字库。

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

C_T_EX 预定义了以下六种中文字库。

`adobe` 使用 Adobe 公司的四款中文字体, 不支持 `pdfLATEX`。
`fandol` 使用 Fandol 中文字体, 不支持 `pdfLATEX`。
`founder` 使用方正公司的中文字体。
`mac` 使用 Mac OS X 系统下的华文字体, 不支持 `(pdf)LATEX`。
`ubuntu` 使用 Ubuntu 系统下的文泉驿和文鼎字体。
`windows` 使用简体中文 Windows 系统下的中文字体, 自动判断 Windows 系统版本, 采用 `windowsnew` 或 `windowsold` 的设置。
`windowsnew` 使用简体中文 Windows Vista 或之后系统下的中易字体和微软雅黑字体。
`windowsold` 使用简体中文 Windows XP 或之前系统下的中易字体。

注意: 使用 `(pdf)LATEX` 编译的时候, 若设置 `zhmap = false` (比如需要使用 `LATEX + Dvips` 编译), 则需要按照传统方式¹¹在本地安装好 CJK 字体。

如果不想使用 C_T_EX 预定义的中文字库, 可以设置 `fontset` 为下述值之一。

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

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

例 5

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

C_T_EX 宏集预定义的中文字库还定义了一些字体命令。

`\songti` 宋体, CJK 等价命令 `\CJKfamily{zhsong}`。

⁹如果需要使用自定义的字体映射文件, 或者希望使用 Type1 字库, 请禁用本选项。

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 `chinese` 和 `plain` 两种方案。

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

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

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

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

Updated: 2014-04-11

设置标点处理格式。预定义好的格式有:

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

`banjiao` 半角式: 所有标点占半个汉字宽度;

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

`CCT` CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

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

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

plain 原样(不调整标点间距)。

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

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

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

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

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

linespread ☆ linespread = `<数值>`

New: 2014-04-23

接受一个浮点数值,设置行距倍数。本选项的初始值与 scheme 有关。

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

scheme = plain CTeX 宏集默认不调整行距倍数,文档中的行距由所选文档类和其他宏包或用户设置决定。

autoindent autoindent = `<true|false|数值|带单位的数值>`

New: 2014-03-13

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

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

true 等价于设置 `autoindent = 2`。

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

例 8

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

linestretch ☆ linestretch = `<数值或长度>`

New: 2014-03-26

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

¹⁵LaTeX 将单个换行视作一个空格。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化,使之以中文的方式显示今天的日期。如编译本文档的日期就是“2016 年 12 月 27 日”。

`today` ★ `today = <small|big|old>`

该选项用来控制 `\today` 命令的输出格式:

`small` 效果为“2016 年 12 月 27 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年十二月二十七日”。使用全汉字的日期格式。
`old` 效果为“December 27, 2016”。使用文档原来的(英文)日期格式。

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

例 9

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

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

6.2 文档标题汉化

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

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

`contentsname` ★ `contentsname = <名字>`

设置目录标题名 `\contentsname`。中文默认为“目录”。

`listfigurename` ★ `listfigurename = <名字>`

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

`listtablename` ★ `listtablename = <名字>`

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

`figurename` ★ `figurename = <名字>`

设置图片环境标题名 `\figurename`。中文默认为“图”。

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

6.3 页面格式设置与汉化

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

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

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

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

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

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

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

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

例 10

```
\documentclass{ctexart}
\pagestyle{headings}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\renewcommand\sectionmark[1]{%
  \markright{\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`, `lotskip`, `lofskip`, `indent`, `hang`, `pagestyle`, `break`, `tocline` 等。

注意,对 `article` 及其衍生的 `ctexart` 等文档类,没有 `chapter` 级别的标题。而对于 `beamer` 文档类,这些选项控制的是由 `\partpage`, `\sectionpage` 和 `\subsectionpage` 产生的标题格式,此时只有 `part`, `section` 和 `subsection` 这三层级别,并且 `runin`, `afterindent`, `fix skip`, `hang`, `break` 和 `tocline` 这六个格式无效。

多级选项之间用斜线分开,例如, `part/name` 选项设置 `\part` 标题的在数字前后的名称,而 `section/number` 选项设置 `\section` 标题的数字类型。

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

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

例 12

```

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

```


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

numbering = true|false

控制是否对章节标题编号。对各级标题的默认值均为 true。

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

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

如果没有特别说明，以下将用“...”代表各级章节标题名。

.../name	★
Updated: 2014-03-08	

name = {<前名字>,<后名字>}

name = {<前名字>}

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

例 13

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

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

表 5 name 选项的默认设置

标题名	scheme = chinese	scheme = plain	注
part	{第,部分}	{\partname\space}	原 \partname 为 Part
chapter	{第,章}	{\chaptername\space}	原 \chaptername 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 \sectionname 为 \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}}{}#2

appendix/numbering ★ numbering = true|false

New: 2015-06-21

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

用法与普通章节 numbering 选项类似。

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

Updated: 2014-03-08

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

用法与普通章节 name 选项类似。

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

表 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 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 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:

$\langle name \rangle = [\langle alternate\ font\ features \rangle] \{ \langle alternate\ font\ name \rangle \}$

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

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

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

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

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

10.1 修改宏包默认选项

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

例 23

```
% 系统自带 ctexopts.cfg 注释中的示例语句, 固定默认字体集为 windowsnew。
% 该设置可以用在安装了 Windows 字体的非 Windows 系统中。
\keys_set:nn { ctex / option } { fontset = windowsnew }
```

如上例所示, 宏包选项通常使用 \LaTeX 的 `\keys_set:nn` 命令完成键值设置, 第一个参数是固定的子模块 `ctex/option`, 第二个参数中是用户定义的新的默认宏包选项。

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

10.2 宏包载入后的配置

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

例 24

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

例 25

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

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

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

<u>cs4size</u> <u>c5size</u>	分别相当于 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 pdf_LA_TE_X 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项, 其中 zhm-CJK 是可选的。
- xeCJK 宏包, 它还依赖
 - fontspec 宏包, 它还依赖
 - * euenc 宏包。
 - * xunicode 宏包, 它还依赖
 - graphics 宏集。

- graphics-cfg 宏包。
- graphics-def 宏包。
- environ 宏包, 它还依赖 trimspaces 宏包。
- ➡ 以上是使用 Xe_{La}T_EX 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_{se}l 宏包, 属于 platex-tools 宏集。
- zhmetrics-uptex 宏包。
- ➡ 以上是使用 up_AT_EX 编译时的依赖项。

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

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

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

若要手工安装, 请遵循如下步骤:

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

第 13 节 开发人员

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

目前比较活跃的开发维护人员是刘海洋、李清和黄晨成。

参考文献

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

第 14 节 代码实现

1 <@@=ctex>

宏包载入检查。

```
2 <*class|ctex>
3 \tl_const:Nx \c__ctex_version_tl
4 { \cs_if_exist_use:cF { ver@ \currname . \@currentx } { 9999/99/99 } }
5 <*class>
6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl
7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl
8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl
9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl
10 </class>
11 <*ctex>
12 \msg_new:nnnn { ctex } { subpackage-loaded }
13 { Package~#1'~can~not~be~loaded~with~`ctex'. }
14 {
15   `#1'~is~actually~a~part~of~`ctex'.\\
16   It~is~not~necessary~to~load~it~separately.
17 }
18 \@ifpackageloaded { ctexsize }
19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }
20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }
21 \@ifpackageloaded { ctexheading }
22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }
23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }
24 </ctex>
25 </class|ctex>
26 <*class|style>
27 \RequirePackage { xparse , l3keys2e }
28 </class|style>
29 <*class|ctex>
```

检查 expl3 和 l3keys2e 的版本。

```
30 \msg_new:nnnn { ctex } { l3-too-old }
31 { Support~package~`#1'~too~old. }
32 {
33   Please~update~an~up~to~date~version~of~the~bundles\\
34   `l3kernel'~and~`l3packages'\\
35   using~your~TeX~package~manager~or~from~CTAN.
36 }
37 \@ifpackagelater { expl3 } { 2015/12/20 } { }
38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
39 <*class>
40 \@ifpackagelater { l3keys2e } { 2015/12/20 } { }
41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
42 </class>
```

\c__ctex_engine_str
\c__ctex_engine_file_str

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

```
43 \str_const:Nx \c__ctex_engine_str
44 { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }
45 \msg_new:nnnn { ctex } { engine-not-supported }
46 { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }
47 { You~can~switch~to~xelatex,~lualatex,~pdflatex,~uplax,~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 , everyrel }

```

14.1 内部函数与变量

```

\l__ctex_tmp_tl 临时变量。
\l__ctex_tmp_int
\l__ctex_tmp_box 58 \tl_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 LATEX3
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 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one
153 </class>ctex|ctexsize>
```

14.2 宏包选项

```
154 <*class>style>
155 \keys_define:nn { ctex / option }
156 {
157 </class>style>

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

```

177     \keys_set:nn { ctex / option } { zihao = -4 }
178   } ,
179   c5size .value_forbidden:n = true ,
180   cs4size .value_forbidden:n = true ,

```

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

```

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

```

autoindent 自动调整段落的首行缩进功能。

```

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

```

indent 仅为兼容性保留, 已过时。

```

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

```

```

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

```

225   fontset .tl_gset:N = \g__ctex_fontset_tl ,
226   nofonts .code:n =

```

```

227     {
228       \msg_warning:nnn { ctex } { deprecated-option }
229       {
230         Option~`fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
231         fontset~ option.
232       }
233       \keys_set:nn { ctex / option } { fontset = none }
234     } ,
235     adobefonts .code:n =
236     {
237       \msg_warning:nnn { ctex } { deprecated-option }
238       {
239         Option~`fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
240         fontset~ option.
241       }
242       \keys_set:nn { ctex / option } { fontset = adobe }
243     } ,
244     winfonts .code:n =
245     {
246       \msg_warning:nnn { ctex } { deprecated-option }
247       {
248         Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
249         fontset~ option.
250       }
251       \keys_set:nn { ctex / option } { fontset = windows }
252     } ,
253     nofonts .value_forbidden:n = true ,
254     winfonts .value_forbidden:n = true ,
255     adobefonts .value_forbidden:n = true ,
256
zhmap 256     zhmap .choice: ,
257     zhmap .default:n = { true } ,
258     zhmap / zhmCJK .code:n =
259     {
260       \bool_gset_true:N \g__ctex_zhmCJK_bool
261       \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
262     } ,
263     zhmap / true .code:n =
264     {
265       \bool_gset_false:N \g__ctex_zhmCJK_bool
266       \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
267     } ,
268     zhmap / false .code:n =
269     {
270       \bool_gset_false:N \g__ctex_zhmCJK_bool
271       \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
272     } ,
273     nozhmap .code:n =
274     {
275       \msg_warning:nnn { ctex } { deprecated-option }
276       { Option~`zhmap=false'~ is~ set. }
277       \keys_set:nn { ctex / option } { zhmap = false }
278     } ,
279     nozhmap .value_forbidden:n = true ,

```

punct 设置标点符号输出格式。

```

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

```



```

288     } ,
289     nopunct    .value_forbidden:n = true ,

space 290     space .choices:nn =
291         { true , auto , false }
292         {
293             \exp_args:Nx \ctex_at_end:n
294             { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
295         } ,
296     space .default:n = { true } ,
297     nospace .code:n =
298     {
299         \msg_warning:nnn { ctex } { deprecated-option }
300         { Option~ `space=false' ~ is~ set. }
301         \keys_set:nn { ctex / option } { space = false }
302     } ,
303     nospace .value_forbidden:n = true ,

heading 304     heading .bool_set:N = \l__ctex_heading_bool ,

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

sub3section 307 <!*beamer>
sub4section 308     sub3section .code:n =
309         { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
310     sub4section .code:n =
311         { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
312     sub3section .value_forbidden:n = true ,
313     sub4section .value_forbidden:n = true ,
314 <!/beamer>

scheme 315     scheme .tl_set:N = \l__ctex_scheme_tl ,
316 <*ctexheading>
317     scheme .default:n = { plain } ,
318     scheme .initial:n = { plain }
319 }
320 </ctexheading>
321 <!*ctexheading>
322     scheme .default:n = { chinese } ,
323     scheme .initial:n = { chinese } ,
324 </ctexheading>

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

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

```

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

```

341     fntef .code:n =

```

```

342     {
343       \msg_warning:nnn { ctex } { deprecated-option }
344       { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
345     } ,
346     fancyhdr .code:n =
347     {
348       \msg_warning:nnn { ctex } { deprecated-option }
349       { `fancyhdr'~ package~ is~ loaded. }
350       \RequirePackage { fancyhdr }
351     } ,
352     hyperref .code:n =
353     {
354       \msg_warning:nnn { ctex } { deprecated-option }
355       { `hyperref'~ package~ will~ be~ loaded. }
356       \ctex_at_end:n
357       {
358         \cs_if_exist:NF \hypersetup
359         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
360       }
361       \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
362     } ,
363   }

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

```

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

```

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

```

将未知选项传给标准文档类。

```

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

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

```

```

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

```

五号字使用标准文档类的 10pt 字体大小设置,小四号字则使用 12pt。

```

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

```

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

```

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

```

14.3 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。

```

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

```

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

```

439 \seq_new:N \c__ctex_cmap_encoding_seq
440 \seq_set_from_clist:Nn \c__ctex_cmap_encoding_seq
441 { C19 , C10 , C00 , C09 , C40 , C60 }

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

首先检查选项,决定是否载入 `zhmCJK` 宏包。

```

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

```

不载入 `zhmCJK` 宏包时直接调用 `CJK` 及相关宏包。

```

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

```

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

```

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

```

```

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

```

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

```

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

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

```

zhmCJK 判断结束。

```

537 \fi:

```

`\ctex_CJK_input:n` `\CJK@input` **breqn** 包可能会在正文中将 ~ 的 `\catcode` 改为 12 或 13,这将破坏 CJK 对汉字的首字节的定义 (`\CJK@loadBinding` 和 `\CJK@loadEncoding`)。因此需要确保载入 .enc 和 .bdg 文件时,~ 的 `\catcode` 为 7。

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

580 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

启用中文字符功能。GBK 编码时, 将汉字的首字节设置为活动字符, 并对这些字符初始化; UTF8 编码时, 上游宏包已经处理好。 \CJK@makeActive 应该先于 ctex-name-gbk.cfg 等文件的载入。注意 \CJK@loadBinding 需要调用补丁后的 \CJK@input。使用 zhmCJK 时, 此功能已经被启用。

```

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

```

在导言区结束时调用 \CJK@envStart 启用完整的中文功能。

\CJK@envStart 的定义是

```

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

```

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

```

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

```

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

```

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

```

zhmCJK 判断结束。

```
593 \fi:
```

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

```

594 <@@=>
595 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces

```

恢复名字空间,要把它放在一个 macrocode 环境中,l3doc 才能正确工作。

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

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

```

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

```

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

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

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

```

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

```



```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

`\ctex_punct_space:nn` 定义标点的边界信息。

`\ctexspadef`

```

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

```

载入边界信息文件。

```

657 \ctex_file_input:n { ctexspa.def }

```

658 [⟨pdfTeX⟩](#)

14.3.2 ctex-engine-xetex.def

659 [⟨*xetex⟩](#)

```

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

```

最新版本的 `fontspec` 默认对 `\rmfamily` 和 `\sffamily` 设置 `Ligatures=TeX`, 对 `\ttfamily` 设置 `WordSpace={1,0,0}` 和 `PunctuationSpace=WordSpace`。

```

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

```

14.3.3 ctex-engine-luatex.def

```

669 <*/luatex>

```

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

```

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

```

14.3.3.1 LuaTeX-jā 的默认设置

```

684 \ExplSyntaxOff

```

以下设置抄录自 `ltjdefs.sty`。

```

685 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
686 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
687 \ltjdefcharrange{3}{%
688     "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
689 \ltjdefcharrange{4}{%
690     "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
691     "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
692     "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
693 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
694 \ltjdefcharrange{6}{%
695     "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
696     "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
697 \ltjdefcharrange{7}{
698     "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
699     "AC00-"D7FF}
700 \ltjdefcharrange{8}{ "A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
701 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, -8}}
702 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 `ltj-latex.sty`。

```

703 \directlua{
704     local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
705     luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
706 }
707 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
708     xkanjiskip=.25\zw plus 1pt minus 1pt,

```

```

709 autospacing, autoxspacing, jacharrange={-1},
710 yalbaselineshift=\z@, yjabaselineshift=\z@,
711 jcharwidowpenalty=500, differentjfm=paverage
712 }
713 \ExplSyntaxOn

```

14.3.3.2 LuaTeX-jā 的补丁

```
714 <@@=-ctex_ltx>
```

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

```

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

```

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

```

723 \cs_set_protected:Npn \fontspec_visible_space:
724 {
725   \font_glyph_if_exist:NnTF \font { "2423 }
726   { \ltjalchar "2423 \exp_stop_f: }
727   { \fontspec_visible_space_fallback: }
728 }

```

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。

```

729 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
730 {
731   \group_begin:
732   \cs_set_protected:Npn #1 ##1##2##3
733   { \ltjsetmathletter { ##1 } }
734   #2
735   \group_end:
736 }
737 \ctex_at_end_package:nn { unicode-math }
738 {
739   \cs_if_exist:NTF \um_input_math_symbol_table:
740   {
741     \ctex_ltj_set_math_letter:NN
742     \um_sym:nnn
743     \um_input_math_symbol_table:
744   }
745   {
746     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
747     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
748     { \__um_switchto_literal: }
749     \ctex_ltj_set_math_letter:NN
750     \__um_sym:nnn
751     \__um_input_math_symbol_table:
752   }
753 }

```

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。

```

754 \ctex_at_end_package:nn { listings }
755 {
756   \use:x

```

```

757     {
758       \exp_not:N \RequirePackage { lltpj-listings }
759       \tl_set:Nn \exp_not:N \lstlistingname
760       { \exp_not:o { \lstlistingname } }
761       \tl_set:Nn \exp_not:N \lstlistlistingname
762       { \exp_not:o { \lstlistlistingname } }
763     }
764   }

```

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。
`\CJK@family`

```

765 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
766   {
767     \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
768     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
769   }
770 \tl_new:N \CJK@family
771 \tl_new:N \l__ctex_ltj_current_font_tl
772 \tl_set:Nn \l__ctex_ltj_current_font_tl
773   { \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。

```

774 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
775   {
776     \group_begin:
777     \tl_set_eq:NN \f@encoding \CJK@encoding
778     \tl_set_eq:NN \f@family \CJK@family
779     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
780     \ctex_ltj_pickup_font:
781     \group_end:
782     \font@name
783     \__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。

```

784     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
785     { \__ctex_ltj_select_font_aux: }
786   }
787 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
788   {
789     \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
790     \cs_gset_nopar:Npx \font@name {#1}
791   }
792 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
793   { \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`。

```

794 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
795   {
796     \exp_after:wN \cs_if_exist:NF \font@name
797     {
798       \group_begin:
799       \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
800       \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
801       \define@newfont
802       \group_end:
803     }
804   }
805 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

%%\font\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。

```

806 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
807 {
808   \get@external@font
809   \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
810   {
811     \tl_set:Nx \external@font
812     { \exp_after:wN __ctex_ltj_patch_external_font:w \external@font }
813   }
814   \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。

```

815   \font@name
816   \lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
817   \use:c { \f@encoding + \f@family }
818   \use:c { \curr@fontshape }
819 }

```

\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。

```

820 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
821 {
822   \ctex_ltj_if_alterate_shape_exist:nF { \curr@fontshape }
823   {
824     \group_begin:
825     \tl_set_eq:NN \CJK@family \f@family
826     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
827     {
828       \cs_gset_protected_nopar:Npx \subst@correction
829       {
830         \cs_new_eq:NN

```

```

831             \exp_not:c { \l__ctex_ltj_current_font_tl }
832             \font@name
833         }
834         \group_insert_after:N \group_insert_after:N
835         \group_insert_after:N \subst@correction
836     }
837     \group_end:
838 }
839 }

```

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

```

840 \prg_new_conditional:Npnn \ctex_ltj_if_alter_nate_shape_exist:n #1 { T , F , TF }
841 {
842     \lua_now_x:n { luatexja.jfont.does_alt_set (' \lua_escape_x:n { #1 } ') }
843     \prg_return_true: \else: \prg_return_false: \fi:
844 }

```

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。

```

845 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
846 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

在 `\selectfont` 中更新替代字体。

```

847 \cs_new_protected_nopar:Npn \ctex_ltj_select_alter_nate_font:
848 {
849     \ctex_ltj_if_alter_nate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
850     {
851         \lua_now_x:n
852         {
853             luatexja.jfont.output_alt_font_cmd
854             ('y', ' \lua_escape_x:n { \l__ctex_ltj_current_shape_tl } ')
855         }
856         \lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size' ) }
857     }
858 }
859 \tl_new:N \l__ctex_ltj_current_shape_tl
860 \tl_set:Nn \l__ctex_ltj_current_shape_tl
861 { \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。

```

862 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
863 {
864     \cs_if_exist:cF { #1 / \f@size }
865     {
866         \group_begin:
867         \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
868         \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
869         \ctex_ltj_pickup_font:
870         \group_end:
871         \__ctex_ltj_pop_fontname:
872     }
873 }

```

874 <@@=>

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

```

875 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
876 {
877   \ltj@@getjfontnumber #1
878   \lua_now_x:n
879   {
880     luatexja.jfont.pickup_alt_font_b
881     ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
882   }
883 }
```

14.3.3.4 数学字体族

884 `<@@=ctex_ltj>`

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

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

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

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

```

885 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
886 {
887   \lua_now_x:n
888   { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[~/]+') ) }
889   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
890 }
```

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

```

891 \cs_new:Npn \ctex_ltj_if_jfont_math:nTF #1
892 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
893 \group_begin:
894   \char_set_catcode_other:N M
895   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
896   { \ctex_ltj_if_jfont:nTF {#3} }
897 \group_end:
898 <@@= >
```

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

`\ctex_ltj_get_and_define_fonts:nN`

```

899 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
900 {
901   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
902   { \ctex_ltj_get_and_define_fonts_ja:nN }
903   { \ctex_ltj_get_and_define_fonts_al:nN }
904   {#1} #2
905 }
906 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
907 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
908 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
909 {
910   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
911   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
912   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
913   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
914   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
915   \ctex_ltj_pickup_font:
916   \tl_put_right:Nx \math@fonts
```



```

917     {
918       \ltj@setpar@global
919       \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
920       \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
921       \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
922     }
923   }

924 <@@=-ctex_ltj>

```

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

```

\use@mathgroup
\ctex_ltj_use_math_group:Nn

925 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
926 {
927   \mode_if_math:T
928   {
929     \math@bgroup
930     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
931     \ctex_ltj_math_group_hook:
932     \ctex_ltj_if_jfont_math:NTF #1
933     { \jfam } { \mathgroup } #2 \scan_stop:
934     \math@egroup
935   }
936 }
937 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
938 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

14.3.3.5 字体族的定义与使用

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

```

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

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

```

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

```

\CJK@encoding
\__ctex_ltj_change_encoding:

948 \tl_const:Nn \CJK@encoding { LTJY3 }
949 \DeclareFontEncoding { \CJK@encoding } { } { }
950 \use:x
951 {
952   \exp_not:N \DeclareFontSubstitution
953   { \CJK@encoding } { song } { \mddefault } { \updefault }
954 }
955 \lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
956 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
957 { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
958 \DeclareFontFamily { \CJK@encoding } { song } { }
959 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
960 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
961 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
962 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
963 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
964 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
965 { \CJK@encoding } { song } { \mddefault } { \updefault }
966 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }

```

```

967 { \CJK@encoding } { song } { \bfdefault } { \updefault }
968 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
969 \jfam \c__ctex_ltj_math_fam_int

```

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

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

```

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

```

973 \keys_define:nn { fontspec-preparse-external }
974 {
975   NoEmbed .code:n =
976   { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
977 }
978 \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。

```

979 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
980 {
981   \group_begin:
982   \clist_clear:N \l__ctex_ltj_char_range_clist
983   \seq_clear:N \l__ctex_ltj_alternate_seq
984   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
985   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
986   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
987   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
988   {
989     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
990     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
991     {#1} { \l__ctex_ltj_font_options_clist }
992     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
993     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
994     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
995   }
996   \group_end:
997 }
998 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
999 \clist_new:N \l__ctex_ltj_font_options_clist
1000 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1001 {
1002   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1003   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1004 }

```

`\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
1005 \prop_new:N \g__ctex_ltj_family_name_prop
1006 \prop_new:N \g__ctex_ltj_family_font_name_prop
1007 \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。

```

1008 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1009 {
1010   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1011   {
1012     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1013     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1014     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1015     {
1016       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1017       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1018       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }

```

```

1019         \prop_gremove:Nn \g__ctex_ltj_reset_alterate_prop {#1}
1020     }
1021     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1022 }
1023 }
1024 \tl_new:N \l__ctex_ltj_tmp_tl
1025 \msg_new:nnn { ctex } { redefine-family }
1026 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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

```

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

```

\CJKfamily 切换字体。

```

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

```

\ctex_ltj_family_if_exist:nTF 判断 CJK 字体族 #1 是否存在,若存在则把实际族名保存到 #2 中。

```

1068 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1069 {
1070     \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1071     { \prg_return_true: }
1072     {
1073         \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1074         {
1075             \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl

```

```

1076         \prg_return_true:
1077     }
1078     { \prg_return_false: }
1079 }
1080 }
1081 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

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

\ctex_ltj_fontspec:nn 1120 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1121 {
1122     \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1123     { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1124     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1125     {
1126         \int_gincr:N \g__ctex_ltj_family_int
1127         \__ctex_ltj_fontspec:xnn
1128         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1129         {#1} {#2}
1130     }
1131 }
1132 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1133 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1134 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1135 {
1136     \bool_if:NT \l__ctex_ltj_add_alternate_bool
1137     {
1138         \cs_if_free:cF
1139         { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1140         {

```

```

1141         \cs_gset_eq:cc
1142         { \l__ctex_ltj_altername_cs:n { reset / #1 } }
1143         { \l__ctex_ltj_altername_cs:n { reset / \l_ctex_ltj_family_tl } }
1144         \cs_gset_eq:cc
1145         { \l__ctex_ltj_altername_cs:n { clear / #1 } }
1146         { \l__ctex_ltj_altername_cs:n { clear / \l_ctex_ltj_family_tl } }
1147     }
1148     \bool_set_false:N \l__ctex_ltj_add_altername_bool
1149 }
1150 \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1151 \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1152 \ctex_ltj_switch_family:n {#1}
1153 }
1154 \cs_generate_variant:Nn \l__ctex_ltj_fontspec:nnn { x }
1155 \prop_new:N \g__ctex_ltj_fontspec_prop

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

```

为了支持字体属性可选项在前在后两种语法,给出两个辅助工具,自带展开功能。

```

1181 \cs_new_protected:Npn \l__ctex_ltj_pass_args:nnnn #1#2#3#4
1182 {
1183     \IfNoValueTF {#2}
1184     { \l__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1185     {
1186         \use:x { #1 {#2} {#3} }
1187         #4
1188     }
1189 }
1190 \NewDocumentCommand \l__ctex_ltj_post_arg:w { m m m O { } }
1191 {
1192     \use:x { #1 {#4} {#2} }
1193     #3
1194 }

```

```

\setCJKfamilyfont 1195 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 1196 {
    \CJKfontspec 1197     \l__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 1198     { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1199     { }
1200 }
1201 \NewDocumentCommand \newCJKfontfamily { o m o m }

```

```

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

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

1257 \tl_if_exist:NF \CJKfamilydefault
1258 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1259 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1260 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1261 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1262 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1263 { \CJKfamily { \CJKrmdefault } }
1264 { }
1265 { \ctex_patch_failure:N \rmfamily }
1266 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1267 { \CJKfamily { \CJKsfdefault } }

```

```

1268 { }
1269 { \ctex_patch_failure:N \sffamily }
1270 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1271 { \CJKfamily { \CJKttdefault } }
1272 { }
1273 { \ctex_patch_failure:N \ttfamily }
1274 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1275 { \CJKfamily { \CJKfamilydefault } }
1276 { \cs_set_eq:NN \reset@font \normalfont }
1277 { \ctex_patch_failure:N \normalfont }

```

在导言区末尾更新 \CJKfamilydefault。

```

1278 \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。

```

1279 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1280 {
1281   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1282   {
1283     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1284     {
1285       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1286       { \use:n }
1287       {
1288         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1289         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1290         { \use:n }
1291       }
1292     }
1293     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1294     {
1295       \prop_map_break:n
1296       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1297     }
1298   }
1299 }
1300 \normalfont
1301 \ctex_ltj_update_mathfont:
1302 }
1303 }

```

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

```

1304 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1305 {
1306   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1307   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1308   {
1309     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1310     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1311   }
1312 }
1313 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1314 {
1315   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1316   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1317   { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1318   \cs_if_free:cTF
1319   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1320   {
1321     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1322     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1323   }
1324   {
1325     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }

```



```

1326      { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1327    }
1328  }

```

14.3.3.6 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1329 \keys_define:nn { ctex_ltj / fontspec }
1330 {
1331   AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1332   AlternateFont .value_required:n = true ,
1333   CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1334   CharRange .value_required:n = true
1335 }

```

`\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。

```

1336 \group_begin:
1337   \char_set_catcode_other:N \|
1338   \cs_set:Npn \__ctex_ltj_tmp:w #1
1339   {
1340     \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1341     {
1342       \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1343       {
1344         \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1345         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1346         \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1347         \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1348         { ! \tl_if_blank_p:n { ###1 } }
1349         \seq_concat:NNN \l__ctex_ltj_alternate_seq
1350         \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1351       }
1352     }
1353   }
1354   \char_set_catcode_active:N \|
1355   \__ctex_ltj_tmp:w { || }
1356 \group_end:
1357 \seq_new:N \l__ctex_ltj_tmp_seq
1358 \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。

```

1359 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1360 {
1361   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1362   {
1363     \__ctex_ltj_check_family:n {#1}
1364     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1365     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1366     #3
1367   }
1368   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1369 }

```

`\ctex_ltj_save_alternate_seq:Nn`

保存由 `AlternateFont` 设置的替代字体序列。

`\ctex_ltj_save_alternate_seq:Nnnwn`

```

1370 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1371 {
1372   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1373   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1374 }
1375 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }

```

```

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

```

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

```

1397 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1398 {
1399   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1400   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1401   \ctex_ltj_set_alternate_family:coonn
1402   { \__ctex_ltj_alternate_cs:n {#1} }
1403   { \l__ctex_ltj_char_range_clist }
1404   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1405 }
1406 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1407 {
1408   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1409   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1410   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1411 }
1412 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

\ctex_ltj_save_alternate_family:Nnnn 保存替代字体序列的定义,以备定义主字体时使用。

```

1413 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1414 {
1415   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1416   \cs_gset_protected_nopar:Npx #1
1417   { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1418 }

```

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

```

1419 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1420 {
1421   \group_begin:
1422   \__ctex_ltj_change_encoding:
1423   \cs_set_eq:NN \CJKfamily \use_none:n
1424   \ctex_ltj_swap_cs:NN
1425   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1426   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1427   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1428   \group_end:
1429 }
1430 \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。

```

1431 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1432 {
1433   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1434   \cs_set_eq:NN #1 #2
1435   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1436   \cs_undefine:N \__ctex_ltj_tmp:w
1437 }

```

`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。

```

1438 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1439 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1440 {
1441   \int_gincr:N \g__ctex_ltj_family_int
1442   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1443 }
1444 \int_new:N \g__ctex_ltj_family_int

```

`\ctex_ltj_declare_alter_nate_shape:nnnnnn`

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

```

1445 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1446 {
1447   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1448   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1449   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1450   { \l_fontspec_family_tl } {#3} {#4}
1451 }

```

`\ctex_ltj_set_alter_nate_shape:Nnnnnnn`

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

```

1452 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1453 {
1454   \clist_map_inline:Nn #1
1455   {
1456     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1457     {
1458       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1459       \l__ctex_ltj_char_range_tl
1460     }
1461     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1462   }
1463   \__ctex_ltj_save_alter_nate_shape:cn
1464   { \__ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1465   { luatexja.jfont.clear_alt_font_latex }
1466   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1467 }

```

`\ctex_ltj_set_alter_nate_shape:nnn`

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

```

1468 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1469 { m m > { \SplitArgument { \c_one } { -> } } m }
1470 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1471 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1472 {
1473   \ctex_ltj_set_alter_nate_shape:n
1474   {
1475     \IfNoValueTF {#4}

```

```

1476         { \int_eval:n {#3} , \int_eval:n {#3} , }
1477         {
1478             \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1479             \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1480         }
1481         '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1482         '\lua_escape_x:n { \CJK@encoding/#1 }'
1483     }
1484 }
1485 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1486 {
1487     \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1488     \__ctex_ltj_save_alternate_shape:cnm
1489     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1490     { luatexja.jfont.set_alt_font_latex } {#1}
1491 }

```

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

```

1492 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1493 {
1494     \tl_map_inline:Nn #3
1495     {
1496         \ctex_ltj_set_alternate_shape:n
1497         {
1498             ##1 ,
1499             '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1500             '\lua_escape_x:n { \CJK@encoding/#1 }'
1501         }
1502     }
1503 }

```

__ctex_ltj_save_alternate_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。

```

1504 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1505 {
1506     \group_begin:
1507     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1508     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1509     \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1510     \cs_gset_protected_nopar:Npx #1
1511     { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1512     \group_end:
1513 }
1514 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nnn { c }

```

clearalternatefont
resetalternatefont

清除和重置操作总是全局的。

```

1515 \keys_define:nn { ctex }
1516 {
1517     clearalternatefont .code:n =
1518     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1519     resetalternatefont .code:n =
1520     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1521     clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1522     resetalternatefont .default:n = \l_ctex_ltj_family_tl
1523 }
1524 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1525 {
1526     \group_begin:
1527     \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1528     {
1529         \cs_if_exist_use:cT { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1530         {

```

```

1531         \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1532         {#1} { \l__ctex_ltj_base_family_tl }
1533         \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1534         \selectfont
1535     }
1536 }
1537 { \__ctex_ltj_family_unknown_warning:n {#1} }
1538 \group_end:
1539 }
1540 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1541 {
1542     \group_begin:
1543     \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1544     {
1545         \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1546         \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1547         \selectfont
1548     }
1549     \group_end:
1550 }
1551 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1552 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

declarecharrange 预先声明字符范围。

```

1553 \keys_define:nn { ctex }
1554 {
1555     declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1556     declarecharrange .value_required:n = true
1557 }
1558 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1559 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1560 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1561 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1562 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

`\ctex_ltj_declare_char_range:nn` #1 是名字, #2 是范围。

`\g__ctex_ltj_char_range_prop`

```

1563 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1564 {
1565     \tl_clear:N \l__ctex_ltj_char_range_tl
1566     \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1567     \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1568     \ctex_ltj_def_char_range_key:n {#1}
1569     \tl_clear:N \l__ctex_ltj_char_range_tl
1570 }
1571 \tl_new:N \l__ctex_ltj_char_range_tl
1572 \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。

```

1573 \NewDocumentCommand \ctex_ltj_save_char_range:n
1574 { > { \SplitArgument { \c_one } { -> } } m }
1575 { \ctex_ltj_save_char_range:nn #1 }
1576 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1577 {
1578     \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1579     { {
1580         \IfNoValueTF {#2}
1581         { \int_eval:n {#1} , \int_eval:n {#1} }
1582         {
1583             \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1584             \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1585         }
1586     } }
1587 }

```

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

```

1588 \cs_new_protected_nopar:Npn \ctex_ljt_def_char_range_key:n #1
1589 {
1590   \keys_if_exist:nnF { ctex_ljt / fontspec } {#1}
1591   {
1592     \keys_define:nn { ctex_ljt / fontspec }
1593     { #1 .code:n = \ctex_ljt_char_range_key:nn {#1} { ##1 } }
1594   }
1595 }

```

`\ctex_ljt_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。

```

1596 \cs_new_protected:Npn \ctex_ljt_char_range_key:nn #1#2
1597 {
1598   \tl_if_blank:nTF {#2}
1599   { \tl_set:Nn \l__ctex_ljt_char_range_clist {#1} }
1600   {
1601     \clist_if_empty:NT \l__ctex_ljt_char_range_clist
1602     {
1603       \tl_set:Nn \l__ctex_ljt_tmp_tl { {#1} }
1604       \__ctex_ljt_char_range_parse_feature:w #2 \q_stop
1605     }
1606   }
1607 }

```

`__ctex_ljt_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]`。

```

1608 \NewDocumentCommand \__ctex_ljt_char_range_parse_feature:w
1609 { +o o u { \q_stop } }
1610 {
1611   \exp_args:NNf \tl_put_right:Nn \l__ctex_ljt_tmp_tl
1612   {
1613     \IfNoValueTF {#1} { {#3} }
1614     {
1615       \IfNoValueTF {#2}
1616       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] { #3 } } }
1617       { [ { #1 } ] { [ #2 ] } }
1618     }
1619   }
1620   \seq_put_right:No \l__ctex_ljt_alternate_seq { \l__ctex_ljt_tmp_tl }
1621 }

```

14.3.3.7 其它设置

在抄录环境中禁用 `autospacing` 和 `autoxspacing`。然而, `LuaTeX-jā` 还是会使 `Jāchar` 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 `Jāchar` 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
}

1622 \AtBeginDocument
1623 {

```

```

1624 \ctex_appto_cmd:NnnTF \verbatim@font
1625 { \char_set_catcode_letter:n { 64 } }
1626 { \CTEX@verbatim@font@hook }
1627 { }
1628 { \ctex_patch_failure:N \verbatim@font }
1629 }
1630 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1631 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1632 <@@=)

```

\@@italiccorr L^AT_EX 的倾斜校正也要重新定义。

```

1633 \cs_set_eq:NN \@@italiccorr \
1634 <@@=ctex>
1635 </luatex>

```

14.3.4 ctex-engine-uptex.def

```
1636 <*uptex|aptex>
```

按 CJK 的命名习惯模拟部分命令,并设置默认字体。

```

1637 \tl_set:Nn \CJKrmdefault { rm }
1638 \tl_set:Nn \CJKsfdefault { sf }
1639 \tl_set:Nn \CJKttdefault { tt }
1640 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1641 \tl_put_left:Nn \rmfamily { \kanjifamily { \CJKrmdefault } }
1642 \tl_put_left:Nn \sffamily { \kanjifamily { \CJKsfdefault } }
1643 \tl_put_left:Nn \ttfamily { \kanjifamily { \CJKttdefault } }
1644 \NewDocumentCommand \CJKfamily { m }
1645 {
1646   \kanjifamily { #1 }
1647   \selectfont
1648 }
1649 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```

在导言区末尾更新 \CJKfamilydefault。

```
1650 \ctex_at_end_preamble:n { \ctex_update_default_family: }
```

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

```

1651 \ctex_patch_cmd_once:NnnTF \em
1652 { \ExplSyntaxOff }
1653 { \eminnershape \else \gtfamily \itshape }
1654 { \eminnershape \else \itshape }
1655 { }
1656 {
1657   \ctex_patch_cmd:Nnn \em
1658   { \mcfamily \upshape \else \gtfamily \itshape }
1659   { \eminnershape \else \itshape }
1660 }
1661 \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。

```

1662 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1663 {
1664   \DeclareKanjiFamily{JY2}{#1}{#2}{#3}

```

```

1665 \DeclareKanjiFamily{JT2}{#1}{ }
1666 \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{ }
1667 \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{ }
1668 \tl_if_empty:nF { #3 }
1669 {
1670   \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{ }
1671   \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{ }
1672 }
1673 }
1674 \@onlypreamble \ctex_set_upfamily:nnn

```

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

```

1675 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1676 {
1677   \ctex_set_zhmap:n
1678   {
1679     \special{ pdf:mapline~ #1-h~ UniGB-UTF16-H~ #2 }
1680     \special{ pdf:mapline~ #1-v~ UniGB-UTF16-V~ #2 }
1681     \tl_if_empty:nF { #3 }
1682     {
1683       \special{ pdf:mapline~ #1b-h~ UniGB-UTF16-H~ #3 }
1684       \special{ pdf:mapline~ #1b-v~ UniGB-UTF16-V~ #3 }
1685     }
1686   }
1687 }
1688 \@onlypreamble \ctex_set_upmap:nnn

```

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

```

1689 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1690 {
1691   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1692   \ctex_set_upmap:nnn { upserifit } { #3 } { }
1693   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1694   \ctex_set_upmap:nnn { upmono } { #6 } { }
1695 }
1696 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

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

```

1697 \RequirePackage { pxeveryysel }
1698 </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。

```

1699 <*pdf|luatex|uptex|aptex>
1700 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1701 {
1702   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1703   {
1704     \group_begin:
1705     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1706     \tl_gset:Nx \CJKfamilydefault
1707     {
1708       \str_case:onF { \familydefault }
1709       {

```



```

1710             { \rmdefault } { \exp_not:N \CJKrmdefault }
1711             { \sfdefault } { \exp_not:N \CJKsfdefault }
1712             { \ttdefault } { \exp_not:N \CJKttdefault }
1713         }
1714         { \CJKfamilydefault }
1715     }
1716     \group_end:
1717 }

```

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

```

1718 <*\uotex>
1719     \ctex_ltj_ensure_default_family:
1720 </\uotex>
1721 }

```

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

```

1722 \tl_new:N \l__ctex_family_default_init_tl
1723 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1724 \tl_set:Nx \l__ctex_family_default_init_tl
1725 {
1726     \exp_not:N \__ctex_family_default_wrap:n
1727     { \exp_not:o { \CJKfamilydefault } }
1728 }
1729 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1730 </pdfAtex|luaAtex|uptAtex|aptAtex>

```

14.3.6 操作系统的判断

\ctex_detected_platform: 在 Lua^AT_EX 下直接用调用 os.name 来判断。

```

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

```

Ap^AT_EX 可以使用 \ngostype 来判断。

```

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

```

```

1761     }
1762   }
1763 </aptex>

```

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

```

1764 <*xetex|pdfTeX|uptex>
1765 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1766 {
1767   \file_if_exist:nTF { /dev/null }
1768   {
1769     \file_if_exist:nTF { nul: }
1770     {
1771       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1772       { \ctex_if_macosx:TF { mac } { windows } }
1773       { \ctex_if_macosx:TF { mac } { fandol } }
1774     }
1775     { \ctex_if_macosx:TF { mac } { fandol } }
1776   }
1777   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1778 }
1779 \str_const:Nx \c__ctex_upper_case_file_str
1780 { \exp_args:No \str_upper_case:n { \g_file_current_name_tl } }

```

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

```

1781 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1782 {
1783   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1784   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1785   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1786 }
1787 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1788 </xetex|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。

```

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

```

在 pdfTeX 下使用 GBK 编码, DVIPDFMx 驱动可以直接用它的 \special 命令, 其它模式用 xCJK2uni 宏包处理。使用 UTF-8 编码时, CJKutf8 已经处理了书签问题, 但仍需要设置 pdfencoding 为 unicode, 目的是在书签的开头写入 BOM (\376\377), 提示这是 UTF-16BE

字节流。

```

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

```

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

```

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

```

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

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

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

```

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

```

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

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

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

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

```

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

1829 < *pdfTeX|xetex|luatex|uptex|aptex >

```

14.3.8 CJKfntef、xeCJKfntef 相关设置

$\text{CT}_{\text{E}}\text{X}$ 宏集对 pdfTeX 与 $\text{X}_{\text{E}}\text{TeX}$ 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 $\backslash\text{CTEX}$ 开头的宏定义是过时命令，仅做兼容性保留。

```

1830 < *pdfTeX >
1831 \RequirePackage { CJKfntef }
1832 \normalem
1833 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1834 { \tl_clear:c { CJK#1color } }
1835 < /pdfTeX >
1836 < *xetex >
1837 \RequirePackage { xeCJKfntef }
1838 \ifpackagelater { xeCJKfntef } { 2014/11/04 }
1839 {
1840   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1841   { \xeCJKsetup { #1 / format = { } } }
1842 }
1843 {
1844   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1845   { \tl_clear:c { CJK#1color } }
1846 }
1847 < /xetex >
1848 < *luatex|uptex|aptex >
1849 \msg_new:nnn { ctex } { fntef-not-available }
<luatex> 1850 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
<uptex> 1851 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
<aptex> 1852 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1853 \msg_warning:nn { ctex } { fntef-not-available }
1854 < /luatex|uptex|aptex >
1855 \clist_map_inline:nn
1856 { underdot , underline , underdblline , underwave , sout , xout }
1857 < *pdfTeX|xetex >
1858 {
1859   \__ctex_clear_fntef_color:n {#1}
1860   \cs_new_protected_nopar:cpx { CTEX#1 }
1861   {
1862     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1863     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1864     \exp_not:c { CJK#1 }
1865   }
1866 }
1867 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1868 {
1869   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1870   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1871   \CJKfilltwosides
1872 }
1873 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1874 < /pdfTeX|xetex >
1875 < *luatex|uptex|aptex >
1876 { \cs_new_eq:cN { CTEX#1 } \use:n }

```

```

1877 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1878 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1879 </luatex|uptex|aptex>
1880 <*pdfTeX>
1881 \clist_map_inline:nn
1882 {
1883     underdotbasesep , underdotsep , underlinebasesep ,
1884     underlinesep , underdbllinesep , underdbllinebasesep ,
1885     underwavebasesep , underwavesep , southeight ,
1886     underdotcolor , underwavecolor , underlinicolor ,
1887     underdbllinecolor , soutcolor , xoutcolor
1888 }
1889 {
1890     \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1891     \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1892 }
1893 </pdfTeX>

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1894 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1895 <*pdfTeX|xetex>
1896 {
1897     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1898     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1899 }
1900 </pdfTeX|xetex>
1901 <*luatex>
1902 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1903 </luatex>
1904 <*uptex|aptex>
1905 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1906 </uptex|aptex>
1907 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1908 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1909 <*pdfTeX>
1910 {
1911     \cs_set_protected_nopar:Npn \CJKglue
1912     { \skip_horizontal:N \l__ctex_ccglue_skip }
1913 }
1914 </pdfTeX>
1915 <*xetex>
1916 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1917 </xetex>
1918 <*luatex>
1919 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1920 </luatex>
1921 <*uptex|aptex>
1922 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1923 </uptex|aptex>
1924 \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

```

1925 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1926 {
1927     <*pdfTeX|xetex>
1928     \if_meaning:w \CJKglue \l__ctex_ccglue:
1929     \prg_return_false: \else: \prg_return_true: \fi:
1930 </pdfTeX|xetex>
1931 <*luatex>
1932 \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1933 { \prg_return_false: } { \prg_return_true: }

```

```

1934 </luatex>
1935 <*uptex|aptex>
1936   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1937   { \prg_return_false: } { \prg_return_true: }
1938 </uptex|aptex>
1939 }

```

注意下面的标记不能用 %<pdf_{tex}|xet_{ex}>, 它会导致旧版本的 l3docstrip 不能替换 @@。

```

1940 <*pdftex|xetex>
1941 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1942 </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 pdf_T_EX 和 X_Y_T_EX 下, 直接使用 `\f@size\p@` 作为汉字的宽度, 这应该对大多数汉字字体都成立, 但不适用于诸如“方正兰亭黑长”之类的特殊字体。在 X_Y_T_EX 可以用 `\fontcharwd` 来改进。而在 pdf_T_EX 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 Lua_T_EX 下, Lua_T_EX-ja 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 `\zw` 作为汉字宽度。up_T_EX 可以直接使用原生的长度单位 `zw`。

```

1943 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
<pdftex|xetex> 1944 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1945 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1946 { \dim_set:Nn \ccwd { 1zw } }

```

14.3.10 其它

`\ctex_add_to_selectfont:n` `\CTEX@selectfont@hook` `\EverySelectfont` 直到文档开始时才有效。为了 `\ccwd` 和 Lua_T_EX-ja 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。every_{sel} 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。pxevery_{sel} 宏包取消了检查, 但也需要恢复定义, 避免重复使用钩子。

```

1947 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1948 {
1949   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1950   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1951 }
1952 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1953 \if_cs_exist:N \@EverySelectfont@Init
1954   \group_begin:
1955   \cs_set:Npn \__ctex_tmp:N #1
1956   {
1957     \tl_set:Nn \l__ctex_tmp_tl {#1}
1958     \cs_new_eq:NN \CTEX@selectfont@save #1
1959     \cs_new_protected_nopar:Npn \__ctex_restore_selectfont:
1960     {
1961       \tl_put_left:Nn \@EverySelectfont@Init
1962       { \let #1 \CTEX@selectfont@save }
1963       \cs_undefine:N \__ctex_restore_selectfont:
1964     }
1965   }
1966   \ctex_parse_name:NN \__ctex_tmp:N \selectfont
1967   \exp_last_unbraced:NNo \group_end:
1968   \ctex_preto_cmd:NnnTF { \l__ctex_tmp_tl } { \ExplSyntaxOff }
1969   { \CTEX@selectfont@hook }
1970   { \__ctex_restore_selectfont: }
1971   { \ctex_patch_failure:N \selectfont }
1972 \fi:

```

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

```

1973 <*pdfTeX>
1974 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1975 </pdfTeX>
1976 <*xetex|luatex|uptex|aptex>
1977 \EverySelectfont { \CTEX@selectfont@hook }
1978 </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。

```

1979 <*luatex>
1980 \ctex_add_to_selectfont:n
1981 {
1982     \ctex_ltj_select_font:
1983     \ctex_ltj_select_alternate_font:
1984 }
1985 \tl_set:Nn \CJK@family { song } \selectfont
1986 \tl_clear:N \CJK@family
1987 </luatex>

```

`\ctex_update_xkanjiskip:` `upTeX` 和 `LuaTeX-j` 对 `\xkanjiskip` 都是即时赋值。单位 `zw` 与字体相关,因此需要每次 `\selectfont` 的时候更新一次 `\xkanjiskip`。如果用户设置过 `\xkanjiskip`,就不更新。注意,同 `TeX` 的 `\baselineskip` 一样,如果在一个段落内多次设置了 `\kanjiskip` 或 `\xkanjiskip`,只有最后的设置会影响全段。

```

1988 <*luatex|uptex|aptex>
1989 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
1990 {
1991     \skip_if_eq:nnT
<luatex> 1992     { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
<uptex|aptex> 1993     { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
1994     {
1995         \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
<luatex> 1996         \ltjsetxkanjiskip \l__ctex_xkanjiskip_skip
<uptex|aptex> 1997         \skip_set_eq:NN \ptex_xkanjiskip:D \l__ctex_xkanjiskip_skip
1998     }
1999 }
2000 \tl_new:N \l__ctex_xkanjiskip_tl
2001 \tl_set:Nn \l__ctex_xkanjiskip_tl
<luatex> 2002 { .25\zw plus 1pt minus 1pt }
<uptex|aptex> 2003 { .25zw plus 1pt minus 1pt }
2004 \skip_new:N \l__ctex_xkanjiskip_skip
2005 \skip_set:Nn \l__ctex_xkanjiskip_skip
<luatex> 2006 { \ltjgetparameter { xkanjiskip } }
<uptex|aptex> 2007 { \ptex_xkanjiskip:D }

2008 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2009 </luatex|uptex|aptex>

```

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

```

2010 \keys_define:nn { ctex }
2011 {
2012 <*pdfTeX|xetex>
2013     space .choice: ,
2014     space / true .code:n =
<pdfTeX> 2015     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<xetex> 2016     { \xeCJKsetup { CJKspace = true } } ,
2017     space / auto .code:n =
<pdfTeX> 2018     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
<xetex> 2019     { \xeCJKsetup { CJKspace = false } } ,
2020     space / false .code:n =
<pdfTeX> 2021     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,

```

```

2022 { \xeCJKsetup { CJKspace = false } } ,
2023 space .default:n = { true } ,
2024 space .initial:n = { auto }
2025 </pdf|xetex>
2026 <*/luatex|uptex|aptex>
2027 space .code:n =
2028 { \msg_warning:nn { ctex } { invalid-option } }
2029 </luatex|uptex|aptex>
2030 }

```

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

```

2031 \keys_define:nn { ctex }
2032 {
2033   punct .code:n =
2034   {
2035     \tl_set:Nx \l__ctex_punct_tl { #1 }
2036     \punctstyle { \l__ctex_punct_tl }
2037     \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2038     \ctex_mono_jfm:o { \l__ctex_punct_tl }
2039     \msg_warning:nn { ctex } { invalid-option }
2040   } ,
2041   punct .default:n = { quanjiao } ,
2042 }

```

X_YLaTeX、LuaLaTeX 和 upLaTeX 总是使用 UTF8 编码。

```

2043 <*/xetex|luatex|uptex|aptex>
2044 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2045 </xetex|luatex|uptex|aptex>
2046 </pdf|xetex|luatex|uptex|aptex>
2047 <*/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。

```

2048 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.4 用户设置接口

```

2049 <*/class|ctex|ctexheading>
2050 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
2051 </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

```

2052 <*/class|ctex>
2053 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
2054 {
2055   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
2056   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
2057   \IfNoValueF {#1} { \keys_set:nn { ctex } / #2 } {#1} }
2058 }
2059 \NewDocumentCommand \CTEXoptions { +o }
2060 {
2061   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
2062   { \ctexset~ {~ #1~ }~ is~ set. }
2063   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
2064 }
2065 </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。

```

2066 \keys_define:nn { ctex }
2067 {
2068   autoindent .choice: ,
2069   autoindent .default:n = { true } ,
2070   autoindent / true .code:n =
2071   {
2072     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2073     \ctex_select_size:
2074   } ,
2075   autoindent / false .code:n =
2076   { \tl_clear:N \l__ctex_autoindent_tl } ,
2077   autoindent / unknown .code:n =
2078   {
2079     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2080     \ctex_select_size:
2081   }
2082 }

```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2083 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2084 \cs_new_protected_nopar:Npn \ctex_select_size:
2085 { \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。

```

2086 \cs_new_protected_nopar:Npn \ctex_update_size:
2087 {
2088   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2089   {
2090     \ctex_update_stretch:
2091     \ctex_update_parindent:
2092   }
2093   { \ctex_update_ziju: }
2094 }
2095 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2096 \tl_new:N \l__ctex_ziju_tl
2097 \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。

```

2098 \ctex_add_to_selectfont:n
2099 { \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。

```

2100 \keys_define:nn { ctex }
2101 {
2102   linestretch .code:n =
2103   {
2104     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2105     \ctex_select_size:
2106   } ,
2107   linestretch .value_required:n = true
2108 }
2109 \tl_new:N \l__ctex_line_stretch_tl
2110 \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。

```

2111 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2112 {
2113   \ctex_update_em_unit:
2114   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2115   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2116   {
2117     \ctex_if_ccglue_touched:TF
2118     { \ctex_update_ccwd: }
2119     {
2120       \dim_set:Nn \l__ctex_tmp_dim
2121       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2122       \skip_set:Nn \l__ctex_ccglue_skip
2123       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2124       \ctex_update_ccglue:
2125     }
2126   }
2127   {
2128     \int_set:Nn \l__ctex_tmp_int
2129     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2130     \tex_divide:D \l__ctex_tmp_int \ccwd
2131     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2132     {
2133       \skip_set:Nn \l__ctex_ccglue_skip
2134       {
2135         \c_zero_dim plus \dim_eval:n
2136         {
2137           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2138           \l__ctex_tmp_int
2139         }
2140       }
2141     }
2142     { \skip_zero:N \l__ctex_ccglue_skip }
2143     \ctex_update_ccglue:
2144   }
2145 }

```

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

```

2146 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2147 {
2148   \tl_if_empty:NF \l__ctex_autoindent_tl
2149   {
2150     \dim_compare:nNnF \parindent = \c_zero_dim
2151     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2152   }
2153 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2154 \NewDocumentCommand \ziju { m }
2155 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2156 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2157 {
2158   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2159   \ctex_select_size:
2160 }

```

`\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。

```
2161 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2162 {
2163   \ctex_update_em_unit:
2164   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2165   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2166   \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。

```
2167 {
2168   \dim_set:Nn \l__ctex_tmp_dim
2169   { \linewidth - \ccwd + \l__ctex_ziju_dim }
2170   \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2171   \tex_divide:D \l__ctex_tmp_int \ccwd
2172   \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。

```
2173   \dim_compare:nNnF \parindent = \c_zero_dim
2174   {
2175     \int_compare:nNnF \l__ctex_tmp_int < \c_three
2176     { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2177   }
2178   \skip_set:Nn \l__ctex_ccglue_skip
2179   {
2180     \l__ctex_ziju_dim
2181     plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2182     minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2183     { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2184   }
2185 }
2186 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2187 \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 $2\ccwd$ 。

```
2188   \ctex_update_parindent:
2189 }
2190 \dim_new:N \l__ctex_ziju_dim
```

`\CTEXindent`
`\CTEXnoindent`

过时命令。

```
2191 \NewDocumentCommand \CTEXindent { }
2192 {
2193   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2194   { \parindent is~ set~ to~ 2\ccwd. }
2195   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2196 }
2197 \NewDocumentCommand \CTEXnoindent { }
2198 {
2199   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2200   { \parindent is~ set~ to~ 0pt. }
2201   \dim_zero:N \parindent
2202 }
```

14.6 中文数字与日期

```
2203 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2204 \RequirePackage { zhnumber }
```

```

\chinese 2205 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2206 \cs_new_eq:NN \@chinese \@zhnum
2207 \cs_new_eq:NN \Chinese \chinese
2208 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

```

2209 \ctex_at_end_package:nn { enumitem }
2210 {
2211   \cs_if_free:NF \AddEnumerateCounter
2212   {
2213     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2214     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2215     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2216   }
2217 }

\CTEXnumber 2218 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 2219 { \protected@edef #1 { \zhnumber {#2} } }
2220 \NewDocumentCommand \CTEXdigits { m m }
2221 { \protected@edef #1 { \zhdigits {#2} } }

today 2222 \cs_set_eq:NN \CTEX@todayold \today
2223 \keys_define:nn { ctex }
2224 {
2225   today .choice: ,
2226   today / old .code:n =
2227     { \cs_set_eq:NN \today \CTEX@todayold } ,
2228   today / small .code:n =
2229     {
2230       \cs_set_eq:NN \today \zhtoday
2231       \zhnumsetup { time = Arabic }
2232     } ,
2233   today / big .code:n =
2234     {
2235       \cs_set_eq:NN \today \zhtoday
2236       \zhnumsetup { time = Chinese }
2237     } ,
2238   today / unknown .code:n =
2239     { \msg_error:nxx { ctex } { today-undef } {#1} }
2240 }
2241 \msg_new:nnnn { ctex } { today-undef }
2242 { Today~format~`#1'~is~undefined. }
2243 { 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。

```

2244 \tl_if_exist:NF \proofname
2245 {
2246   \tl_new:N \proofname
2247   \tl_set:Nn \proofname { Proof }
2248 }

2249 \keys_define:nn { ctex }
2250 {
2251   contentsname .tl_set:N = \contentsname ,
2252   listfigurename .tl_set:N = \listfigurename ,
2253   listtablename .tl_set:N = \listtablename ,
2254   figurename .tl_set:N = \figurename ,
2255   tablename .tl_set:N = \tablename ,
2256   abstractname .tl_set:N = \abstractname ,
2257   indexname .tl_set:N = \indexname ,
2258   appendixname .tl_set:N = \appendixname ,
2259   proofname .tl_set:N = \proofname ,
<article> 2260 bibname .tl_set:N = \refname

```

```

2261 bibname .tl_set:N = \bibname
2262 <*beamer>
2263 algorithmname .tl_set:N = \algorithmname ,
2264 bibname .tl_set:N = \bibname ,
2265 refname .tl_set:N = \refname ,
2266 continuation .tl_set:N = \insertcontinuationtext
2267 </beamer>
2268 }

2269 <*ctex>
2270 \msg_new:nnn { ctex } { ctexbibname }
2271 {
2272   Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.~\
2273   The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2274 }
2275 \tl_if_exist:NTF \insertcontinuationtext
2276 {
2277   \keys_define:nn { ctex }
2278   {
2279     algorithmname .tl_set:N = \algorithmname ,
2280     bibname .tl_set:N = \bibname ,
2281     refname .tl_set:N = \refname ,
2282     continuation .tl_set:N = \insertcontinuationtext
2283   }
2284 }
2285 {
2286   \tl_if_exist:NTF \bibname
2287   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2288   {
2289     \tl_if_exist:NTF \refname
2290     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2291     {
2292       \msg_warning:nn { ctex } { ctexbibname }
2293       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2294     }
2295   }
2296 }
2297 </ctex>

2298 </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。

```
2299 <*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。

```

2300 <*article|book|report>
2301 \seq_new:N \c__ctex_section_headings_seq
2302 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2303 { section , subsection , subsubsection , paragraph , subparagraph }
2304 </article|book|report>

```

```

\c__ctex_headings_seq 2305 \seq_new:N \c__ctex_headings_seq
2306 <*article|book|report>
2307 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2308 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2309 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2310 </article|book|report>
2311 <*beamer>
2312 \seq_gset_from_clist:Nn \c__ctex_headings_seq

```

```

2313 { part , section , subsection }
2314 </beamer>

```

```

__ctex_initial_heading:n 2315 \cs_new_protected_nopar:Npn __ctex_initial_heading:n #1
2316 {
2317   \tl_new:c { CTEX@pre#1 }
2318   \tl_new:c { CTEX@post#1 }
2319   \tl_const:cx { CTEXthe#1 }
2320   {
2321     \exp_not:c { CTEX@pre#1 }
2322     \exp_not:c { CTEX@the#1 }
2323     \exp_not:c { CTEX@post#1 }
2324   }
2325   \tl_const:cx { CTEX@#1name }
2326   {
2327     \group_begin:
2328     \exp_not:c { CTEX@#1@nameformat }
2329     {
2330       \exp_not:c { CTEX@pre#1 }
2331       \exp_not:N \tl_if_empty:NTF
2332       \exp_not:c { CTEX@#1@numberformat }
2333       { \exp_not:c { CTEX@the#1 } }
2334       {
2335         \group_begin:
2336         \exp_not:c { CTEX@#1@numberformat }
2337         \exp_not:c { CTEX@the#1 }
2338         \group_end:
2339       }
2340       \exp_not:c { CTEX@post#1 }
2341     }
2342   \group_end:
2343 }
2344 }

```

```

__ctex_def_heading_keys:n 2345 \cs_new_protected_nopar:Npn __ctex_def_heading_keys:n #1
2346 {
2347   \tl_put_right:Nx \l__ctex_tmp_tl
2348   {
2349     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2350     #1 / name .code:n =
2351     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2352     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2353     #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2354     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2355     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2356     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2357     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2358     #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2359     #1 / before skip .tl_set:N = \exp_not:c { CTEX@#1@before skip } ,
2360     #1 / after skip .tl_set:N = \exp_not:c { CTEX@#1@after skip } ,
2361     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2362     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2363     <*article|book|report>
2364     #1 / after indent .bool_set:N = \exp_not:c { CTEX@#1@after indent } ,
2365     #1 / fix skip .bool_set:N = \exp_not:c { CTEX@#1@fix skip } ,
2366     #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2367     #1 / hang .initial:n = true ,
2368     #1 / run in .bool_set:N = \exp_not:c { CTEX@#1@run in } ,
2369     #1 / break .tl_set:N = \exp_not:c { CTEX@#1@break } ,
2370     #1 / break+ .code:n =
2371     { \tl_put_right:Nn \exp_not:c { CTEX@#1@break } { #####1 } } ,
2372     #1 / to cline .code:n =
2373     {
2374       \cs_set:Npn \exp_not:c { CTEX@#1@to cline }
2375       \exp_not:n { #####1#####2 } { #####1 }
2376     } ,

```

```

2377 </article|book|report>
2378     #1 / format+ .code:n =
2379     { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2380     #1 / nameformat+ .code:n =
2381     { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2382     #1 / numberformat+ .code:n =
2383     { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2384     #1 / aftername+ .code:n =
2385     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2386     #1 / titleformat+ .code:n =
2387     { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2388     #1 / aftertitle+ .code:n =
2389     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2390     #1 / numbering .initial:n = true ,
2391     #1 / beforeskip .initial:n = \c_zero_skip ,
2392     #1 / afterskip .initial:n = \c_zero_skip ,
2393     #1 / indent .initial:n = \c_zero_dim ,
2394     #1 / beforeskip .value_required:n = true ,
2395     #1 / afterskip .value_required:n = true ,
2396     #1 / indent .value_required:n = true ,
2397 }
2398 }

```

```

\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。

```

2399 \NewDocumentCommand \ctex_assign_heading_name:nn
2400 { m > { \SplitArgument { \c_one } { , } } +m }
2401 { \__ctex_assign_heading_name:nnn {#1} {#2} }
2402 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2403 {
2404   \tl_set:cn { CTEX@pre#1 } {#2}
2405   \IfNoValueTF {#3}
2406   { \tl_clear:c { CTEX@post#1 } }
2407   { \tl_set:cn { CTEX@post#1 } {#3} }
2408 }

```

```

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

```

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2409 \group_begin:
2410 <*book|report>
2411 \tl_set:Nn \l__ctex_tmp_tl
2412 {
2413   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2414   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2415   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2416   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2417   chapter / lofskip .initial:n = \c_zero_skip ,
2418   chapter / lotskip .initial:n = \c_zero_skip ,
2419   chapter / lofskip .value_required:n = true ,
2420   chapter / lotskip .value_required:n = true ,
2421 }
2422 </book|report>
2423 <*article|beamer>
2424 \tl_clear:N \l__ctex_tmp_tl
2425 </article|beamer>

```

定义标题键值选项。

```

2426 \seq_map_inline:Nn \c__ctex_headings_seq
2427 {
2428   \__ctex_initial_heading:n {#1}
2429   \__ctex_def_heading_keys:n {#1}

```

```

2430 }
2431 \use:x
2432 {
2433   \group_end:
2434   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2435 }
2436 <*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。

```

2437 <*book|report>
2438 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2439 {
2440   \CTEX@fixheadingskip
2441   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2442     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2443 }
2444 </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。

```

2445 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2446 {
2447   \par
2448   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2449   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2450 }
2451 \skip_new:N \l__ctex_heading_skip
2452 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2453 { \skip_set:Nn \l__ctex_heading_skip }
2454 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\partmark` 提供 `\partmark`。

```

2455 \ProvideDocumentCommand \partmark { m }
2456 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2457 \cs_new_eq:NN \CTEXifname \use_ii:nn
2458 \cs_new_protected_nopar:Npn \CTEX@ifnametrue
2459 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2460 \cs_new_protected_nopar:Npn \CTEX@ifnamefalse
2461 { \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。

```

2462 <*book|report>
2463 \cs_new_protected_nopar:Npn \CTEX@addloflotskip #1
2464 {
2465   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1lofskip } }
2466   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2467   {
2468     \addtocontents { lof }
2469     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2470   }
2471   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1lotskip } }
2472   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }

```



```

2473     {
2474         \addtocontents { lot }
2475         { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2476     }
2477 }
2478 </book|report>

\CTEX@addtocline 2479 \cs_new_protected:Npn \CTEX@addtocline #1#2
2480 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

2481 <@@=>

\part 2482 <*article>
2483 \renewcommand\part{%
2484     \if@noskipsec \leavevmode \fi
2485     \par
2486     \CTEX@part@break
2487     % \addvspace{4ex}%
2488     \CTEX@setheadingskip \CTEX@part@beforeskip
2489     \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2490     \addvspace \CTEX@headingskip
2491     \ifodd \CTEX@part@afterindent
2492         \@afterindenttrue
2493     \else
2494         \@afterindentfalse
2495     \fi
2496     \secdef\@part\@spart}
2497 </article>
2498 <*book|report>
2499 \renewcommand\part{%
2500 % \if@openright
2501 % \cleardoublepage
2502 % \else
2503 % \clearpage
2504 % \fi
2505 \CTEX@part@break
2506 % \thispagestyle{plain}%
2507 \thispagestyle{\CTEX@part@pagestyle}%
2508 \if@twocolumn
2509     \onecolumn
2510     \@tempwatrue
2511 \else
2512     \@tempwafalse
2513 \fi
2514 % \null\vfil
2515 \CTEX@setheadingskip \CTEX@part@beforeskip
2516 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2517 \vspace*{\CTEX@headingskip}%
2518 \secdef\@part\@spart}
2519 </book|report>

\@part 2520 <*article>
2521 \def\@part[#1]#2{%
2522     \ifnum \c@secnumdepth >\m@ne
2523         \ifodd \CTEX@part@numbering
2524             \CTEX@ifnametrue
2525             \refstepcounter{part}%
2526             % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2527         \else
2528             \CTEX@ifnamefalse
2529             \CTEX@makeanchor{part*}%
2530             % \addcontentsline{toc}{part}{#1}%
2531         \fi
2532     \else

```

```

2533 \CTEX@ifnamefalse
2534 \CTEX@makeanchor{part*}%
2535 % \addcontentsline{toc}{part}{#1}%
2536 \fi
2537 \CTEX@addtocline{part}{#1}%
2538 {\interlinepenalty \@M
2539 % \normalfont \parindent \z@ \raggedright
2540 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2541 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2542 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2543 \CTEX@partname \CTEX@part@aftername
2544 \fi \fi
2545 % \huge\bfseries #2%
2546 \CTEX@part@titleformat{#2}%
2547 % \markboth{}{}%
2548 \partmark{#1}%
2549 \CTEX@part@aftertitle}%
2550 \nobreak
2551 % \vskip 3ex
2552 \CTEX@setheadingskip \CTEX@part@afterskip
2553 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2554 \vskip \CTEX@headingskip
2555 \@afterheading}
2556 </article>
2557 <*book|report>
2558 \def\@part[#1]#2{%
2559 \ifnum \c@secnumdepth >-2\relax
2560 \ifodd \CTEX@part@numbering
2561 \CTEX@ifnametrue
2562 \refstepcounter{part}%
2563 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2564 \else
2565 \CTEX@ifnamefalse
2566 \CTEX@makeanchor{part*}%
2567 % \addcontentsline{toc}{part}{#1}%
2568 \fi
2569 \else
2570 \CTEX@ifnamefalse
2571 \CTEX@makeanchor{part*}%
2572 % \addcontentsline{toc}{part}{#1}%
2573 \fi
2574 \CTEX@addtocline{part}{#1}%
2575 % \markboth{}{}%
2576 \partmark{#1}%
2577 {\interlinepenalty \@M
2578 % \normalfont \centering
2579 \normalfont \CTEX@part@format
2580 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2581 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2582 \CTEX@partname \CTEX@part@aftername
2583 \fi \fi
2584 % \Huge\bfseries #2\par}%
2585 \CTEX@part@titleformat{#2}%
2586 \CTEX@part@aftertitle}%
2587 \@endpart}
2588 </book|report>

\@spart 2589 <*article>
2590 \def\@spart#1{%
2591 \CTEX@ifnamefalse
2592 \CTEX@makeanchor@spart{part*}%
2593 {\interlinepenalty \@M
2594 % \normalfont \parindent \z@ \raggedright
2595 \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2596 \CTEX@part@format
2597 % \huge \bfseries #1\par}%
2598 \CTEX@part@titleformat{#1}%

```

```

2599 \CTEX@part@aftertitle}%
2600 \nobreak
2601 % \vskip 3ex
2602 \CTEX@setheadingskip \CTEX@part@afterskip
2603 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2604 \vskip \CTEX@headingskip
2605 \@afterheading}
2606 </article>
2607 <*book|report>
2608 \def\@spart#1{%
2609 \CTEX@ifnamefalse
2610 \CTEX@makeanchor@spart{part*}%
2611 {\interlinepenalty \@M
2612 % \normalfont \centering
2613 \normalfont \CTEX@part@format
2614 % \Huge \bfseries #1\par}%
2615 \CTEX@part@titleformat{#1}%
2616 \CTEX@part@aftertitle}%
2617 \@endpart}
2618 </book|report>

\@endpart 2619 <*book|report>
2620 \def\@endpart{%
2621 % \vfil
2622 \CTEX@setheadingskip \CTEX@part@afterskip
2623 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2624 \vskip \CTEX@headingskip
2625 \newpage
2626 \if@twoside
2627 \if@openright
2628 \null
2629 \thispagestyle{empty}%
2630 \newpage
2631 \fi
2632 \fi
2633 \if@tempwa
2634 \twocolumn
2635 \fi}
2636 </book|report>

14.8.2.2 chapter 的标题
2637 <*book|report>

\chapter 2638 \renewcommand\chapter{%
2639 % \if@openright\cleardoublepage\else\clearpage\fi
2640 % \thispagestyle{plain}%
2641 \CTEX@chapter@break
2642 \thispagestyle{\CTEX@chapter@pagestyle}%
2643 \global\@topnum\z@
2644 % \@afterindentfalse
2645 \ifodd \CTEX@chapter@afterindent
2646 \afterindenttrue
2647 \else
2648 \afterindentfalse
2649 \fi
2650 \secdef\@chapter\@schapter}

\@chapter 2651 \def\@chapter[#1]#2{%
2652 \ifnum \c@secnumdepth >\m@ne
2653 <*book>
2654 \if@mainmatter
2655 </book>
2656 \ifodd \CTEX@chapter@numbering
2657 \CTEX@ifnametrue
2658 \refstepcounter{chapter}%
2659 % \typeout{\@chapapp\space\thechapter.}%
2660 \typeout{\CTEXthechapter}%

```

```

2661 % \addcontentsline{toc}{chapter}
2662 % {\protect\numberline{\thechapter}{#1}}%
2663 \else
2664 \CTEX@ifnamefalse
2665 \CTEX@makeanchor{\Hy@chapapp*}%
2666 % \addcontentsline{toc}{chapter}{#1}%
2667 \fi
2668 <*book>
2669 \else
2670 \CTEX@ifnamefalse
2671 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2672 % \addcontentsline{toc}{chapter}{#1}%
2673 \fi
2674 </book>
2675 \else
2676 \CTEX@ifnamefalse
2677 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2678 % \addcontentsline{toc}{chapter}{#1}%
2679 \fi
2680 \CTEX@addtoctline{chapter}{#1}%
2681 \chaptermark{#1}%
2682 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2683 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2684 \CTEX@addloflotskip{chapter}%
2685 \if@twocolumn
2686 \topnewpage[\@makechapterhead{#2}]%
2687 \else
2688 \makechapterhead{#2}%
2689 \@afterheading
2690 \fi}

\@schapter 2691 \def\@schapter#1{%
2692 \CTEX@ifnamefalse
2693 \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2694 \if@twocolumn
2695 \topnewpage[\@makeschapterhead{#1}]%
2696 \else
2697 \makeschapterhead{#1}%
2698 \@afterheading
2699 \fi}

\@makechapterhead 2700 \def\@makechapterhead#1{%
2701 % \vspace*{50\p@}%
2702 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2703 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2704 \vspace*{\CTEX@headingskip}%
2705 % {\normalfont \parindent \z@ \raggedright
2706 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2707 \CTEX@chapter@format
2708 % \ifnum \c@secnumdepth >\m@ne
2709 % \if@mainmatter
2710 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2711 % \fi
2712 % \fi
2713 \CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}%
2714 \interlinepenalty\@M
2715 % \Huge \bfseries #1\par\nobreak
2716 \CTEX@chapter@titleformat{#1}%
2717 \CTEX@chapter@aftertitle
2718 \nobreak
2719 % \vskip 40\p@
2720 \CTEX@setheadingskip \CTEX@chapter@afterskip
2721 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2722 \vskip \CTEX@headingskip
2723 }}

\@makeschapterhead 2724 \def\@makeschapterhead#1{%

```

```

2725 % \vspace*{50\p@}%
2726 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2727 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2728 \vspace*{\CTEX@headingskip}%
2729 % {\normalfont \parindent \z@ \raggedright
2730 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2731 \CTEX@chapter@format
2732 \interlinepenalty\@M
2733 % \Huge \bfseries #1\par\nobreak
2734 \CTEX@chapter@titleformat{#1}
2735 \CTEX@chapter@aftertitle
2736 \nobreak
2737 % \vskip 40\p@
2738 \CTEX@setheadingskip \CTEX@chapter@afterskip
2739 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2740 \vskip \CTEX@headingskip
2741 }}
2742 </book|report>

```

14.8.2.3 section 类的标题

```

\@startsection 2743 \def\@startsection#1#2#3#4#5#6{%
2744 \if@noskipsec \leavevmode \fi
2745 \par
2746 % \@tempskipa #4\relax
2747 % \@afterindenttrue
2748 % \ifdim \@tempskipa <\z@
2749 % \@tempskipa -\@tempskipa \@afterindentfalse
2750 % \fi
2751 \CTEX@update@sectionformat@n{#1}%
2752 \ifodd \CTEX@afterindent
2753 \@afterindenttrue
2754 \else
2755 \@afterindentfalse
2756 \fi
2757 \if@nobreak
2758 \everypar{}%
2759 \else
2760 % \addpenalty\@secpenalty\addvspace\@tempskipa
2761 \csname CTEX@#1@break\endcsname
2762 \CTEX@setheadingskip{#4}%
2763 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2764 \addvspace \CTEX@headingskip
2765 \fi
2766 \ifstar
2767 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2768 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%
\@secntformat 2769 \def\@secntformat#1{%
2770 % \csname the#1\endcsname\quad}%
2771 \csname CTEX@#1name\endcsname
2772 \csname CTEX@#1@aftername\endcsname}
\@sect 2773 \def\@sect#1#2#3#4#5#6[#7]#8{%
2774 \ifnum #2>\c@secnumdepth
2775 \CTEX@ifnamefalse
2776 \CTEX@makeanchor@sect{#1*}%
2777 \let\@svsec\@empty
2778 \else
2779 \ifodd \csname CTEX@#1@numbering\endcsname
2780 \CTEX@ifnametrue
2781 \refstepcounter{#1}%
2782 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2783 \else
2784 \CTEX@ifnamefalse
2785 \CTEX@makeanchor{#1*}%
2786 \let\@svsec\@empty

```

```

2787     \fi
2788     \fi
2789 % \@tempskipa #5\relax
2790 % \ifdim \@tempskipa>\z@
2791 \unless \ifodd \CTEX@runin
2792     \begingroup
2793     #6{%
2794         \CTEX@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2795 %         \interlinepenalty \@M #8\@par}%
2796         \interlinepenalty \@M
2797         \csname CTEX@#1@titleformat\endcsname{#8}%
2798         \csname CTEX@#1@aftertitle\endcsname}%
2799     \endgroup
2800     \csname #1mark\endcsname{#7}%
2801 %     \addcontentsline{toc}{#1}{%
2802 %         \ifnum #2>\c@secnumdepth \else
2803 %             \protect\numberline{\csname the#1\endcsname}%
2804 %         \fi
2805 %         #7}%
2806     \CTEX@addtoctline{#1}{#7}%
2807 \else
2808     \def\@svsechd{%
2809     #6{\hskip\glueexpr #3\relax
2810 %     \@svsec #8}%
2811     \@svsec
2812     \csname CTEX@#1@titleformat\endcsname{#8}%
2813     \csname CTEX@#1@aftertitle\endcsname}%
2814     \csname #1mark\endcsname{#7}%
2815 %     \addcontentsline{toc}{#1}{%
2816 %         \ifnum #2>\c@secnumdepth \else
2817 %             \protect\numberline{\csname the#1\endcsname}%
2818 %         \fi
2819 %         #7}%
2820     \CTEX@addtoctline{#1}{#7}%
2821     \fi
2822     \@xsect{#5}}

\@sssect 2823 \def\@sssect#1#2#3#4#5{%
2824     \CTEX@ifnamefalse
2825 % \@tempskipa #3\relax
2826 % \ifdim \@tempskipa>\z@
2827 \unless \ifodd \CTEX@runin
2828     \begingroup
2829     #4{%
2830         \CTEX@hangfrom{\hskip\glueexpr #1\relax}%
2831 %         \interlinepenalty \@M #5\@par}%
2832         \interlinepenalty \@M
2833         \CTEX@titleformat@n{#5}%
2834         \CTEX@aftertitle}%
2835     \endgroup
2836 \else
2837 %     \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2838     \def\@svsechd{#4{\hskip\glueexpr #1\relax
2839         \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2840     \fi
2841     \@xsect{#3}}

\@xsect 2842 \def\@xsect#1{%
2843 % \@tempskipa #1\relax
2844 % \ifdim \@tempskipa>\z@
2845 \unless \ifodd \CTEX@runin
2846     \par \nobreak
2847 %     \vskip \@tempskipa
2848     \CTEX@setheadingskip{#1}%
2849     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2850     \vskip \CTEX@headingskip
2851     \@afterheading

```

```

2852 \else
2853   \@nobreakfalse
2854   \global\@noskipsectrue
2855   \everypar{%
2856     \if@noskipsec
2857       \global\@noskipsecfalse
2858       {\setbox\z@\lastbox}%
2859       \clubpenalty\@M
2860       \beginingroup \@svsechd \endgroup
2861       \unskip
2862       \@tempskipa #1\relax
2863       \hskip -\@tempskipa
2864       \hskip\glueexpr #1\relax
2865     \else
2866       \clubpenalty \@clubpenalty
2867       \everypar{}%
2868     \fi}%
2869 \fi
2870 \ignorespaces}

2871 <@@=-ctex>

```

\CTEX@hangfrom hang 选项控制是否采用悬挂缩进。

```

2872 \cs_new_protected_nopar:Npn \CTEX@hangfrom
2873 {
2874   \bool_if:NTF \CTEX@hang
2875     { \hangfrom }
2876     { \noindent \use:n }
2877 }

```

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

```

2878 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2879 {
2880   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2881   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2882   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2883   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2884   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2885   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2886 }
2887 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2888 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2889 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2890 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2891 \cs_new_eq:NN \CTEX@hang \c_true_bool
2892 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

```

\CTEX@part@tocline 2893 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 2894 {
2895   \CTEXifname
2896   { \CTEXthepart \hspace { 1em } }
2897   { }
2898   #2
2899 }
2900 <*book|report>
2901 \cs_new:Npn \CTEX@chapter@tocline #1#2
2902 {
2903   \CTEXifname
2904   { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2905   { }
2906   #2
2907 }
2908 </book|report>

```

\CTEXnumberline 2909 \cs_new_nopar:Npn \CTEXnumberline #1

```

2910 {
2911   \CTEXifname
2912   { \protect \numberline { \use:c { CTEXthe #1 } } }
2913   { }
2914 }

2915 \int_zero:N \l__ctex_tmp_int
2916 \seq_map_inline:Nn \c__ctex_section_headings_seq
2917 {
2918   \int_incr:N \l__ctex_tmp_int
2919   \cs_gset_protected_nopar:cpx {#1}
2920   {
2921     \exp_not:N \@startsection {#1}
2922     { \int_use:N \l__ctex_tmp_int }
2923     { \exp_not:c { CTEX@#1@indent } }
2924     { \exp_not:c { CTEX@#1@beforeskip } }
2925     { \exp_not:c { CTEX@#1@afterskip } }
2926     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2927   }
2928   \cs_new:cpn { CTEX@#1@tocline } ##1##2
2929   { \CTEXnumberline { ##1 } ##2 }
2930 }

```

14.8.2.4 附录标题

```

appendix/name 2931 \keys_define:nn { ctex }
appendix/number 2932 {
appendix/numbering 2933   appendix .meta:nn = { ctex / appendix } {#1} ,
2934   appendix / name .code:n =
2935   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2936   appendix / number .tl_set:N = \CTEX@appendix@number ,
2937   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2938   appendix / numbering .initial:n = true
2939 }
2940 \tl_new:N \CTEX@preappendix
2941 \tl_new:N \CTEX@postappendix

\appendix 2942 \cs_new_eq:NN \CTEX@save@appendix \appendix
2943 \cs_gset_protected_nopar:Npn \appendix
2944 {
2945   \CTEX@save@appendix
2946   <*article>
2947   \gdef \CTEX@presection { \CTEX@preappendix }
2948   \gdef \CTEX@thesection { \CTEX@appendix@number }
2949   \gdef \CTEX@postsection { \CTEX@postappendix }
2950   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2951   </article>
2952   <*book|report>
2953   \gdef \CTEX@prechapter { \CTEX@preappendix }
2954   \gdef \CTEX@thechapter { \CTEX@appendix@number }
2955   \gdef \CTEX@postchapter { \CTEX@postappendix }
2956   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2957   </book|report>
2958 }

```

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。

```

2959 \cs_new_protected_nopar:Npn \CTEX@makeanchor #1
2960 { }

```

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

```

2961 \seq_new:N \c__ctex_headings_cs_seq
2962 \seq_gset_from_clist:Nn \c__ctex_headings_cs_seq

```



```

2963 { part , spart , sect , ssect }
2964 { part , spart , chapter , schapter , sect , ssect }
2965 \seq_map_inline:Nn \c__ctex_headings_cs_seq
2966 {
2967   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
2968   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
2969 }

```

\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。

```

2970 \cs_new_protected_nopar:Npn \CTEX@hyperheadinghook
2971 {
2972   \group_begin:
2973     \ifHy@implicit
2974       \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
2975       \seq_map_inline:Nn \c__ctex_headings_cs_seq
2976       {
2977         \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
2978         {
2979           \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
2980           \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
2981         }
2982       }
2983     \else:
2984       \seq_map_inline:Nn \c__ctex_headings_cs_seq
2985       { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
2986     \fi:
2987   \group_end:
2988 }

2989 \ctex_at_end_package:nn { hyperref }
2990 {
2991   \cs_gset_protected_nopar:Npn \CTEX@makeanchor #1
2992   {
2993     \Hy@MakeCurrentHrefAuto {#1}
2994     \Hy@raisedlink
2995     {
2996       \hyper@anchorstart { \@currentHref }
2997       \hyper@anchorend
2998     }
2999   }
3000   \CTEX@hyperheadinghook
3001 }

```

14.8.2.6 兼容 titlesec 宏包

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

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

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

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

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

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

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

```

3002 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
3003 {
3004   \@ifpackagewith { titlesec } { explicit }
3005   {
3006     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3007     \__ctex_titlesec_format_explicit:Nn
3008   }
3009   { }
3010   \clist_map_inline:nn
3011   { indentafter , noindentafter , indentfirst , nonindentfirst }
3012   {
3013     \@ifpackagewith { titlesec } { ##1 }
3014     {
3015       \clist_map_break:n
3016       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3017     }
3018     { }
3019   }
3020   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3021 }
3022 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
3023 {
3024   \__ctex_titlesec_format:n {#1}
3025   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3026 }
3027 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
3028 {
3029   \cs_if_free:cF { ttlf@#1 }
3030   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3031 }
3032 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
3033 {
3034   \tl_set:Nx #1
3035   {
3036     \bool_if:cTF { CTEX@#2@runin }
3037     { \exp_not:N \ttlh@runin }
3038     { \exp_not:N \ttlh@hang }
3039     \tl_tail:N #1
3040   }
3041 }
3042 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3043 {
3044   \cs_set_nopar:Npx #1 ##1
3045   {
3046     \bool_if:cTF { CTEX@#2@runin }
3047     { \exp_not:N \ttlh@runin }
3048     { \exp_not:N \ttlh@hang }
3049     \exp_args:No \tl_tail:n { #1 { } }
3050   }
3051 }
3052 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
3053 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3054 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3055 {
3056   \exp_not:n { {#1} {#2} {#3} {#4} }
3057   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3058 }

3059 \@ifpackageloaded { titlesec }
3060 { }

```

```

3061 {
3062   \ctex_at_end_package:nn { titlesec }
3063   {
3064     \@ifpackagewith { titlesec } { loadonly }
3065     { }
3066     { \ctex_titlesec_hook: }
3067   }
3068 }

```

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

```

3069 \ctex_at_end_package:nn { titlesec }
3070 {
3071   <*book|report>
3072   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3073   </book|report>
3074   \cs_if_free:NF \ttl@tocpart
3075   {
3076     \cs_set_protected_nopar:Npn \ttl@tocpart
3077     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3078   }
3079   \seq_map_inline:Nn \c__ctex_headings_seq
3080   {
3081     \cs_if_exist:cF { ttl@toc #1 }
3082     {
3083       \cs_new_protected_nopar:cpx { ttl@toc #1 }
3084       {
3085         \tl_set:Nn \exp_not:N \ttl@a
3086         {
3087           \exp_not:N \protect
3088           \exp_not:N \numberline { \exp_not:c { CTExthe #1 } }
3089         }
3090       }
3091     }
3092   }
3093 }

```

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。

```

3094 \group_begin:
3095 \char_set_catcode_other:N \#
3096 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
3097 {
3098   \ctex_patch_cmd:Nnn \ttl@settopmark
3099   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3100   {
3101     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3102     \CTEX@titlepslabel@set {#1}
3103   }
3104   \ctex_patch_cmd:Nnn \ttl@setsubmark
3105   { \protect \@namedef { the#1 } { } }
3106   {
3107     \protect \@namedef { the#1 } { }
3108     \CTEX@titlepslabel@clear {#1}
3109   }
3110   \ctex_patch_cmd:Nnn \ttl@setsubmark
3111   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3112   {

```

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

```

3113         \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3114         \CTEX@titlepslabel@set {#2}
3115     }
3116 }
3117 \group_end:

```

```

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

```

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

```

3118 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
3119 {
3120     \cs_if_free:cF { CTEXthe#1 }
3121     { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3122 }
3123 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
3124 {
3125     \cs_if_free:cF { CTEXthe#1 }
3126     { \protect \@namedef { CTEXthe#1 } { } }
3127 }

```

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

```

3128 \ctex_at_end_package:nn { titlesec }
3129 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3130 \ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }

```

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

```

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

```

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

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

```

3131 \ctex_at_end_package:nn { titleps }
3132 {
3133     \cs_set_protected_nopar:Npn \ttl@setifthe #1
3134     {
3135         \exp_args:Nco \cs_set_nopar:Npn { ifthe #1 }
3136         {
3137             \CTEXifname
3138             { \protect \@firstoftwo }
3139             { \protect \@secondoftwo }
3140         }
3141     }
3142     \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3143 }

```

14.8.3 目录标签的宽度

```

\numberline 3144 \cs_new_protected:Npn \CTEX@toc@width@n #1
3145 {
3146     \hbox_set:Nn \l__ctex_tmp_box {#1}
3147     \dim_set:Nn \@tempdima
3148     {
3149         \dim_max:nn { \@tempdima }
3150         { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
3151     }
3152 }
3153 \group_begin:
3154 \char_set_catcode_other:N \#
3155 \use:n
3156 {
3157     \group_end:
3158     \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }

```

```

3159     { \CTEX@toc@width@n {#1} }
3160     { }
3161     { \ctex_patch_failure:N \numberline }
3162     \@ifpackageloaded { tocloft }
3163     { }
3164     {
3165         \ctex_at_end_package:nn { tocloft }
3166         {
3167             \ctex_preto_cmd:NnnTF \numberline
3168             { \char_set_catcode_letter:n { 64 } }
3169             { \CTEX@toc@width@n {#1} }
3170             { }
3171             { \ctex_patch_failure:N \numberline }
3172         }
3173     }
3174 }

```

14.8.4 页眉信息的修改

```

\ps@headings 3175 <*article>
3176 \if@twoside
3177     \ctex_patch_cmd:Nnn \ps@headings
3178     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3179     { \CTEXifname { \CTEXthesection \quad } { } }
3180     \ctex_patch_cmd:Nnn \ps@headings
3181     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3182     { \CTEXifname { \CTEXthesubsection \quad } { } }
3183 \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。

```

3184     \ctex_patch_cmd:Nnn \ps@headings
3185     { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3186     { \CTEXifname { \CTEXthesection \quad } { } }
3187 \fi:
3188 </article>
3189 <*book|report>
3190 \ctex_patch_cmd:Nnn \ps@headings
3191 {
3192     \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3193     \ifnum \c@secnumdepth > \m@ne
3194         \@chapapp \ thechapter . ~ \ %
3195     \fi
3196     \ifnum \c@secnumdepth > \m@ne
3197     \fi \fi
3198     { \CTEXifname { \CTEXthechapter \quad } { } }
3199 \if@twoside
3200     \ctex_patch_cmd:Nnn \ps@headings
3201     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3202     { \CTEXifname { \CTEXthesection } { } }
3203 \fi:
3204 </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。

```

3205 \ctex_at_end_package:nn { fancyhdr }
3206 {
3207 <*article>
3208     \ctex_patch_cmd:Nnn \ps@fancy
3209     { \ifnum \c@secnumdepth > \z@ \thesection \hspace{1em} \relax \fi }
3210     { \CTEXifname { \CTEXthesection \quad } { } }
3211     \ctex_patch_cmd:Nnn \ps@fancy
3212     { \ifnum \c@secnumdepth > \@ne \thesubsection \hspace{1em} \relax \fi }
3213     { \CTEXifname { \CTEXthesubsection \quad } { } }
3214 </article>
3215 <*book|report>

```

```

3216 \ctex_patch_cmd:Nnn \ps@fancy
3217 { \ifnum \c@secnumdepth > \m@ne \chapapp \ \thechapter . ~ \ \fi }
3218 { \CTEXifname { \CTEXthechapter \quad } { } }
3219 \ctex_patch_cmd:Nnn \ps@fancy
3220 { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3221 { \CTEXifname { \CTEXthesection } { } }
3222 </book|report>
3223 }
3224 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

```

3225 <*beamer>
3226 \ExplSyntaxOff

```

对应 \partpage。

```

3227 \defbeamertemplate*{part page}{CTEX}[1][]{
3228 \begingroup
3229 % \centering
3230 % {\usebeamerfont{part name}\usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3231 % \vskip1em\par
3232 \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3233 \parindent \dimexpr \CTEX@part@indent \relax
3234 \CTEX@part@format
3235 \ifodd \CTEX@part@numbering
3236 \CTEX@partname \CTEX@part@aftername
3237 \fi
3238 \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3239 % \usebeamerfont{part title}\insertpart\par
3240 \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3241 \end{beamercolorbox}
3242 \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3243 \endgroup
3244 }

```

对应 \sectionpage。

```

3245 \defbeamertemplate*{section page}{CTEX}[1][]{
3246 \begingroup
3247 % \centering
3248 % {\usebeamerfont{section name}\usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}
3249 % \vskip1em\par
3250 \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3251 \parindent \dimexpr \CTEX@section@indent \relax
3252 \CTEX@section@format
3253 \ifodd \CTEX@section@numbering
3254 \CTEX@sectionname \CTEX@section@aftername
3255 \fi
3256 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3257 % \usebeamerfont{section title}\insertsection\par
3258 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3259 \end{beamercolorbox}
3260 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3261 \endgroup
3262 }

```

对应 \subsectionpage。

```

3263 \defbeamertemplate*{subsection page}{CTEX}[1][]{
3264 \begingroup
3265 % \centering
3266 % {\usebeamerfont{subsection name}\usebeamercolor[fg]{subsection name}\subsectionname~\insertsectionnumber}
3267 % \vskip1em\par
3268 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3269 \parindent \dimexpr \CTEX@subsection@indent \relax
3270 \CTEX@subsection@format

```

```

3271 \ifodd \CTEX@subsection@numbering
3272 \CTEX@subsectionname \CTEX@subsection@aftername
3273 \fi
3274 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3275 % \usebeamerfont{subsection title}\insertsubsection\par
3276 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3277 \end{beamercolorbox}
3278 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3279 \endgroup
3280 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3281 \defbeamerplatealias{part page}{default}{CTEX}
3282 \defbeamerplatealias{section page}{default}{CTEX}
3283 \defbeamerplatealias{subsection page}{default}{CTEX}
3284 \ExplSyntaxOn
3285 </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。

```

3286 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
3287 {
3288   \protected@edef \@currentlabel
3289   {
3290     \cs_if_exist:cTF { CTEX@the#1 }
3291     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3292     { \exp_not:o { \@currentlabel } }
3293   }
3294 }

```

`\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。

```

3295 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3296 {
3297   \seq_map_inline:Nn \c__ctex_headings_seq
3298   { \ctex_fix_varioref_label:n { ##1 } }
3299 }

```

`__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。

```

3300 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3301 {
3302   \tl_if_empty:cT { p@#1 }
3303   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3304 }

```

如果 `varioref` 已经被载入,则使用它来设置。

```

3305 \ifpackageloaded { varioref }
3306 { \ctex_varioref_hook: }
3307 {
3308   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3309   \RenewDocumentCommand \refstepcounter { m }

```

```

3310     {
3311         \CTEX@save@refstepcounter {#1}
3312         \CTEX@setcurrentlabel@n {#1}
3313     }
3314     \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3315 }

```

14.8.7 载入 $\langle scheme \rangle$ 文件

```

3316 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3317 </class|heading>

```

14.8.8 标题格式的 scheme 定义

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

```

3318 <*scheme&(article|book|report|beamer)>

3319 \keys_set:nn { ctex / part }
3320 {
3321     aftertitle = \par ,
3322     <*plain>
3323     name       = \partname \space ,
3324     <*article|book|report>
3325     number     = \thepart ,
3326     </article|book|report>
3327     <*beamer>
3328     number     = \insertromanpartnumber ,
3329     </beamer>
3330     </plain>
3331     <*chinese>
3332     number     = \chinese { part } ,
3333     </chinese>
3334     <*article>
3335     beforeskip = 4ex ,
3336     afterskip  = 3ex ,
3337     <*plain>
3338     format     = \raggedright ,
3339     nameformat = \Large \bfseries ,
3340     aftername   = \par \nobreak ,
3341     titleformat = \huge \bfseries ,
3342     afterindent = false
3343     </plain>
3344     <*chinese>
3345     format     = \Large \bfseries \centering ,
3346     aftername   = \quad ,
3347     afterindent = true
3348     </chinese>
3349     </article>
3350     <*book|report>
3351     aftername   = \par \vskip 20 \p@ ,
3352     beforeskip = 0pt \@plus 1fil ,
3353     afterskip  = 0pt \@plus 1fil ,
3354     pagestyle   = plain ,
3355     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3356     <*plain>
3357     format     = \centering ,
3358     nameformat = \huge \bfseries ,
3359     titleformat = \Huge \bfseries
3360     </plain>
3361     <*chinese>
3362     format     = \huge \bfseries \centering
3363     </chinese>
3364     </book|report>
3365     <*beamer>

```



```

3366     format      = \centering ,
3367     nameformat   = \usebeamerfont { part ~ name }
3368                 \usebeamercolor [fg] { part ~ name } ,
3369     aftername     = \vskip 1em \par ,
3370     titleformat   = \usebeamerfont { part ~ title }
3371 </beamer>
3372 }

3373 <*book|report>
3374 \keys_set:nn { ctex / chapter }
3375 {
3376     pagestyle     = plain ,
3377     aftertitle    = \par ,
3378     beforekip     = 50 \p@ ,
3379     afterskip     = 40 \p@ ,
3380     lofskip       = 10 \p@ ,
3381     lotskip       = 10 \p@ ,
3382     break         = \if@openright \cleardoublepage \else \clearpage \fi ,
3383 <*plain>
3384     name          = \chaptername \space ,
3385     number        = \thechapter ,
3386     format        = \raggedright ,
3387     nameformat    = \huge \bfseries ,
3388     aftername     = \par \nobreak \vskip 20 \p@ ,
3389     titleformat   = \Huge \bfseries ,
3390     afterindent   = false ,
3391     tocline       = \CTEXnumberline {#1} #2
3392 </plain>
3393 <*chinese>
3394     number        = \chinese { chapter } ,
3395     format        = \huge \bfseries \centering ,
3396     aftername     = \quad ,
3397     afterindent   = true
3398 </chinese>
3399 }
3400 </book|report>

3401 <@@= >
3402 \keys_set:nn { ctex / section }
3403 {
3404 <*article|book|report>
3405     number        = \thesection ,
3406     aftername     = \quad ,
3407     aftertitle    = \@@par ,
3408     beforekip     = 3.5ex \@plus 1ex \@minus .2ex ,
3409     afterskip     = 2.3ex \@plus .2ex ,
3410     runin         = false ,
3411     break         = \addpenalty \@secpenalty ,
3412 <*plain>
3413     format        = \Large \bfseries ,
3414     afterindent   = false
3415 </plain>
3416 <*chinese>
3417     format        = \Large \bfseries \centering ,
3418     afterindent   = true
3419 </chinese>
3420 </article|book|report>
3421 <*beamer>
3422 <*plain>
3423     name          = \sectionname \space ,
3424 </plain>
3425     format        = \centering ,
3426     number        = \insertsectionnumber ,
3427     nameformat    = \usebeamerfont { section ~ name }
3428                 \usebeamercolor [fg] { section ~ name } ,
3429     aftername     = \vskip 1em \par ,
3430     titleformat   = \usebeamerfont { section ~ title } ,

```

```

3431     aftertitle = \par
3432 </beamer>
3433 }

3434 \keys_set:nn { ctex / subsection }
3435 {
3436 <*article|book|report>
3437     number      = \thesubsection ,
3438     format      = \large \bfseries ,
3439     aftername    = \quad ,
3440     aftertitle   = \@@par ,
3441     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
3442     afterskip    = 1.5ex \@plus .2ex ,
3443     runin        = false ,
3444     break        = \addpenalty \@secpenalty ,
3445 <*plain>
3446     afterindent = false
3447 </plain>
3448 <*chinese>
3449     afterindent = true
3450 </chinese>
3451 </article|book|report>
3452 <*beamer>
3453 <*plain>
3454     name         = \subsectionname \space ,
3455     number       = \insertsubsectionnumber ,
3456 </plain>
3457 <*chinese>
3458     number       = \arabic { section } . \arabic { subsection } ,
3459 </chinese>
3460     format       = \centering ,
3461     nameformat    = \usebeamerfont { subsection ~ name }
3462                   \usebeamerfont { subsection ~ name } ,
3463     aftername     = \vskip 1em \par ,
3464     titleformat   = \usebeamerfont { subsection ~ title } ,
3465     aftertitle    = \par
3466 </beamer>
3467 }

3468 <*article|book|report>

3469 \keys_set:nn { ctex / subsubsection }
3470 {
3471     number      = \thesubsubsection ,
3472     format      = \normalsize \bfseries ,
3473     aftername    = \quad ,
3474     aftertitle   = \@@par ,
3475     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
3476     afterskip    = 1.5ex \@plus .2ex ,
3477     runin        = false ,
3478     break        = \addpenalty \@secpenalty ,
3479 <*plain>
3480     afterindent = false
3481 </plain>
3482 <*chinese>
3483     afterindent = true
3484 </chinese>
3485 }

3486 \keys_set:nn { ctex / paragraph }
3487 {
3488     number      = \theparagraph ,
3489     format      = \normalsize \bfseries ,
3490     aftername    = \quad ,
3491     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
3492     break        = \addpenalty \@secpenalty ,
3493 <*plain>
3494     afterindent = false
3495 </plain>

```

```

3496 <*chinese>
3497     afterindent = true
3498 </chinese>
3499 }

3500 \keys_set:nn { ctex / subparagraph }
3501 {
3502     number      = \thesubparagraph ,
3503     format      = \normalsize \bfseries ,
3504     aftername    = \quad ,
3505     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
3506     break       = \addpenalty \@secpenalty ,
3507 <*plain>
3508     afterindent = false
3509 </plain>
3510 <*chinese>
3511     afterindent = true
3512 </chinese>
3513 }

```

处理 sub3section 与 sub4section 的格式。

```

3514 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3515 {
3516     \keys_set:nn { ctex / paragraph }
3517     {
3518         aftertitle = \@@par ,
3519         afterskip  = 1ex \@plus .2ex ,
3520         runin      = false
3521     }
3522 }
3523 {
3524     \keys_set:nn { ctex / paragraph }
3525     {
3526         afterskip  = 1em ,
3527         runin      = true
3528     }
3529 }
3530 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3531 {
3532     \keys_set:nn { ctex / subparagraph }
3533     {
3534         aftertitle = \@@par ,
3535         afterskip  = 1ex \@plus .2ex ,
3536         runin      = false
3537     }
3538 }
3539 {
3540     \keys_set:nn { ctex / subparagraph }
3541     {
3542         afterskip  = 1em ,
3543         runin      = true
3544     }
3545 }
3546 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3547 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3548 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3549 <@@=ctex>

```

处理附录的格式。

```

3550 \keys_set:nn { ctex / appendix }
3551 <*article>
3552 { number      = \@Alph \c@section }
3553 </article>
3554 <*book|report>
3555 {
3556     name       = \appendixname \space ,

```

```

3557     number      = \@Alph \c@chapter
3558   }
3559 </book|report>

3560 </article|book|report>

3561 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

```
3562 <*ctex|ctexheading>
```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3563 \clist_map_inline:nn { article , book , report , beamer }
3564 {
3565   \@ifclassloaded {#1}
3566   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3567   { }
3568 }

```

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。

```

3569 \msg_new:nnn { ctex } { not-standard-class }
3570 {
3571   None~of~the~standard~document~classes~was~loaded.\\
3572   Heading~`#1'~is~selected.\\
3573   ctex~may~not~work~as~expected.
3574 }
3575 <ctex> \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 3576 \use:n
3577 {
3578   \tl_if_exist:NTF \c__ctex_std_class_tl
3579   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3580   {
3581     \cs_if_exist:NTF \chapter
3582     {
3583       \cs_if_exist:NF \if@mainmatter
3584       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3585       \tl_const:Nn \c__ctex_class_tl { book }
3586     }
3587     { \tl_const:Nn \c__ctex_class_tl { article } }
3588     \msg_warning:nx { ctex } { not-standard-class } { \c__ctex_class_tl }
3589   }
3590   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3591 }
3592 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3593 </ctex|ctexheading>

```

14.8.10 标题配置文件

```

3594 <*name>
3595 \keys_set_known:nn { ctex }
3596 {
3597   contentsname = 目录 ,
3598   listfigurename = 插图 ,
3599   listtablename = 表格 ,
3600   figurename = 图 ,
3601   tablename = 表 ,
3602   abstractname = 摘要 ,
3603   indexname = 索引 ,
3604   bibname = 参考文献 ,
3605   appendixname = 附录 ,
3606   proofname = 证明 ,

```

```

3607     algorithmname = 算法 ,
3608     refname       = 参考文献 ,
3609     continuation  = (续) ,
3610     part / name = { 第 , 部分 } ,
3611     chapter / name = { 第 , 章 }
3612 }
3613 </name>

```

14.9 chinese 方案的其他设置

```
3614 <*scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
<article|book|report> 3615 \pagestyle { headings }
```

日期格式。

```
3616 \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。

```

3617 \ctex_if_autoindent_touched:F
3618 { \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。

```

3619 <*!generic>
3620 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3621 <*beamer>
3622 {
3623   \uselanguage { ChineseGBK }
3624   \languagealias { chinese } { ChineseGBK }
3625   \ctex_file_input:n { ctex-name-gbk.cfg }
3626 }
3627 {
3628   \uselanguage { ChineseUTF8 }
3629   \languagealias { chinese } { ChineseUTF8 }
3630   \ctex_file_input:n { ctex-name-utf8.cfg }
3631 }

```

让 translator 包优先查找中文翻译。

```

3632 \clist_put_left:Nn \trans@languagepath { chinese }
3633 </beamer>
3634 <*!beamer>
3635 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3636 { \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。

```

3637 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3638 { \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。

```

3639 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3640 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3641 </!beamer>
3642 </!generic>

```

不使用标题定义时的通用设置。

```

3643 <*generic>
3644 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3645 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3646 {
3647   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3648   {

```

```

3649     \uselanguage { ChineseGBK }
3650     \languagealias { chinese } { ChineseGBK }
3651     \ctex_file_input:n { ctex-name-gbk.cfg }
3652   }
3653   {
3654     \uselanguage { ChineseUTF8 }
3655     \languagealias { chinese } { ChineseUTF8 }
3656     \ctex_file_input:n { ctex-name-utf8.cfg }
3657   }
3658   \clist_put_left:Nn \trans@languagepath { chinese }
3659 }
3660 {
3661   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3662   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3663   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3664   \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3665   { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3666   \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3667   { \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。

```

3668   \tl_if_exist:NT \c__ctex_std_class_tl
3669   { \RequirePackage { indentfirst } }
3670 }
3671 </generic>
3672 </scheme&chinese>

```

14.10 中文字号

```

3673 <*class|ctex|ctexsize>
\zihao 3674 \NewDocumentCommand \zihao { m }
3675   { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }
\ctex_zihao:n 3676 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3677   {
3678     \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3679     { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3680     { \msg_error:nnn { ctex } { fontsize } {#1} }
3681   }
3682   \msg_new:nnnn { ctex } { fontsize }
3683   { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3684   {
3685     The~old~font~size~is~used~if~you~continue.\\
3686     The~available~font~sizes~are~listed~as~follow.\\
3687     \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3688   }

```

14.10.1 定义中文字号

`\c__ctex_font_size_prop` 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。

```

3689 \prop_new:N \c__ctex_font_size_prop
3690 \seq_new:N \c__ctex_font_size_seq
3691 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3692   {
3693     \use:x
3694     {
3695       \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3696       {
3697         { \dim_to_decimal:n {#2} }
3698         { \dim_to_decimal:n { (#2) * \c_six / \c_five } }

```

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

```

3699     }
3700   }
3701   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3702 }
3703 \clist_map_inline:nn
3704 {
3705   { 8 } { 5 bp } ,
3706   { 7 } { 5.5 bp } ,
3707   { -6 } { 6.5 bp } ,
3708   { 6 } { 7.5 bp } ,
3709   { -5 } { 9 bp } ,
3710   { 5 } { 10.5 bp } ,
3711   { -4 } { 12 bp } ,
3712   { 4 } { 14 bp } ,
3713   { -3 } { 15 bp } ,
3714   { 3 } { 16 bp } ,
3715   { -2 } { 18 bp } ,
3716   { 2 } { 22 bp } ,
3717   { -1 } { 24 bp } ,
3718   { 1 } { 26 bp } ,
3719   { -0 } { 36 bp } ,
3720   { 0 } { 42 bp }
3721 }
3722 { \__ctex_save_font_size:nn #1 }

\ctex_declare_math_sizes:nnnn 3723 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3724 {
3725   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3726   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3727 }

\__ctex_get_font_sizes:Nn 3728 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3729 {
3730   \tl_clear:N #1
3731   \tl_map_inline:nn {#2}
3732   {
3733     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3734     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3735     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3736   }
3737 }

3738 \clist_map_inline:nn
3739 {
3740   { 8 } { 8 } { 5pt } { 5pt } ,
3741   { 7 } { 7 } { 5pt } { 5pt } ,
3742   { -6 } { -6 } { 5pt } { 5pt } ,
3743   { 6 } { 6 } { 5pt } { 5pt } ,
3744   { -5 } { -5 } { 6pt } { 5pt } ,
3745   { 5 } { 5 } { 7pt } { 5pt } ,
3746   { -4 } { -4 } { 8pt } { 6pt } ,
3747   { 4 } { 4 } { 5 } { 6 } ,
3748   { -3 } { -3 } { -4 } { -5 } ,
3749   { 3 } { 3 } { 4 } { 5 } ,
3750   { -2 } { -2 } { -3 } { -4 } ,
3751   { 2 } { 2 } { 3 } { 4 } ,
3752   { -1 } { -1 } { -2 } { -3 } ,
3753   { 1 } { 1 } { 2 } { 3 } ,
3754   { -0 } { -0 } { -1 } { -2 } ,
3755   { 0 } { 0 } { 1 } { 2 }
3756 }
3757 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3758 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3

```

```

3759 {
3760   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3761   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3762   { \msg_error:nnn { ctex } { fontsize } {#2} }
3763 }
3764 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3765 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

3766 \if_case:w \g__ctex_font_size_flag
3767   \ctex_file_input:n { ctex-c5size.clo }
3768 \or:
3769   \ctex_file_input:n { ctex-cs4size.clo }
3770 \fi:
3771 </class>ctex|ctexsize>
3772 <*c5size>
3773 \ctex_set_font_size:Nnn \normalsize { 5 }
3774 {
3775   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3776   \abovedisplayshortskip \z@ \@plus3\p@
3777   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3778   \belowdisplayskip \abovedisplayskip
3779   \let\@listi\@listI
3780 }
3781 \ctex_set_font_size:Nnn \small { -5 }
3782 {
3783   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3784   \abovedisplayshortskip \z@ \@plus2\p@
3785   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3786   \def\@listi{\leftmargin\leftmargini
3787             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3788             \parsep 2\p@ \@plus\p@ \@minus\p@
3789             \itemsep \parsep}
3790   \belowdisplayskip \abovedisplayskip
3791 }
3792 \ctex_set_font_size:Nnn \footnotesize { 6 }
3793 {
3794   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3795   \abovedisplayshortskip \z@ \@plus\p@
3796   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3797   \def\@listi{\leftmargin\leftmargini
3798             \topsep 3\p@ \@plus\p@ \@minus\p@
3799             \parsep 2\p@ \@plus\p@ \@minus\p@
3800             \itemsep \parsep}
3801   \belowdisplayskip \abovedisplayskip
3802 }
3803 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3804 \ctex_set_font_size:Nnn \tiny { 7 } { }
3805 \ctex_set_font_size:Nnn \large { -4 } { }
3806 \ctex_set_font_size:Nnn \Large { -3 } { }
3807 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3808 \ctex_set_font_size:Nnn \huge { 2 } { }
3809 \ctex_set_font_size:Nnn \Huge { 1 } { }
3810 </c5size>
3811 <*cs4size>
3812 \ctex_set_font_size:Nnn \normalsize { -4 }
3813 {
3814   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3815   \abovedisplayshortskip \z@ \@plus3\p@
3816   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3817   \belowdisplayskip \abovedisplayskip
3818   \let\@listi\@listI
3819 }
3820 \ctex_set_font_size:Nnn \small { 5 }
3821 {
3822   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3823   \abovedisplayshortskip \z@ \@plus3\p@
3824   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@

```



```

3825 \def\@listi{\leftmargin\leftmargini
3826 \topsep 9\p@ \@plus3\p@ \@minus5\p@
3827 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3828 \itemsep \parsep}
3829 \belowdisplayskip \abovedisplayskip
3830 }
3831 \ctex_set_font_size:Nnn \footnotesize { -5 }
3832 {
3833 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3834 \abovedisplayshortskip \z@ \@plus3\p@
3835 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3836 \def\@listi{\leftmargin\leftmargini
3837 \topsep 6\p@ \@plus2\p@ \@minus2\p@
3838 \parsep 3\p@ \@plus2\p@ \@minus\p@
3839 \itemsep \parsep}
3840 \belowdisplayskip \abovedisplayskip
3841 }
3842 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3843 \ctex_set_font_size:Nnn \tiny { -6 } { }
3844 \ctex_set_font_size:Nnn \large { -3 } { }
3845 \ctex_set_font_size:Nnn \Large { -2 } { }
3846 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3847 \ctex_set_font_size:Nnn \huge { -1 } { }
3848 \ctex_set_font_size:Nnn \Huge { 1 } { }
3849 </cs4size>
<ctexsize> 3850 \normalsize
3851 <*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`。

```

3852 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3853 {
3854 \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。

```

3855 \group_begin: \footnotesize \exp_args:NNNo \group_end:
3856 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3857 }

```

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

```

3858 \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。

```

3859 \NewDocumentCommand \CTeX { }
3860 { \ifincsname CTeX \else: C \TeX \fi: }
3861 \ctex_at_end_package:nn { hyperref }
3862 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

`captiondelimit` 过时选项。

```

3863 \keys_define:nn { ctex }
3864 {
3865   captiondelimit .code:n =
3866   {
3867     \msg_warning:nnn { ctex } { deprecated-option }
3868     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3869   }
3870 }

3871 </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`

```

3872 <*scheme&chinese&(article|book|report)>
3873 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3874 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3875 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3876 </scheme&chinese&(article|book|report)>

3877 <*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。

```

3878 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3879 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3880 \msg_new:nnn { ctex } { fontset-unavailable }
3881 { 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。

```

3882 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3883 {
3884   \tl_if_empty:NTF \g__ctex_fontset_tl
3885   {
3886     \ctex_detected_platform:
3887     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3888   }
3889   {
3890     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3891     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3892     {
3893       \use:x
3894       {
3895         \ctex_detected_platform:
3896         \msg_error:nxxx { ctex } { fontset-not-found }
3897         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3898       }
3899       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3900     }
3901   }
3902 }
3903 \onlypreamble \ctex_load_fontset:
3904 \msg_new:nnnn { ctex } { fontset-not-found }
3905 {
3906   CTeX~fontset~`#1'~could~not~be~found.\\
3907   Fontset~`#2'~will~be~used~instead.
3908 }
3909 { You~may~run~`mktexlsr'~firstly. }

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

3910 \keys_define:nn { ctex }
3911 {
3912   fontset .code:n =
3913   {
3914     \ctex_if_preamble:TF
3915     {
3916       \str_if_eq_x:nnTF {#1} { none }
3917       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3918       {
3919         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3920         {
3921           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3922           \ctex_load_fontset:
3923         }
3924         {
3925           \msg_error:nxxx { ctex } { fontset-loaded }
3926           { \g__ctex_fontset_tl } {#1}
3927         }
3928       }
3929     }
3930     { \msg_error:nn { ctex } { fontset-only-preamble } }
3931   }
3932 }
3933 \msg_new:nnnn { ctex } { fontset-loaded }
3934 {
3935   CTeX~fontset~`#1'~has~been~loaded.
3936   \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
3937 }
3938 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3939 \msg_new:nnn { ctex } { fontset-only-preamble }
3940 {
3941   The~`fontset'~ option~ can~ be~ used~ only~ in~ preamble.
3942 }

```

载入中文字库。

```

3943 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3944 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

3945 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3946 </class|ctex>
3947 <*config>
3948 %%
3949 </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。

```

3950 <*ctexopts>
3951 %%
3952 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3953 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

3954 <*c19|c70>
3955 %%
3956 %% Chinese characters
3957 %%
3958 <c19> %% character set: GBK (extension of GB 2312)
3959 <c70> %% character set: Unicode
3960 %% font encoding: Unicode
3961 %%
3962 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm|c19> 3963 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm|c70> 3964 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf|c19> 3965 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf|c70> 3966 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt|c19> 3967 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt|c70> 3968 \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。

```

3969 <*uptex|aptex>
3970 \DeclareKanjiFamily{JY2}{rm}{\font\m@ne}
3971 \DeclareKanjiFamily{JY2}{sf}{\font\m@ne}
3972 \DeclareKanjiFamily{JY2}{tt}{\font\m@ne}
3973 \DeclareKanjiFamily{JT2}{rm}{\font\m@ne}
3974 \DeclareKanjiFamily{JT2}{sf}{\font\m@ne}
3975 \DeclareKanjiFamily{JT2}{tt}{\font\m@ne}
3976 </uptex|aptex>

3977 <*rm>
3978 <*c19>
3979 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3980 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3981 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3982 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3983 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3984 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3985 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3986 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3987 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3988 </c19>
3989 <*c70>
3990 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3991 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3992 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3993 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3994 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3995 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3996 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3997 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3998 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3999 </c70>
4000 </rm>
4001 <*uptex|aptex>
4002 \DeclareFontShape{JY2}{rm}{m}{n}{<-> upzhserif-h}{\font\m@ne}
4003 \DeclareFontShape{JY2}{rm}{m}{it}{<-> upzhserifit-h}{\font\m@ne}
4004 \DeclareFontShape{JY2}{rm}{bx}{n}{<-> upzhserifb-h}{\font\m@ne}
4005 \DeclareFontShape{JT2}{rm}{m}{n}{<-> upzhserif-v}{\font\m@ne}
4006 \DeclareFontShape{JT2}{rm}{m}{it}{<-> upzhserifit-v}{\font\m@ne}
4007 \DeclareFontShape{JT2}{rm}{bx}{n}{<-> upzhserifb-v}{\font\m@ne}
4008 </uptex|aptex>
4009 <*sf>

```

```

4010 <*c19>
4011 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4012 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4013 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4014 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
4015 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4016 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4017 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4018 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4019 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4020 </c19>
4021 <*c70>
4022 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
4023 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
4024 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4025 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4026 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4027 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4028 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4029 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4030 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4031 </c70>
4032 </sf>
4033 <*uptex|aptex>
4034 \DeclareFontShape{JY2}{sf}{m}{n}{<-> upzhsans-h}{\}
4035 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{\}
4036 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{\}
4037 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{\}
4038 </uptex|aptex>
4039 <*tt>
4040 <*c19>
4041 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4042 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4043 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4044 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4045 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4046 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4047 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4048 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4049 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4050 </c19>
4051 <*c70>
4052 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4053 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4054 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4055 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4056 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4057 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4058 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4059 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4060 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4061 </c70>
4062 </tt>
4063 <*uptex|aptex>
4064 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{\}
4065 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{\}
4066 </uptex|aptex>
4067 <*fontset>

```

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

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。)

```

4068 <*windows>

```

```

4069 \file_if_exist:nTF { C:/bootfont.bin }
4070 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
4071 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
4072 </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。

```

4073 <*windowsnew>
4074 \tl_new:N \l__ctex_msyh_suffix_tl
4075 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
4076 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
4077 {
4078   \file_if_exist:nF { msyh.ttc }
4079   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
4080 }
4081 </windowsnew>
4082 <*windowsnew|windowsold>
4083 \sys_if_engine_pdfTeX:TF
4084 {
4085   \ctex_zhmap_case:nnn
4086   {
4087     \ctex_punct_set:n { windows }
4088     \setCJKmainfont
4089     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
4090 <*windowsold>
4091     \setCJKsansfont { simhei.ttf }
4092     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4093 </windowsold>
4094 <*windowsnew>
4095     \setCJKsansfont [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4096     \setCJKfamilyfont { zhyahei }
4097     [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4098     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4099     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
4100 </windowsnew>
4101     \setCJKmonofont { simfang.ttf }
4102     \setCJKfamilyfont { zhkai } { simkai.ttf }
4103     \setCJKfamilyfont { zhfs } { simfang.ttf }
4104     \setCJKfamilyfont { zhsong } { simsun.ttc }
4105     \setCJKfamilyfont { zhhei } { simhei.ttf }
4106     \setCJKfamilyfont { zhli } { simli.ttf }
4107     \setCJKfamilyfont { zhyou } { simyou.ttf }
4108     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4109     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4110     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4111     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4112   }
4113   {
4114     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
4115     \ctex_punct_set:n { windows }
4116     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4117     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4118     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4119   }
4120   {
4121     \tl_set:Nn \CJKrmdefault { rm }
4122     \tl_set:Nn \CJKsfdefault { sf }
4123     \tl_set:Nn \CJKttdefault { tt }
4124   }
4125 }
4126 {
4127   \sys_if_engine_uptex:TF
4128   {
4129     \ctex_set_upfonts:nnnnnn
4130     {simsun.ttc} {simhei.ttf} {simkai.ttf}

```

```

4131      {msyh\l__ctex_msyh_suffix_tl} {msyhbd\l__ctex_msyh_suffix_tl}
4132      {simhei.ttf} {simhei.ttf}
4133      {simfang.ttf}
4134      \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4135      \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4136      \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4137      \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4138      \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4139      \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4140      \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4141      \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4142      \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4143    }
4144    {
4145      <*windowsold>
4146        \setCJKmainfont
4147          [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
4148        \setCJKsansfont { SimHei }
4149        \setCJKmonofont { FangSong_GB2312 }
4150        \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
4151        \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
4152      </windowsold>
4153      <*windowsnew>
4154        \setCJKmainfont
4155          [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
4156        \setCJKsansfont
4157          [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4158        \setCJKmonofont { FangSong }
4159        \setCJKfamilyfont { zhkai } { KaiTi }
4160        \setCJKfamilyfont { zhfs } { FangSong }
4161      </windowsnew>
4162        \setCJKfamilyfont { zhsong } { SimSun }
4163        \setCJKfamilyfont { zhhei } { SimHei }
4164        \setCJKfamilyfont { zhli } { LiSu }
4165        \setCJKfamilyfont { zhyou } { YouYuan }
4166      <*windowsnew>
4167        \setCJKfamilyfont { zhyahei }
4168          [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4169      </windowsnew>
4170    }
4171  }
4172 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```
4173 <*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。

```

4174 \sys_if_engine_pdftex:TF
4175 {
4176   \sys_if_output_pdf:TF
4177     { \ctex_fontset_error:n { adobe } }
4178     {
4179       \ctex_zhmap_case:nnn
4180       {
4181         \setCJKmainfont
4182           [
4183             cmap = UniGB-UTF16-H ,
4184             BoldFont = AdobeHeitiStd-Regular.otf ,
4185             ItalicFont = AdobeKaitiStd-Regular.otf
4186           ] { AdobeSongStd-Light.otf }
4187         \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4188         \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4189         \setCJKfamilyfont { zhsong }
4190           [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }

```



```

4191 \setCJKfamilyfont { zhhei }
4192 [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4193 \setCJKfamilyfont { zhkai }
4194 [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
4195 \setCJKfamilyfont { zhfs }
4196 [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4197 \ctex_punct_set:n { adobe }
4198 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4199 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4200 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4201 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4202 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4203 }
4204 {
4205 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
4206 \ctex_punct_set:n { adobe }
4207 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4208 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4209 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4210 }
4211 { \ctex_fontset_error:n { adobe } }
4212 }
4213 }
4214 {
4215 \sys_if_engine_uptex:TF
4216 {
4217 \ctex_set_upfonts:nnnnnn
4218 {AdobeSongStd-Light.otf}{AdobeHeitiStd-Regular.otf}{AdobeKaitiStd-Regular.otf}
4219 {AdobeHeitiStd-Regular.otf}{AdobeHeitiStd-Regular.otf}
4220 {AdobeFangsongStd-Regular.otf}
4221 \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4222 \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4223 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4224 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4225 }
4226 {
4227 \setCJKmainfont
4228 [
4229 BoldFont = AdobeHeitiStd-Regular ,
4230 ItalicFont = AdobeKaitiStd-Regular
4231 ] { AdobeSongStd-Light }
4232 \setCJKsansfont { AdobeHeitiStd-Regular}
4233 \setCJKmonofont { AdobeFangsongStd-Regular}
4234 \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4235 \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4236 \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4237 \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4238 }
4239 }
4240 </adobe>
4241 <*fandol>

```

14.15.4 ctex-fontset-fandol.def

```

4242 \sys_if_engine_pdftex:TF
4243 {
4244 \sys_if_output_pdf:TF
4245 { \ctex_fontset_error:n { fandol } }
4246 {
4247 \ctex_zhmap_case:nnn
4248 {
4249 \setCJKmainfont
4250 [
4251 cmap = UniGB-UTF16-H ,
4252 BoldFont = FandolSong-Bold.otf ,
4253 ItalicFont = FandolKai-Regular.otf
4254 ] { FandolSong-Regular.otf }

```



```

4255 \setCJKsansfont
4256 [
4257     cmap = UniGB-UTF16-H ,
4258     BoldFont = FandolHei-Bold.otf
4259 ] { FandolHei-Regular.otf }
4260 \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4261 \setCJKfamilyfont { zhsong }
4262 [
4263     cmap = UniGB-UTF16-H ,
4264     BoldFont = FandolSong-Bold.otf
4265 ] { FandolSong-Regular.otf }
4266 \setCJKfamilyfont { zhhei }
4267 [
4268     cmap = UniGB-UTF16-H ,
4269     BoldFont = FandolHei-Bold.otf
4270 ] { FandolHei-Regular.otf }
4271 \setCJKfamilyfont { zhfs }
4272 [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4273 \setCJKfamilyfont { zhkai }
4274 [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4275 \ctex_punct_set:n { fandol }
4276 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4277 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4278 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4279 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4280 \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4281 \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4282 }
4283 {
4284     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4285     \ctex_punct_set:n { fandol }
4286     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4287     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4288     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4289 }
4290 { \ctex_fontset_error:n { fandol } }
4291 }
4292 }
4293 {
4294     \sys_if_engine_uptex:TF
4295     {
4296         \ctex_set_upfonts:nnnnnn
4297         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4298         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4299         {FandolFang-Regular.otf}
4300         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4301         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4302         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4303         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4304     }
4305     {
4306         \setCJKmainfont
4307         [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
4308         { FandolSong-Regular }
4309         \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4310         \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4311         \setCJKfamilyfont { zhsong }
4312         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4313         \setCJKfamilyfont { zhhei }
4314         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4315         \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4316         \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4317     }
4318 }
4319 </fandol>
4320 <*mac>

```

14.15.5 ctex-fontset-mac.def

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

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

在 DVIPDFMx 下,可以通过下述方式使用华文宋体和华文楷体:

```
\special{pdf:mapline unisong@Unicode@ unicode :4:Songti.ttc}
\special{pdf:mapline unikai@Unicode@ unicode :4:Kaiti.ttc}
```

而 \pdfmapline 似乎不支持带索引的 ttc 字体,Songti.ttc 默认使用的是 Songti SC Black, Kaiti.ttc 默认使用的是 Kaiti SC Black。华文黑体不能通过这种方式使用:

```
\special{pdf:mapline unihei@Unicode@ unicode \detokenize{华文黑体}.ttf}
```

DVIPDFMx 不能生成 PDF,报下述错误:

```
** WARNING ** UCS-4 TrueType cmap table...
** ERROR ** Unable to read OpenType/TrueType Unicode cmap table.
```

如果将 CMap 改为 UniGB-UTF16-H,错误信息是

```
** WARNING ** No usable TrueType cmap table found for font "华文黑体.ttf".
** WARNING ** CID character collection for this font is set to "Adobe-GB1"
** ERROR ** Cannot continue without this...
```

在 pdfTeX 下生成的 PDF 只有方框²³。华文细黑和华文仿宋的情况类似。这确认为 DVIPDFMx 的 bug²⁴。

4321 \sys_if_engine_pdfTeX:TF

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

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>

```

4322 { \ctex_fontset_error:n { mac } }
4323 {
4324   \sys_if_engine_uptex:TF
4325     { \ctex_fontset_error:n { mac } }
4326     {
4327       \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4328       \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4329       \setCJKmonofont { STFangsong }
4330       \setCJKfamilyfont { zhsong } { STSong }
4331       \setCJKfamilyfont { zhhei } { STHeiti }
4332       \setCJKfamilyfont { zhfs } { STFangsong }
4333       \setCJKfamilyfont { zhkai } { STKaiti }
4334     }
4335   }
4336 </mac>
4337 <*founder>

```

14.15.6 ctex-fontset-founder.def

```

4338 \sys_if_engine_pdftex:TF
4339 {
4340   \ctex_zhmap_case:nnn
4341   {
4342     \setCJKmainfont
4343       [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4344     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4345     \setCJKmonofont { FZFSK.TTF }
4346     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4347     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4348     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4349     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4350     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4351     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4352     \ctex_punct_set:n { founder }
4353     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4354     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4355     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4356     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4357     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4358     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4359     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4360   }
4361   {
4362     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4363     \ctex_punct_set:n { founder }
4364     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4365     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4366     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4367   }
4368   { \ctex_fontset_error:n { founder } }
4369 }
4370 {
4371   \sys_if_engine_uptex:TF
4372   {
4373     \ctex_set_upfonts:nnnnnn
4374       { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4375       { FZXH1K.TTF } { FZHTK.TTF }
4376       { FZFSK.TTF }
4377     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4378     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4379     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4380     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4381     \ctex_set_upfamily:nnn { zhli } { upschrm } {}
4382     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4383     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4384     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}

```

```

4385     }
4386     {
4387         \setCJKmainfont
4388         [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }

```

在 WPS For Linux 附带的 5.00 版和家庭版 5.20 版的“方正细黑一_GBK”的字体名称是 FZXiHeiI-Z08。但在网上发现不少文档和资料都是 FZXiHeiI-Z08,而在官方资料《2013 方正字库字体样张》中对应的英文名字是 FZXiHeiI。可以用 Postscript 名字 FZXH1K--GBK1-0 来统一。经测试时发现(WPS 中的字体),X_YTeX 找该字体时会出现明显的延迟,而用字体文件名 FZXH1K.TTF 又可能会出现大小写问题,遂采用汉字名称。由于 luaotfload 不支持汉字名称,故使用 Postscript 名字,速度不受影响。

```

4389         \setCJKsansfont [ BoldFont = FZHei-B01 ]
4390         { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4391         \setCJKmonofont { FZFangSong-Z02 }
4392         \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4393         \setCJKfamilyfont { zhhei } { FZHei-B01 }
4394         \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4395         \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4396         \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4397         \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4398     }
4399 }
4400 </founder>
4401 <*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。

```

4402 \sys_if_engine_pdftex:TF
4403 {
4404     \ctex_zhmap_case:nnn
4405     {
4406         \setCJKmainfont
4407         [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4408         \setCJKsansfont { wqy-zenhei.ttc }
4409         \setCJKmonofont { uming.ttc }
4410         \setCJKfamilyfont { zhsong } { uming.ttc }
4411         \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4412         \setCJKfamilyfont { zhkai } { ukai.ttc }
4413         \ctex_punct_set:n { ubuntu }
4414         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4415         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4416         \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4417         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4418         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4419     }
4420     {
4421         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4422         \ctex_punct_set:n { ubuntu }
4423         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4424         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4425         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4426     }
4427     { \ctex_fontset_error:n { ubuntu } }
4428 }

```

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

```

4429 {
4430   \sys_if_engine_uptex:TF
4431   {
4432     \ctex_set_upfonts:nnnnnn
4433     { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4434     { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4435     { uming.ttc }
4436     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4437     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4438     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4439   }
4440   {
4441     \setCJKmainfont
4442     [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4443     \setCJKsansfont { WenQuanYi~Zen~Hei }
4444     \setCJKmonofont { AR~PL~UMing~CN }
4445     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4446     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4447     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4448   }
4449 }
4450 </ubuntu>

```

14.15.8 中文字体命令

```

4451 <*/windows>
4452 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4453 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
4454 <ubuntu> \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4455 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4456 <*windowsnew|windowsold|founder>
4457 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4458 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4459 </windowsnew|windowsold|founder>
4460 <windowsnew> \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4461 </!windows>
4462 </fontset>
4463 <*/zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4464 \begingroup\catcode61\catcode48\catcode32=10\relax%
4465 \catcode 35=6 % #
4466 \catcode123=1 % {
4467 \catcode125=2 % }
4468 \toks0{\endlinechar=\the\endlinechar\relax}%
4469 \toks2{\endlinechar=13 }%
4470 \def\x#1 #2 {%
4471   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4472   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4473 \x 13 5 % carriage return
4474 \x 32 10 % space
4475 \x 35 6 % #
4476 \x 40 12 % (
4477 \x 41 12 % )
4478 \x 45 12 % -
4479 \x 46 12 % .
4480 \x 47 12 % /
4481 \x 58 12 % :
4482 \x 60 12 % <
4483 \x 61 12 % =
4484 \x 64 11 % @
4485 \x 91 12 % [

```

```

4486 \x 93 12 % ]
4487 \x 123 1 % {
4488 \x 125 2 % }
4489 \edef\x#1{\endgroup%
4490 \edef\noexpand#1{%
4491 \the\toks0 %
4492 \let\noexpand\noexpand\noexpand#1%
4493 \noexpand\noexpand\noexpand\undefined%
4494 \noexpand\noexpand\noexpand\endinput}%
4495 \the\toks2}%
4496 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4497 \begingroup\expandafter\endgroup
4498 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4499 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4500 \expandafter\ifx\csname pdfoutput\endcsname\relax
4501 \iffalse\else\ifnum\pdfoutput < 1 \iffalse\else \iftrue\fi\fi
4502 \else ifctexpdf\fi
4503 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4504 \begingroup
4505 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4506 \long\def\x#1\ProvidesFile#2[#3]{%
4507 #1%
4508 \immediate\write-1{File: #2 #3}%
4509 \expandafter\xdef\csname ver@#2\endcsname{#3}}
4510 \expandafter\x%
4511 \fi
4512 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4513 <*windows>
4514 \ProvidesFile{zhwindowsfonts.tex}%
4515 [2016/12/27 v2.4.7 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4516
4517 \ifzhmappdf
4518 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4519 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4520 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4521 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4522 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4523 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4524 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4525
4526 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4527 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4528 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4529 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4530 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4531 \pdfmapline{=unili@Unicode@ <simli.ttf}
4532 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4533
4534 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4535 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4536 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4537 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4538 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4539 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4540
4541 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4542 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4543 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4544 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4545 \pdfmapline{=unilisl@Unicode@ <simli.ttf}

```



```

4546 \pdfmapline{=uniyouisl@Unicode@ <simyou.ttf}
4547
4548 \else
4549 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4550 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4551 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4552 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4553 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4554 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4555 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4556
4557 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4558 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4559 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4560 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4561 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4562 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4563 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4564
4565 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4566 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4567 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4568 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4569 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4570 \special{pdf:mapline gbkyouisl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4571
4572 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4573 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4574 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4575 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4576 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4577 \special{pdf:mapline uniyouisl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4578
4579 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4580 <*adobe>
4581 \ProvidesFile{zhadobefonts.tex}%
4582 [2016/12/27 v2.4.7 Adobe font map loader for DVIPDFMx (CTEX)]
4583
4584 \ifzhmappdf
4585 %% pdfTeX does not support OTF fonts
4586
4587 \else
4588 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4589 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4590 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4591 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4592 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4593
4594 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4595 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4596 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4597 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4598 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4599
4600 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4601 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4602 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4603 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4604
4605 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4606 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4607 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4608 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4609
4610 </adobe>

```

14.15.9.3 zhfandolfonts.tex

```

4611 <*fandol>
4612 \ProvidesFile{zhfandolfonts.tex}%
4613 [2016/12/27 v2.4.7 Fandol font map loader for DVIPDFMx (CTEX)]
4614
4615 \ifzhmappdf
4616 %% pdfTeX does not support OTF fonts
4617
4618 \else
4619 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4620 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4621 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4622 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4623 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4624
4625 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4626 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4627 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4628 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4629 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4630
4631 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4632 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4633 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4634 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4635
4636 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4637 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4638 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4639 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4640
4641 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4642 <*founder>
4643 \ProvidesFile{zhfounderfonts.tex}%
4644 [2016/12/27 v2.4.7 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4645
4646 \ifzhmappdf
4647 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4648 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4649 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4650 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4651 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4652 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4653 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4654
4655 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4656 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4657 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4658 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4659 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4660 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4661 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4662
4663 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4664 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4665 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4666 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4667 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4668 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4669
4670 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4671 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4672 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4673 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}

```



```

4674 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4675 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4676
4677 \else
4678 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4679 \special{pdf:mapline gbkkai@UGBK@ unicode FZSSK.TTF}
4680 \special{pdf:mapline gbkkai@UGBK@ unicode FZSSK.TTF}
4681 \special{pdf:mapline gbkkai@UGBK@ unicode FZSSK.TTF}
4682 \special{pdf:mapline gbkkai@UGBK@ unicode FZSSK.TTF}
4683 \special{pdf:mapline gbkkai@UGBK@ unicode FZSSK.TTF}
4684 \special{pdf:mapline gbkkai@UGBK@ unicode FZSSK.TTF}
4685
4686 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4687 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4688 \special{pdf:mapline unikai@Unicode@ unicode FZSSK.TTF}
4689 \special{pdf:mapline unihei@Unicode@ unicode FZSSK.TTF}
4690 \special{pdf:mapline unifs@Unicode@ unicode FZSSK.TTF}
4691 \special{pdf:mapline unili@Unicode@ unicode FZSSK.TTF}
4692 \special{pdf:mapline uniyousl@Unicode@ unicode FZSSK.TTF}
4693
4694 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4695 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZSSK.TTF -s .167}
4696 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZSSK.TTF -s .167}
4697 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZSSK.TTF -s .167}
4698 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZSSK.TTF -s .167}
4699 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZSSK.TTF -s .167}
4700
4701 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4702 \special{pdf:mapline unikaisl@Unicode@ unicode FZSSK.TTF -s .167}
4703 \special{pdf:mapline uniheisl@Unicode@ unicode FZSSK.TTF -s .167}
4704 \special{pdf:mapline unifssl@Unicode@ unicode FZSSK.TTF -s .167}
4705 \special{pdf:mapline unilisl@Unicode@ unicode FZSSK.TTF -s .167}
4706 \special{pdf:mapline uniyousl@Unicode@ unicode FZSSK.TTF -s .167}
4707
4708 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4709 <*ubuntu>
4710 \ProvidesFile{zhubuntufonts.tex}%
4711 [2016/12/27 v2.4.7 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4712
4713 \ifzhmappdf
4714 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4715 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4716 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4717 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4718 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4719 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4720
4721 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4722 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4723 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4724 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4725 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4726 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4727
4728 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4729 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4730 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4731 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4732 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4733
4734 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4735 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4736 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4737 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4738 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}

```

```

4739
4740 \else
4741 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4742 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4743 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4744 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4745 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4746
4747 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4748 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4749 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4750 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4751 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4752
4753 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4754 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4755 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4756 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4757
4758 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4759 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4760 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4761 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4762
4763 </ubuntu>
4764 \fi
4765
4766 \ctex@zhmap@endinput
4767 </zhmap>

```

14.15.10 制作 spa 文件

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

```

4768 <*spa>
4769 <*macro>
4770 \input expl3-generic %
4771 \ExplSyntaxOn
4772 \sys_if_engine_xetex:F
4773 {
4774   \msg_new:nn { ctex } { xetex }
4775   { XeTeX~is~required~to~compile~this~document! }
4776   \msg_fatal:nn { ctex } { xetex }
4777 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

4778 \seq_new:N \c__ctex_punct_seq
4779 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4780 {
4781   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4782   "3008 , "300A , "3016 , "3010 ,
4783   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4784   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4785   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4786 }

```

\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。

```

4787 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4788 {

```

```

4789 \iow_open:Nn \g__ctex_spa_iow {#1}
4790 \clist_map_inline:nn {#2}
4791 { \__ctex_write_family:nn ##1 }
4792 \iow_close:N \g__ctex_spa_iow
4793 }
4794 \iow_new:N \g__ctex_spa_iow
4795 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

4796 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4797 {
4798   \group_begin:
4799   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4800   \l__ctex_punct_font
4801   \clist_clear:N \l__ctex_punct_bounds_clist
4802   \seq_map_inline:Nn \c__ctex_punct_seq
4803   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4804   \iow_now:Nx \g__ctex_spa_iow
4805   {
4806     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4807       { \l__ctex_punct_bounds_clist , , , }
4808     }
4809   \group_end:
4810 }
4811 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4812 {
4813   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4814   {
4815     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4816     \__ctex_calc_bounds:nn { \c_three } {#1}
4817   }
4818 }
4819 \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。

```

4820 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4821 {
4822   \fp_eval:n
4823   {
4824     round
4825     (
4826       \dim_to_decimal_in_unit:nn
4827       { 100 \xetex_glyphbounds:D #1 ~ #2 }
4828       { 1 em }
4829     )
4830   }
4831 }
4832 \ExplSyntaxOff
4833 </macro>

```

下面是 CT_EX 定义的一些字体。

```

4834 <*make>
4835 \input ctexspamacro %
4836
4837 \MAKESPA {ctexpunct.spa}
4838 {
4839   {adobezhsong}      {AdobeSongStd-Light} ,
4840   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4841   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4842   {adobezhfs}        {AdobeFangsongStd-Regular} ,
4843   {fandolzhsong}     {FandolSong} ,
4844   {fandolzhsongb}    {FandolSong-Bold} ,
4845   {fandolzhhei}      {FandolHei} ,
4846   {fandolzhheib}     {FandolHei-Bold} ,
4847   {fandolzhkai}      {FandolKai} ,

```

```

4848 {fandolzhfs} {FandolFang} ,
4849 {founderzhsong} {FZShuSong-Z01} ,
4850 {founderzhsongb} {FZXiaoBiaoSong-B05} ,
4851 {founderzhhei} {FZHei-B01} ,
4852 {founderzhheil} {FZXiHeiI-Z08} ,
4853 {founderzhkai} {FZKai-Z03} ,
4854 {founderzhfs} {FZFangSong-Z02} ,
4855 {founderzhli} {FZLiShu-S01} ,
4856 {founderzhyou} {FZXiYuan-M01} ,
4857 {founderzhyoub} {FZZhunYuan-M02} ,
4858 {ubuntuzhsong} {AR PL Uming CN} ,
4859 {ubuntuzhhei} {WenQuanYi Zen Hei} ,
4860 {ubuntuzhkai} {AR PL UKai CN} ,
4861 {windowszhsong} {SimSun} ,
4862 {windowszhhei} {SimHei} ,
4863 {windowszhkai} {KaiTi} ,
4864 {windowszhfs} {FangSong} ,
4865 {windowszhli} {LiSu} ,
4866 {windowszhyou} {YouYuan} ,
4867 {windowszhyawei} {Microsoft YaHei} ,
4868 {windowszhyaweiB} {Microsoft YaHei Bold}
4869 }
4870
4871 \primitive\end
4872 </make>
4873 </spa>

```

14.16 translator 宏包的中文字典

```
4874 <*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。

```

4875 <*theorem>
4876 \providetranslation{Comments}{评论}
4877 \providetranslation{comments}{评论}
4878 \providetranslation{Comment}{评论}
4879 \providetranslation{comment}{评论}
4880 \providetranslation{Corollaries}{推论}
4881 \providetranslation{corollaries}{推论}
4882 \providetranslation{Corollary}{推论}
4883 \providetranslation{corollary}{推论}
4884 \providetranslation{Definitions}{定义}
4885 \providetranslation{definitions}{定义}
4886 \providetranslation{Definition}{定义}
4887 \providetranslation{definition}{定义}
4888 \providetranslation{Examples}{例}
4889 \providetranslation{examples}{例}
4890 \providetranslation{Example}{例}
4891 \providetranslation{example}{例}
4892 \providetranslation{Exercises}{练习}
4893 \providetranslation{exercises}{练习}
4894 \providetranslation{Exercise}{练习}
4895 \providetranslation{exercise}{练习}
4896 \providetranslation{Facts}{事实}
4897 \providetranslation{facts}{事实}
4898 \providetranslation{Fact}{事实}
4899 \providetranslation{fact}{事实}
4900 \providetranslation{Key Lemmas}{关键引理}
4901 \providetranslation{key lemmas}{关键引理}
4902 \providetranslation{Key Lemma}{关键引理}
4903 \providetranslation{key lemma}{关键引理}
4904 \providetranslation{Key Observations}{关键观察}
4905 \providetranslation{key observations}{关键观察}
4906 \providetranslation{Key Observation}{关键观察}

```

```

4907 \providetranslation{key observation}{关键观察}
4908 \providetranslation{Lemmas}{引理}
4909 \providetranslation{lemmas}{引理}
4910 \providetranslation{Lemma}{引理}
4911 \providetranslation{lemma}{引理}
4912 \providetranslation{Main Theorems}{主要定理}
4913 \providetranslation{main theorems}{主要定理}
4914 \providetranslation{Main Theorem}{主要定理}
4915 \providetranslation{main theorem}{主要定理}
4916 \providetranslation{Observations}{观察}
4917 \providetranslation{observations}{观察}
4918 \providetranslation{Observation}{观察}
4919 \providetranslation{observation}{观察}
4920 \providetranslation{Problems}{问题}
4921 \providetranslation{problems}{问题}
4922 \providetranslation{Problem}{问题}
4923 \providetranslation{problem}{问题}
4924 \providetranslation{Proofs}{证明}
4925 \providetranslation{proofs}{证明}
4926 \providetranslation{Proof}{证明}
4927 \providetranslation{proof}{证明}
4928 \providetranslation{Proof Sketch}{证明提要}
4929 \providetranslation{Proof sketch}{证明提要}
4930 \providetranslation{proof sketch}{证明提要}
4931 \providetranslation{Proof Sketches}{证明提要}
4932 \providetranslation{Proof sketches}{证明提要}
4933 \providetranslation{proof sketches}{证明提要}
4934 \providetranslation{Sketch of Proof}{证明提要}
4935 \providetranslation{Sketch of Proofs}{证明提要}
4936 \providetranslation{Sketch of proof}{证明提要}
4937 \providetranslation{Sketch of proofs}{证明提要}
4938 \providetranslation{sketch of proof}{证明提要}
4939 \providetranslation{sketch of proofs}{证明提要}
4940 \providetranslation{Propositions}{命题}
4941 \providetranslation{propositions}{命题}
4942 \providetranslation{Proposition}{命题}
4943 \providetranslation{proposition}{命题}
4944 \providetranslation{Remarks}{注}
4945 \providetranslation{remarks}{注}
4946 \providetranslation{Remark}{注}
4947 \providetranslation{remark}{注}
4948 \providetranslation{Solutions}{解}
4949 \providetranslation{solutions}{解}
4950 \providetranslation{Solution}{解}
4951 \providetranslation{solution}{解}
4952 \providetranslation{Theorems}{定理}
4953 \providetranslation{theorems}{定理}
4954 \providetranslation{Theorem}{定理}
4955 \providetranslation{theorem}{定理}
4956 </theorem>
4957 </dict>

```

14.17 ctexcap 宏包

```
4958 <*ctexcap>
```

ctexcap 是过时宏包。

```

4959 \clist_new:N \l__ctex_ctexcap_options_clist
4960 \clist_set:Nx \l__ctex_ctexcap_options_clist
4961 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
4962 \msg_new:nnn { ctexcap } { deprecated }
4963 {
4964   Package~`ctexcap'~is~deprecated.\\
4965   Please~use~package~`ctex'~with~option~`#1'~instead: \\\
4966   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4967 }
4968 \msg_warning:nnx { ctexcap } { deprecated }

```

```
4969 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }
```

`ctexcap` 是默认打开 `heading` 选项的 `ctex`。

```
4970 \PassOptionsToPackage { heading = true } { ctexcap }
4971 \RequirePackageWithOptions { ctex }
4972 </ctexcap>
```

14.18 ctexhook 宏包

```
4973 <*ctexhook>
```

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

```
\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4974 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4975 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
4976 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4977 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
4978 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4979 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4980 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4981 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4982 \cs_set_nopar:Npx \document
4983 {
4984   \CTEX@document@left@hook
4985   \exp_not:o { \document }
4986   \CTEX@document@right@hook
4987 }
4988 \tl_new:N \g__ctex_end_preamble_hook_tl
4989 \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。

```
\ctex_at_end_package:nn
4990 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4991 {
4992   \@ifpackageloaded {#1}
4993     {#2}
4994     { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
4995 }
```

给钩子附加内容。

```
\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn
4996 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4997 {
4998   \tl_if_exist:NF #1 { \tl_new:N #1 }
4999   \tl_gput_right:Nn #1 {#2}
5000 }
5001 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }
```

宏包末尾钩子, 只执行一次, 用后清除。

```
\ctex_package_end_hook:n
\ctex_package_end_hook:o
5002 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
5003 {
5004   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
5005     { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
5006 }
5007 \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。

```
5008 \tl_put_left:Nn \@popfilename
5009 {
5010   \cs_if_eq:NNT \@currentx \pkgextension
5011     { \ctex_package_end_hook:o { \@currname } }
5012 }
5013 </ctexhook>
```

14.19 ctexpatch 宏包

5014 `<*ctexpatch>`

5015 `\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。

5016 `\cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2`

```
5017 {
5018   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5019   { once } {#2} { \use_i:nn } { \use_ii:nn }
5020 }
```

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

5021 `\cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2`

```
5022 {
5023   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5024   { all } {#2} { \use_i:nn } { \use_ii:nn }
5025 }
```

`\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。

```
5026 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5027 {
5028   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5029   { once }
5030   {
5031     \ExplSyntaxOff
5032     \char_set_catcode_letter:n { 64 }
5033   }
5034   { }
5035   { \ctex_patch_failure:N #1 }
5036 }
5037 \cs_new_protected:Npn \ctex_patch_failure:N #1
5038 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5039 \msg_new:nnn { ctex } { patch-failure }
5040 { Oops!~Command~`#1'~is~NOT~patchable.\\ }
```

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

```
5041 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5042 {
5043   \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
5044   { left } {#2} { \use_i:nn } { \use_ii:nn }
5045 }
```

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

```
5046 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5047 {
5048   \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
5049   { right } {#2} { \use_i:nn } { \use_ii:nn }
5050 }
```

`\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。

```
5051 \cs_new_protected:Npn \ctex_patch_boot:NnnnTF #1#2#3#4#5#6
5052 {
5053   \tl_set:Nn \__ctex_patch_true:w {#5}
5054   \tl_set:Nn \__ctex_patch_false:w {#6}
5055   \group_begin:
5056   \char_set_catcode_other:n { 35 }
5057   \ctex_parse_name:NN #1 #2 {#3} {#4}
5058 }
```

```

5059 \tl_new:N \__ctex_patch_true:w
5060 \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。

```

5061 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5062 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5063 \group_begin:
5064 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5065 {
5066   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5067   {
5068     \bool_if:nTF
5069     {
5070       \cs_if_exist_p:c { ##3 ~ } ||
5071       \cs_if_exist_p:c { #1##3 }
5072     }
5073     {
5074       \group_begin:
5075       \use:x
5076       {
5077         \group_end:
5078         \__ctex_parse_name:nNNNnN
5079         { \token_get_replacement_spec:N ##2 }
5080         \exp_not:N ##2
5081         \exp_not:c { ##3 ~ }
5082         \exp_not:c { #1##3 }
5083       } {##3} ##1
5084     }
5085     { ##1##2 }
5086   }
5087   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5088   {
5089     \exp_args:Nc ##6
5090     {
5091       \str_case:nnTF {##1}
5092       {
5093         { \protect ##3 } { }
5094         { \x@protect ##2 \protect ##3 } { }
5095       }
5096       {
5097         \str_if_eq_x:nnTF
5098         { \exp_not:n { #1@protected@ ##3 #1##3 } }
5099         {
5100           \exp_last_unbraced:Nf \__ctex_parse_name:w
5101           \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
5102         }
5103         { #1##5 ~ } { ##5 ~ }
5104       }
5105     }
5106     \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }

```



```

5107         {
5108             { #1@protected@ ##2 ##4 } { }
5109             { #1@ ##4 } { }
5110         }
5111         { #1##5 } {##5}
5112     }
5113 }
5114 }
5115 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5116 }
5117 \use:x
5118 {
5119     \__ctex_tmp:w
5120     { \c_backslash_str }
5121     { \c_left_brace_str }
5122     { \tl_to_str:n { testopt } }
5123 }
5124 \group_end:
5125 \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。

```

5126 \str_new:N \l__ctex_prefix_str
5127 \str_new:N \l__ctex_parameter_str
5128 \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。

```

5129 \group_begin:
5130 \cs_set_protected:Npn \__ctex_tmp:w #1
5131 {
5132     \prg_new_protected_conditional:Npnn
5133     \ctex_get_macro_meaning:N ##1 { TF }
5134     {
5135         \exp_after:wN \__ctex_get_macro_meaning:w
5136         \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5137     }
5138     \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5139     ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5140     {
5141         \tl_if_empty:nTF { ##4 }
5142         { \prg_return_false: }
5143         {
5144             \str_set:Nn \l__ctex_prefix_str { ##1 }
5145             \str_set:Nn \l__ctex_parameter_str { ##2 }
5146             \str_set:Nn \l__ctex_replacement_str { ##3 }
5147             \prg_return_true:
5148         }
5149     }
5150 }
5151 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5152 \group_end:

```

```

\ctex_if_rescanable:NnTF

```

检查宏是否可以重建。

```

5153 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5154 {
5155     \ctex_get_macro_meaning:NTF #1
5156     {
5157         \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5158         \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5159     }
5160     {#4}
5161 }
5162 \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。

```

5163 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5164 {
5165   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl      \l__ctex_prefix_str      {#2}
5166   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl   \l__ctex_parameter_str   {#2}
5167   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5168   \use:x
5169   {
5170     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5171     \exp_not:o { \l__ctex_parameter_tl }
5172     { \exp_not:o { \l__ctex_replacement_tl } }
5173   }
5174 }
5175 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5176 {
5177   \str_if_empty:NTF #2
5178   { \tl_clear:N #1 }
5179   { \tl_set_rescan:Nno #1 {#3} {#2} }
5180 }
5181 \tl_new:N \l__ctex_prefix_tl
5182 \tl_new:N \l__ctex_parameter_tl
5183 \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。

```

5184 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5185 {
5186   \group_end:
5187   \ctex_if_rescanable:NnTF #1 {#3}
5188   {
5189     \use:x
5190     {
5191       \__ctex_patch_replace:nnnTF {#2}
5192       { \tl_to_str:n {#4} }
5193       { \tl_to_str:n {#5} }
5194     }
5195     {
5196       \__ctex_patch_rebuild:Nn #1 {#3}
5197       \__ctex_patch_true:w
5198     }
5199     { \__ctex_patch_false:w }
5200   }
5201   { \__ctex_patch_false:w }
5202 }

```

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

```

5203 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5204 {
5205   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5206   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5207 }

```

`__ctex_hookto_cmd:Nnnnw` 在宏的前/后附加钩子。

```

5208 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5209 {
5210   \group_end:
5211   \ctex_get_macro_meaning:NTF #1
5212   {
5213     \str_if_empty:NTF \l__ctex_parameter_str
5214     { \__ctex_hookto_cmd_parameterless:Nnnnw }
5215     { \__ctex_hookto_cmd_parameter:Nnnnw }
5216     #1 {#2} {#3} {#4}
5217   }
5218   { \__ctex_patch_false:w }
5219 }

```

`__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。

```

5220 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5221 {
5222   \str_if_empty:NF \l__ctex_prefix_str
5223   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5224   \tex_edef:D #1
5225   {
5226     \use:c { __ctex_ #2 _hook_aux:nn }
5227     { \exp_not:o {#1} }
5228     { \exp_not:n {#4} }
5229   }
5230   \__ctex_patch_true:w
5231 }
5232 \cs_generate_variant:Nn \tl_rescan:nn { no }
5233 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5234 \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。

```

5235 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5236 {
5237   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5238   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5239   {
5240     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5241     \__ctex_patch_rebuild:Nn #1 {#3}
5242     \__ctex_patch_true:w
5243   }
5244   { \__ctex_patch_false:w }
5245 }
5246 </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> 。 82
<code>captiondelimiter</code> 是过时选项。 109		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。 100
<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>fnctef</code> 成为过时选项, 原选项功能总是打开。 40		<code>hyperref</code> : 补充定义 <code>\hypersetup</code> 。 40
<code>hyperref</code> 成为过时选项, 原选项功能总是打开。 40		v2.2 (2015/06/21 – 2015/06/30)
<code>indent</code> , <code>noindent</code> 是过时选项。 38		General: <code>before skip</code> 和 <code>after skip</code> 选项的符号不再有特殊意义。 82
<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。 85
<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。 85
<code>nozhmap</code> 是过时选项。 39		给 <code>enumitem</code> 宏包注册 <code>\chinese</code> 和 <code>\zhnum</code> 。 80
<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。 130
<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。 100
<code>\CTEXindent</code> , <code>\CTEXnoindent</code> 是过时命令。 79		删去 <code>etoolbox</code> 与 <code>breqn</code> 的兼容补丁。 110
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。 14, 76		新的标题格式选项 <code>afterindent</code> 。 82
<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> 。 82
标题设置新增 <code>pagestyle</code> 选项。 83		新的标题格式选项 <code>runin</code> 。 82
调整 <code>\footnotesep</code> 的大小, 以适合行距的变化。 109		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> 。 80		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> 。 106		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。 110		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。 95
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。 82
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。 77		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。 128
新增 <code>zhmCJK</code> 支持选项。 39		提供 <code>up\LaTeX</code> 的 NFSS 字体定义。 112
新增 <code>zihao</code> 选项。 37		正确更新 <code>CJK</code> 包的 <code>\CJKfamilydefault</code> 。 68
新增统一设置接口 <code>\ctexset</code> 。 76		正确设置 <code>up\TeX</code> 下字体命令。 67
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。 132
中文字号不再采用近似值。 106		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。 105
<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。 105
v2.0.1 (2015/05/15)		使用 <code>bootfont.bin</code> 判断 Windows XP 以避免权限问题。 113
General: 修复 10pt、11pt 等选项无效的问题。 41		随字体更新 <code>up\TeX</code> 的 <code>\xkanjiskip</code> 。 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> 。 83
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。 106		正确更新 <code>up\LaTeX</code> 的 <code>\CJKfamilydefault</code> 。 68
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。 82		<code>\chinese</code> : 支持 <code>\pagenumbering</code> 。 80
<code>nameformat</code> 可以接受章节名字为参数。 82		<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。 70		
给 <code>pdf\LaTeX</code> 下的非 UTF8 编码 <code>CJK</code> 字体族加上 <code>CMap</code> 。 42		

v2.4.2	(2016/05/15)	提供 \partmark。	84
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	49	新的标题格式选项 <code>break</code> 。	82
<code>\em</code> : 兼容 <code>upL^AT_EX 2016/05/07u00</code> 的定义。	67	v2.4.5	(2016/10/01 – 2016/10/25)
v2.4.3	(2016/06/03 – 2016/08/26)	General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	83
General: 更新 <code>unicode-math</code> 补丁。	49	新的标题格式选项 <code>hang</code> 。	82
简化 <code>fontspec</code> 补丁。	49	新的标题格式选项 <code>tocline</code> 。	82
确保 <code>\proofname</code> 非空。	80	<code>\ps@headings</code> : 修复补丁失败。	97
删除选项 <code>part/fixbefore skip</code> 和 <code>chapter/fixbefore skip</code> 。	83	v2.4.6	(2016/10/31 – 2016/11/20)
新的标题格式选项 <code>fixskip</code> 。	82	General: <code>\CTEXifname</code> 初始为假。	84
v2.4.4	(2016/09/09 – 2016/09/19)	支持字体属性可选项在后的新语法。	58
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	48	重新初始化 <code>\ifthechapter</code> 等。	96
改进 <code>hyperref</code> 宏包的标题锚点设置。	92	v2.4.7	(2016/12/23 – 2016/12/27)
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	44	General: 依赖 <code>pxeverysel</code> 宏包。	68
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	95	<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	97
提供 <code>\CTEXifname</code> 。	84		

代码索引

意大利体的数字表示描述对应索引项的页码; 带下划线的数字表示定义对应索引项的代码行号; 罗马字体的数字表示使用对应索引项的代码行号。

Symbols	
<code>_</code>	508, 3194, 3201, 3217, 3220
<code>\#</code>	474, 3095, 3154
<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>.../before skip</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>	1633
<code>\</code>	15, 33, 34, 102, 145, 147, 149, 205, 215, 672, 1096, 1178, 2272, 3571, 3572, 3685, 3686, 3906, 3936, 4964, 4965, 4966, 5040
<code>\{</code>	1107, 1108, 4966
<code>\}</code>	1107, 1108, 4966
<code>10pt</code>	8, 366
<code>11pt</code>	8, 366
<code>12pt</code>	8, 366
A	
<code>\abovedisplayshortskip</code>	3776, 3784, 3795, 3815, 3823, 3834
<code>\abovedisplay skip</code>	3775, 3778, 3783, 3790, 3794, 3801, 3814, 3817, 3822, 3829, 3833, 3840
<code>\abstractname</code>	2256
<code>abstractname</code>	12
<code>\addCJKfontfeature</code>	1178, 1222
<code>\addCJKfontfeatures</code>	1195
<code>\addcontentsline</code>	2480, 2526, 2530, 2535, 2563, 2567, 2572, 2661, 2666, 2672, 2678, 2801, 2815
<code>\AddEnumerateCounter</code>	2211, 2213, 2214, 2215
<code>\addpenalty</code>	2760, 3411, 3444, 3478, 3492, 3506
<code>\addtocontents</code>	2468, 2474, 2682, 2683
<code>\addvspace</code>	2469, 2475, 2487, 2490, 2682, 2683, 2760, 2764, 3232, 3242, 3250, 3260, 3268, 3278
<code>adobefonts</code>	29
<code>\algorithmname</code>	2263, 2279
<code>algorithmname</code>	12
<code>AlternateFont</code>	26, 1329
<code>\appendix</code>	2942
<code>appendix/name</code>	23, 2931
<code>appendix/number</code>	23, 2931
<code>appendix/numbering</code>	23, 2931
<code>\appendixname</code>	2258, 3556
<code>appendixname</code>	12
<code>\arabic</code>	3458

`\AtBeginDocument` 575, 1622
`\AtBeginDvi` 432
`\AtBeginShipoutFirst` 434, 1812, 1827
`\AtBeginUTFCommand` 716
`\AtEndOfClass` 135
`\AtEndOfPackage` 136
`\AtEndUTFCommand` 722
`autoindent` 10, 184, 2066

B

`\baselinestretch` 2121
`\begin` 3238, 3256, 3274
`\begingroup` 2792, 2828, 2860, 3228, 3246, 3264, 4464, 4497, 4504
`\belowdisplayshortskip` . 3777, 3785, 3796, 3816, 3824, 3835
`\belowdisplayskip` 3778, 3790, 3801, 3817, 3829, 3840
`\bfdefault` 961, 967, 1319, 1326
`\bfseries` 2542, 2545, 2581, 2584, 2597,
 2614, 2710, 2715, 2733, 3339, 3341, 3345, 3358, 3359,
 3362, 3387, 3389, 3395, 3413, 3417, 3438, 3472, 3489, 3503
`\bibname` 2261, 2264, 2272, 2280, 2286, 2287
`bibname` 12

bool commands:

`\bool_gset_false:N` 265, 270
`\bool_gset_true:N` 260
`\bool_if:NTF` 70, 1136, 2874, 3036, 3046, 3057, 3575
`\bool_if:nTF` 5068
`\bool_new:N` 131, 1173
`\bool_set_false:N` 1148
`\bool_set_true:N` 1166
`\c_false_bool` 2890, 2892
`\c_true_bool` 2889, 2891

box commands:

`\box_ht:N` 3856
`\box_new:N` 60
`\box_wd:N` 1898, 3150

C

`c5size` 29
`cap` 29
`captiondelimiter` 30, 3863
`\catcode` 4464, 4465, 4466, 4467, 4471, 4472
`CCTF` 29
`CCTfont` 29
`\ccwd` 25, 115, 188, 1894, 1944,
 1945, 1946, 2072, 2110, 2129, 2130, 2137, 2164, 2165,
 2166, 2169, 2171, 2172, 2183, 2194, 2195, 3873, 3874, 3875
`\centering` 2578, 2612, 3229,
 3247, 3265, 3345, 3357, 3362, 3366, 3395, 3417, 3425, 3460
`\chapter` 2638, 3581
`chapter/beforeskip` 31
`chapter/fixbeforeskip` 31
`chapter/lofskip` 21, 2409
`chapter/lotskip` 21, 2409
`chapter/numbering` 15
`chapter/pagestyle` 21, 2409
`\chaptermark` 2681
`\chaptername` 3384

char commands:

`\char_set_catcode:n` 73, 512, 513, 548, 549
`\char_set_catcode_active:N` 1354
`\char_set_catcode_comment:n` 507
`\char_set_catcode_escape:n` 508
`\char_set_catcode_letter:n` 67, 1625, 3168, 5032
`\char_set_catcode_math_superscript:n` 543
`\char_set_catcode_other:N` . 474, 894, 1337, 3095, 3154
`\char_set_catcode_other:n` 542, 5056
`\char_value_catcode:n` 73, 512, 513, 548, 549
`CharRange` 26, 1329
`\Chinese` 30, 2207
`\chinese` 25, 2205, 2215, 3332, 3394
`\CJK` 1863
`\CJKbold`
 3986, 3987, 3997, 3998, 4012, 4013, 4015, 4016, 4018,
 4019, 4023, 4024, 4026, 4027, 4029, 4030, 4042, 4043,
 4045, 4046, 4048, 4049, 4053, 4054, 4056, 4057, 4059, 4060
`\CJKfamily` 522, 526, 530, 534, 1033, 1055, 1263, 1267, 1271,
 1275, 1423, 1644, 4452, 4453, 4454, 4455, 4457, 4458, 4460
`\CJKfamilydefault`
 ... 516, 517, 534, 590, 1257, 1258, 1275, 1283, 1285,
 1289, 1296, 1309, 1640, 1649, 1702, 1706, 1714, 1727, 1729
`\CJKfilltwosides` 1871
`\CJKfontspec` 1195
`\CJKglue` 1897, 1911, 1928, 1941
`\CJKhook` 605
`\CJKnormal` 3979, 3980, 3981, 3982, 3983, 3984, 3985,
 3990, 3991, 3992, 3993, 3994, 3995, 3996, 4011, 4014,
 4017, 4022, 4025, 4028, 4041, 4044, 4047, 4052, 4055, 4058
`\CJKpunctmapfamily`
 . 638, 639, 640, 641, 642, 643, 647, 648, 649, 650, 651, 652
`\CJKrmdefault` 497,
 517, 518, 522, 1103, 1114, 1226, 1258, 1259, 1263, 1285,
 1288, 1289, 1637, 1640, 1641, 1710, 4108, 4110, 4111,
 4116, 4117, 4118, 4121, 4198, 4201, 4202, 4207, 4208,
 4209, 4276, 4279, 4280, 4286, 4287, 4288, 4353, 4356,
 4357, 4364, 4365, 4366, 4414, 4417, 4418, 4423, 4424, 4425
`\CJKsfdefault` 498, 519,
 526, 1104, 1115, 1233, 1260, 1267, 1638, 1642, 1711,
 4092, 4098, 4099, 4122, 4199, 4277, 4281, 4354, 4358, 4415
`\CJKtilde` 591
`\CJKttdefault` 499, 520, 530, 1105, 1116, 1239, 1261,
 1271, 1639, 1643, 1712, 4109, 4123, 4200, 4278, 4355, 4416
`clearalternatefont` 27, 1515
`\cleardoublepage` 2501, 2639, 3355, 3382
`\clearpage` 2503, 2639, 3355, 3382
`clist commands:`
`\clist_clear:N` 982, 4801
`\clist_concat:NNN` 1002
`\clist_const:Nn` 618
`\clist_gput_right:Nn` 382, 394, 404, 406
`\clist_gset:Nn` 1249
`\clist_if_empty:NTF` 1342, 1361, 1601
`\clist_map_break:n` 3015, 3566
`\clist_map_function:nN` 1518, 1520, 1552, 1566

- \clist_map_inline:Nn 608, 1454
- \clist_map_inline:nn
 - . 367, 630, 1559, 1855, 1881, 3010, 3563, 3703, 3738, 4790
- \clist_new:N 138, 999, 1250, 1396, 4819, 4959
- \clist_put_left:Nn 1003, 3632, 3658
- \clist_put_right:Nn 1165, 1442, 4813
- \clist_set:Nn 986, 1379, 1380, 4960
- \clist_use:Nn 4969
- \clubpenalty 2859, 2866
- \contentsname 2251
- contentsname 11
- continuation 12
- cs commands:
 - \cs:w 3303
 - \cs_end: 3303
 - \cs_generate_variant:Nn
 - 93, 450, 462, 946, 1067, 1081, 1154, 1174,
 - 1175, 1375, 1412, 1514, 1552, 1560, 5001, 5007, 5125, 5232
 - \cs_gset_eq:NN
 - 161, 162, 163, 261, 266, 271, 309, 311, 377,
 - 381, 482, 789, 793, 1141, 1144, 2979, 2980, 2985, 3638, 3665
 - \cs_gset_nopar:Npx 790
 - \cs_gset_protected_nopar:Npn 2943, 2991
 - \cs_gset_protected_nopar:Npx
 - 828, 1029, 1416, 1510, 2919
 - \cs_if_eq:NNTF 930, 2977, 5010, 5158, 5238
 - \cs_if_exist:NTF
 - 44, 358, 459, 739, 784, 796,
 - 826, 864, 1415, 1507, 1974, 3081, 3290, 3581, 3583, 5015
 - \cs_if_exist_p:N 5070, 5071
 - \cs_if_exist_use:N 1051, 1052
 - \cs_if_exist_use:NTF
 - 4, 767, 1073, 1529, 5004
 - \cs_if_free:NTF
 - 444, 454, 610, 625,
 - 1138, 1318, 2085, 2099, 2211, 3029, 3074, 3120, 3125, 3129
 - \cs_new:Npn
 - 110, 117, 124,
 - 885, 891, 895, 978, 1054, 2893, 2901, 2928, 3054, 5115, 5233
 - \cs_new_eq:NN
 - 6, 7, 8, 9, 20, 23, 95, 106, 107, 108, 133, 134,
 - 152, 359, 451, 595, 656, 805, 830, 906, 937, 1222, 1229,
 - 1723, 1876, 1877, 1878, 1890, 1941, 1952, 1958, 2206,
 - 2207, 2208, 2454, 2457, 2887, 2888, 2889, 2890, 2891,
 - 2892, 2942, 2967, 2968, 3308, 3579, 3584, 4795, 5162, 5234
 - \cs_new_nopar:Npn
 - 845, 1046, 1099, 1110, 2205, 2909, 3118, 3123, 4820
 - \cs_new_protected:Npn
 - 62,
 - 729, 979, 1181, 1340, 1431, 1445, 1596, 1947, 2402,
 - 2479, 3144, 3758, 3764, 4796, 4974, 4976, 4990, 4996,
 - 5016, 5021, 5026, 5037, 5041, 5046, 5051, 5061, 5066,
 - 5087, 5138, 5153, 5163, 5175, 5184, 5203, 5208, 5220, 5235
 - \cs_new_protected_nopar:Npn . 77, 79, 135, 136, 430,
 - 442, 452, 457, 463, 495, 503, 538, 555, 597, 603, 606,
 - 623, 628, 636, 645, 654, 765, 774, 787, 792, 794, 806,
 - 820, 847, 862, 875, 899, 908, 925, 939, 956, 1000, 1008,
 - 1027, 1047, 1057, 1082, 1120, 1132, 1134, 1156, 1158,
 - 1279, 1304, 1313, 1359, 1370, 1397, 1406, 1413, 1419,
 - 1439, 1452, 1471, 1485, 1492, 1504, 1524, 1540, 1558,
 - 1561, 1563, 1576, 1588, 1630, 1662, 1675, 1689, 1700,
 - 1732, 1750, 1765, 1781, 1792, 1796, 1833, 1840, 1844,
 - 1867, 1873, 1894, 1908, 1943, 1959, 1989, 2084, 2086,
 - 2111, 2146, 2156, 2161, 2315, 2345, 2438, 2445, 2452,
 - 2458, 2460, 2463, 2872, 2878, 2959, 2970, 3002, 3022,
 - 3027, 3032, 3042, 3052, 3096, 3286, 3295, 3300, 3676,
 - 3691, 3723, 3728, 3878, 3882, 4787, 4811, 4978, 4980, 5002
- \cs_new_protected_nopar:Npx ... 467, 1205, 1860, 3083
- \cs_set:Npn 1338, 1955, 2374
- \cs_set_eq:NN
 - 109, 189, 194, 199, 535,
 - 553, 573, 600, 604, 612, 626, 746, 799, 800, 907, 938,
 - 976, 1033, 1276, 1415, 1423, 1433, 1434, 1435, 1507,
 - 1508, 1509, 1633, 1705, 2222, 2227, 2230, 2235, 2459,
 - 2461, 2880, 2881, 2882, 2883, 2884, 2885, 2974, 3006, 3016
- \cs_set_nopar:Npn 1661, 3135
- \cs_set_nopar:Npx 1891, 3044, 4982
- \cs_set_protected:Npn 723, 732, 5064, 5130
- \cs_set_protected_nopar:Npn
 - 599, 747, 1911, 3076, 3133, 3765
- \cs_set_protected_nopar:Npx 1949
- \cs_to_str:N 53, 1204, 5062
- \cs_undefine:N . 1012, 1013, 1017, 1018, 1436, 1963, 5005
- cs4size 29
- \csname 2761, 2770, 2771, 2772, 2779, 2797, 2798, 2800, 2803,
- 2812, 2813, 2814, 2817, 4496, 4498, 4499, 4500, 4505, 4509
- \CTeX 25, 3859
- ctex commands:
 - \ctex_add_cmap:n 447, 452
 - \ctex_add_cmap:Nn 455, 457, 462
 - \ctex_add_to_selectfont:n 1947, 1980, 2008, 2098
 - \ctex_after_end_preamble:n 36, 109, 4974
 - \ctex_appto_cmd:NnnTF 478, 1624, 5046
 - \ctex_assign_heading_name:nn 2351, 2399, 2935
 - \ctex_at_end:n 135, 293, 356, 1941, 3945
 - \ctex_at_end_package:nn
 - 130, 433, 481, 737, 754, 1806, 1808, 1826, 2209, 2989,
 - 3062, 3069, 3128, 3130, 3131, 3165, 3205, 3314, 3861, 4990
 - \ctex_at_end_preamble:n 361, 580, 587, 1278, 1650, 4974
 - \ctex_auto_ignorespaces: 594, 604, 2018
 - \ctex_CJK_input:n 538
 - \ctex_declare_math_sizes:nnnn 3723, 3757
 - \ctex_default_pt:n 113, 117
 - \ctex_detected_platform: 1731, 3886, 3895
 - \ctex_family_cmap:nn 442, 451
 - \ctex_file_input:n
 - 77, 397,
 - 657, 2048, 3590, 3625, 3630, 3635, 3636, 3651, 3656,
 - 3662, 3663, 3767, 3769, 3887, 3891, 3899, 3945, 4070, 4071
 - \ctex_file_wrapper:nnn 62, 78, 81, 505, 540
 - \ctex_fix_varioref_label:n 3298, 3300
 - \ctex_fontset_error:n
 - ... 3878, 4177, 4211, 4245, 4290, 4322, 4325, 4368, 4427
 - \ctex_gadd_hook:Nn 605, 4994, 4996
 - \ctex_get_macro_meaning:N 5133
 - \ctex_get_macro_meaning:NTF 5129, 5155, 5211
 - \ctex_hypersetup:n
 - 70, 359, 1789, 1801, 1804, 1819, 1822, 1825
 - \ctex_if_autoindent_touched:TF 133, 189, 194, 199, 3617

- \ctex_if_ccglue_touched:TF [1925](#), [2117](#)
- \ctex_if_ccglue_touched_p: [1925](#)
- \ctex_if_macosx:TF [1772](#), [1773](#), [1775](#), [1781](#)
- \ctex_if_preamble:TF [108](#), [3914](#)
- \ctex_if_rescanable:NnTF [5153](#), [5187](#)
- \ctex_ignorespaces_case:N [597](#), [2015](#), [2018](#), [2021](#)
- \ctex_load_fontset: [3882](#), [3922](#), [3944](#)
- \ctex_load_zhmap:nnnn [495](#), [4114](#), [4205](#), [4284](#), [4362](#), [4421](#)
- \ctex_ltj_add_font_features:n [1156](#), [1219](#)
- \ctex_ltj_add_font_features:nn [1156](#), [1157](#)
- \ctex_ltj_char_range_key:nn [1593](#), [1596](#)
- \ctex_ltj_clear_alternate_font:n [1518](#), [1524](#)
- \ctex_ltj_declare_alternate_shape:nnnnnn [1425](#), [1445](#)
- \ctex_ltj_declare_char_range:n [1555](#), [1558](#), [1560](#)
- \ctex_ltj_declare_char_range:nn [1562](#), [1563](#)
- \ctex_ltj_def_char_range_key:n [1568](#), [1588](#)
- \ctex_ltj_ensure_default_family: [1279](#), [1719](#)
- \ctex_ltj_extract_font: [799](#), [806](#)
- \ctex_ltj_family_if_exist:nN [1068](#)
- \ctex_ltj_family_if_exist:nNTF
..... [1059](#), [1283](#), [1288](#), [1306](#), [1309](#), [1527](#)
- \ctex_ltj_family_if_exist:nTF [1068](#)
- \l_ctex_ltj_family_tl
[1061](#), [1066](#), [1123](#), [1124](#), [1139](#), [1143](#), [1146](#), [1157](#), [1521](#), [1522](#)
- \ctex_ltj_fontspec:nn [1120](#), [1132](#), [1167](#), [1214](#)
- \ctex_ltj_get_and_define_fonts:nN [899](#)
- \ctex_ltj_get_and_define_fonts_al:nN [903](#), [906](#)
- \ctex_ltj_get_and_define_fonts_ja:nN [902](#), [908](#)
- \ctex_ltj_if_alternate_shape_exist:nTF
..... [809](#), [822](#), [840](#), [849](#)
- \ctex_ltj_if_jfont:nTF [54](#), [885](#), [896](#), [901](#)
- \ctex_ltj_if_jfont_math:NTF [891](#), [932](#)
- \ctex_ltj_math_group_hook: [54](#), [747](#), [931](#), [937](#)
- \ctex_ltj_pickup_font: ... [780](#), [794](#), [869](#), [911](#), [913](#), [915](#)
- \ctex_ltj_reset_alternate_font:n [1520](#), [1540](#)
- \ctex_ltj_save_alternate_family:Nnnn [1390](#), [1410](#), [1413](#)
- \ctex_ltj_save_alternate_seq:Nn [1365](#), [1370](#)
- \ctex_ltj_save_alternate_seq:Nnnwn [1370](#)
- \ctex_ltj_save_alternate_seq:Nnnwnw [1373](#), [1376](#)
- \ctex_ltj_save_char_range:n [1566](#), [1573](#)
- \ctex_ltj_save_char_range:nn [1575](#), [1576](#)
- \ctex_ltj_select_alternate_font: [847](#), [1983](#)
- \ctex_ltj_select_font: [765](#), [1982](#)
- \ctex_ltj_set_alternate_family:nn [1368](#), [1397](#)
- \ctex_ltj_set_alternate_family:nnn . [1409](#), [1417](#), [1419](#)
- \ctex_ltj_set_alternate_family:Nnnnn [1401](#), [1406](#), [1412](#)
- \ctex_ltj_set_alternate_family:nnTF [987](#), [1359](#)
- \ctex_ltj_set_alternate_seq:n [1331](#), [1336](#)
- \ctex_ltj_set_alternate_shape:n ... [1473](#), [1485](#), [1496](#)
- \ctex_ltj_set_alternate_shape:nnN [1458](#), [1492](#)
- \ctex_ltj_set_alternate_shape:nnn [1461](#), [1468](#)
- \ctex_ltj_set_alternate_shape:nnnn [1470](#), [1471](#)
- \ctex_ltj_set_alternate_shape:Nnnnnnn .. [1448](#), [1452](#)
- \ctex_ltj_set_family:nnn
..... [979](#), [1151](#), [1198](#), [1208](#), [1226](#), [1233](#), [1239](#), [1245](#)
- \ctex_ltj_set_math_letter:NN [729](#), [741](#), [749](#)
- \ctex_ltj_subst_font: [800](#), [820](#)
- \ctex_ltj_swap_cs:NN [1424](#), [1431](#)
- \ctex_ltj_switch_family:n
..... [1056](#), [1057](#), [1067](#), [1124](#), [1152](#), [1206](#)
- \ctex_ltj_update_mathfont: [1301](#), [1304](#)
- \ctex_ltj_update_mathfont:n [1307](#), [1310](#), [1313](#)
- \ctex_ltj_use_math_group:Nn [746](#), [925](#)
- \ctex_make_spa:nn [4787](#)
- \ctex_mono_jfm:n [939](#), [947](#), [2038](#)
- \ctex_package_end_hook:n [5002](#), [5011](#)
- \ctex_parse_name:NN [1966](#), [5057](#), [5061](#)
- \ctex_parse_name:NNn [5062](#), [5066](#), [5125](#)
- \ctex_patch_boot:NnnnTF
..... [5018](#), [5023](#), [5028](#), [5043](#), [5048](#), [5051](#)
- \ctex_patch_cmd:Nnn
..... [1657](#), [3098](#), [3104](#), [3110](#), [3177](#), [3180](#), [3184](#),
[3190](#), [3200](#), [3208](#), [3211](#), [3216](#), [3219](#), [3873](#), [3874](#), [3875](#), [5026](#)
- \ctex_patch_cmd_all:NnnnTF [5021](#)
- \ctex_patch_cmd_once:NnnnTF [1651](#), [5016](#)
- \ctex_patch_failure:N [484](#), [524](#), [528](#), [532](#), [536](#),
[1265](#), [1269](#), [1273](#), [1277](#), [1628](#), [1971](#), [3161](#), [3171](#), [5035](#), [5037](#)
- \ctex_plane_to_utfxvibe:Nn [554](#)
- \ctex_preto_cmd:NnnTF [521](#), [525](#),
[529](#), [533](#), [1262](#), [1266](#), [1270](#), [1274](#), [1968](#), [3158](#), [3167](#), [5041](#)
- \ctex_punct_map_bfseries:nn
..... [628](#), [4099](#), [4111](#), [4117](#), [4202](#),
[4208](#), [4280](#), [4281](#), [4287](#), [4357](#), [4358](#), [4359](#), [4365](#), [4418](#), [4424](#)
- \ctex_punct_map_family:nn [623](#), [4092](#), [4098](#),
[4108](#), [4109](#), [4116](#), [4198](#), [4199](#), [4200](#), [4207](#), [4276](#), [4277](#),
[4278](#), [4286](#), [4353](#), [4354](#), [4355](#), [4364](#), [4414](#), [4415](#), [4416](#), [4423](#)
- \ctex_punct_map_itshape:nn [645](#),
[4110](#), [4118](#), [4201](#), [4209](#), [4279](#), [4288](#), [4356](#), [4366](#), [4417](#), [4425](#)
- \ctex_punct_map_series:nnn [632](#), [633](#), [636](#)
- \ctex_punct_set:n [606](#),
[4087](#), [4115](#), [4197](#), [4206](#), [4275](#), [4285](#), [4352](#), [4363](#), [4413](#), [4422](#)
- \ctex_punct_space:nn [654](#)
- \ctex_scheme_input:n [79](#), [3316](#), [3592](#)
- \ctex_select_size: .. [2073](#), [2080](#), [2083](#), [2084](#), [2105](#), [2159](#)
- \ctex_set_default_ccwd:Nn [110](#), [198](#), [2079](#), [2104](#)
- \ctex_set_font_size:Nnn [3758](#), [3773](#),
[3781](#), [3792](#), [3803](#), [3804](#), [3805](#), [3806](#), [3807](#), [3808](#), [3809](#),
[3812](#), [3820](#), [3831](#), [3842](#), [3843](#), [3844](#), [3845](#), [3846](#), [3847](#), [3848](#)
- \ctex_set_ignorespaces: [597](#), [605](#)
- \ctex_set_upfamily:nnn
..... [1662](#), [4134](#), [4135](#), [4136](#), [4137](#), [4138](#),
[4139](#), [4141](#), [4221](#), [4222](#), [4223](#), [4224](#), [4300](#), [4301](#), [4302](#),
[4303](#), [4377](#), [4378](#), [4379](#), [4380](#), [4381](#), [4383](#), [4436](#), [4437](#), [4438](#)
- \ctex_set_upfonts:nnnnnn
..... [1689](#), [4129](#), [4217](#), [4296](#), [4373](#), [4432](#)
- \ctex_set_upmap:nnn
... [1675](#), [1691](#), [1692](#), [1693](#), [1694](#), [4140](#), [4142](#), [4382](#), [4384](#)
- \ctex_set_zhmap:n [429](#), [500](#), [1677](#)
- \ctex_titleps_hook: [3094](#), [3129](#), [3130](#)
- \ctex_titlesec_hook: [3002](#), [3066](#)
- \ctex_update_ccglue: [1908](#), [2124](#), [2143](#), [2187](#)
- \ctex_update_ccwd: [1894](#), [2118](#), [2195](#)

- \ctex_update_default_family: 46, 580, 1278, 1650, 1699
- \ctex_update_em_unit: 1943, 2113, 2163
- \ctex_update_parindent: 2091, 2146, 2188
- \ctex_update_size: 2085, 2086
- \ctex_update_stretch: 2090, 2111
- \ctex_update_xkanjiskip: 1988, 2008
- \ctex_update_ziju: 2093, 2161
- \ctex_varioref_hook: 3295, 3306, 3314
- \ctex_zhmap_case:nnn
..... 134, 261, 266, 271, 4085, 4179, 4247, 4340, 4404
- \ctex_zhmap_input:n 500, 503
- \ctex_zihao:n 3675, 3676
- \ctex_ziju:n 2155, 2156
- ctex internal commands:
- \g_ctex_after_end_preamble_hook_tl 4977, 4981, 4989
- __ctex_assign_heading_name:nnn 2399
- \l_ctex_autoindent_tl
..... 132, 188, 193, 198, 2072, 2076, 2079, 2148, 2151
- __ctex_calc_bounds:nn 4815, 4816, 4820
- __ctex_ccglue: 1928, 1941
- \l_ctex_ccglue_skip 1912, 1916,
1919, 1922, 1924, 1932, 1936, 2122, 2133, 2142, 2178, 2186
- \c_ctex_class_tl
.. 84, 86, 409, 414, 419, 424, 3579, 3585, 3587, 3588, 3590
- __ctex_clear_fntef_color:n ... 1833, 1840, 1844, 1859
- \c_ctex_cmap_encoding_seq 439, 446
- \l_ctex_ctexcap_options_clist 4959, 4960, 4969
- __ctex_def_heading_keys:n 2345, 2429
- __ctex_default_pt:w 119, 124
- \l_ctex_encoding_tl 128, 221,
222, 488, 491, 582, 590, 1802, 2044, 2203, 3620, 3647, 3661
- \g_ctex_end_preamble_hook_tl 4975, 4979, 4988
- \c_ctex_engine_file_str 43, 2048
- \c_ctex_engine_str 43
- \l_ctex_family_default_init_tl 1702, 1722
- __ctex_family_default_wrap:n 1705, 1723, 1726
- __ctex_fix_varioref_label:n 3300
- \g_ctex_font_size_flag 151,
161, 162, 163, 377, 381, 403, 3637, 3638, 3664, 3665, 3766
- \c_ctex_font_size_prop 3678, 3689, 3733, 3760
- \c_ctex_font_size_seq 3687, 3690, 3701
- \l_ctex_font_size_tl 3678, 3679, 3725, 3726, 3760, 3761
- \g_ctex_fontset_tl 225, 1734, 1752, 1777, 1784, 1785,
3884, 3887, 3890, 3891, 3897, 3899, 3919, 3921, 3926, 3943
- __ctex_get_font_sizes:Nn 3725, 3728
- __ctex_get_macro_meaning:w 5129
- \l_ctex_heading_bool 304, 3575
- \l_ctex_heading_skip 2442,
2449, 2451, 2453, 2454, 2465, 2466, 2469, 2471, 2472, 2475
- \c_ctex_headings_cs_seq 2961, 2975, 2984
- \c_ctex_headings_seq ... 2305, 2426, 3079, 3142, 3297
- __ctex_hookto_cmd:Nnnnw 5043, 5048, 5208
- __ctex_hookto_cmd_parameter:Nnnnw 5215, 5235
- __ctex_hookto_cmd_parameterless:Nnnnw . 5214, 5220
- __ctex_initial_heading:n 2315, 2428
- __ctex_left_hook_aux:nn 5233
- \l_ctex_line_spread_fp
..... 109, 181, 3639, 3640, 3666, 3667, 3852, 3854
- \l_ctex_line_stretch_tl 2104, 2109, 2110, 2114
- \l_ctex_ltj_add_alternate_bool 1136, 1148, 1166, 1173
- __ctex_ltj_alternate_cs:n
..... 1013, 1016, 1017, 1018, 1051, 1052, 1054, 1139,
1142, 1143, 1145, 1146, 1365, 1402, 1464, 1489, 1529, 1546
- \l_ctex_ltj_alternate_family_tl 1427, 1430
- \l_ctex_ltj_alternate_options_clist
..... 1380, 1381, 1392, 1396
- \l_ctex_ltj_alternate_seq
..... 983, 1349, 1350, 1358, 1364, 1372, 1620
- \l_ctex_ltj_base_CJKfamily_tl
..... 984, 998, 1049, 1464, 1489
- \l_ctex_ltj_base_family_tl 64,
1014, 1044, 1050, 1408, 1449, 1508, 1527, 1532, 1533, 1545
- __ctex_ltj_change_encoding: 948, 1032, 1422
- \l_ctex_ltj_char_range_clist 982,
1333, 1342, 1361, 1379, 1391, 1403, 1426, 1448, 1599, 1601
- __ctex_ltj_char_range_parse_feature:w . 1604, 1608
- \g_ctex_ltj_char_range_prop 1456, 1563
- \l_ctex_ltj_char_range_tl
..... 1456, 1459, 1565, 1567, 1569, 1571, 1578
- __ctex_ltj_check_family:n 1008, 1363
- \l_ctex_ltj_current_font_tl
..... 50, 767, 771, 772, 784, 826, 831
- \l_ctex_ltj_current_shape_tl 849, 854, 859, 860
- __ctex_ltj_declare_char_range:nn 1559, 1561
- \g_ctex_ltj_default_features_clist 1002, 1249, 1250
- __ctex_ltj_family_csname:n ... 1012, 1029, 1046, 1073
- \g_ctex_ltj_family_font_name_prop
..... 989, 1005, 1010, 1084, 1160, 1281, 1293
- \g_ctex_ltj_family_font_options_prop 990, 1005, 1163
- \g_ctex_ltj_family_int . 1126, 1128, 1441, 1442, 1444
- \g_ctex_ltj_family_name_prop
..... 1005, 1014, 1036, 1070, 1408
- __ctex_ltj_family_unknown_warning:n 1064, 1082, 1537
- \l_ctex_ltj_font_options_clist 986,
991, 992, 993, 999, 1035, 1164, 1165, 1168, 1399, 1400, 1404
- __ctex_ltj_fontspec:nnn 1127, 1134, 1154
- \g_ctex_ltj_fontspec_family_tl
..... 1034, 1037, 1038, 1039, 1045, 1050, 1075
- \g_ctex_ltj_fontspec_prop 1122, 1150, 1155
- __ctex_ltj_gset_family_cs:nn 994, 1027
- __ctex_ltj_if_jfont_math:w 892, 895
- \l_ctex_ltj_jfm_tl 939, 960, 962, 1003
- \c_ctex_ltj_math_fam_int 968, 969
- \c_ctex_ltj_math_family_tl
..... 1315, 1317, 1319, 1322, 1326
- \c_ctex_ltj_math_tl
..... 963, 964, 966, 968, 1245, 1306, 1316, 1321, 1325
- __ctex_ltj_msg_def_family_map:n 1097, 1099
- __ctex_ltj_msg_family_map:n 1026, 1096, 1110
- __ctex_ltj_noembed_wrap:n 976, 978
- __ctex_ltj_pass_args:nnnn
..... 1181, 1197, 1207, 1213, 1225, 1232, 1238, 1244

<code>__ctex_ltj_patch_external_font:w</code>	812, 845	<code>\g__ctex_std_options_clist</code>	
<code>__ctex_ltj_pop_fontname:</code>	783, 792, 871	137, 382, 394, 404, 406, 410, 415, 420, 425
<code>__ctex_ltj_post_arg:w</code>	1184, 1190	<code>__ctex_titlesec_format:n</code>	3016, 3024, 3027
<code>__ctex_ltj_push_fontname:n</code>	779, 787, 868	<code>__ctex_titlesec_format:Nn</code>	3006, 3030, 3032
<code>\g__ctex_ltj_reset_alterate_prop</code>		<code>__ctex_titlesec_format_explicit:Nn</code>	3007, 3042
.....	1019, 1531, 1543, 1551	<code>__ctex_titlesec_hook:n</code>	3016, 3020, 3022
<code>__ctex_ltj_save_alterate_shape:Nnn</code>	1463, 1488, 1504	<code>__ctex_titlesec_spacing:Nn</code>	3025, 3052
<code>__ctex_ltj_save_fontname:w</code>	789, 793	<code>__ctex_titlesec_spacing:nnnnn</code>	3053, 3054
<code>__ctex_ltj_select_font_aux:</code>	768, 774	<code>__ctex_tmp:N</code>	1955, 1966
<code>__ctex_ltj_set_alterate_family:n</code>	1040, 1047	<code>__ctex_tmp:w</code>	5064, 5119, 5130, 5151
<code>__ctex_ltj_tmp:w</code>	1338, 1355, 1433, 1435, 1436	<code>\l__ctex_tmp_box</code>	58, 1897, 1898, 3146, 3150
<code>\l__ctex_ltj_tmp_seq</code>	1346, 1347, 1350, 1357	<code>\l__ctex_tmp_dim</code>	58,
<code>\l__ctex_ltj_tmp_tl</code>		2114, 2115, 2120, 2123, 2129, 2168, 2170, 2172, 2181, 2183	
.....	985, 986, 1010, 1021, 1024, 1161, 1169,	<code>\l__ctex_tmp_int</code>	58, 558,
1203, 1206, 1208, 1283, 1288, 1306, 1307, 1309, 1310,		560, 561, 563, 567, 569, 2128, 2130, 2131, 2137, 2138,	
1344, 1345, 1346, 1383, 1385, 1386, 1393, 1603, 1611, 1620		2170, 2171, 2172, 2175, 2176, 2181, 2183, 2915, 2918, 2922	
<code>\g__ctex_ltj_unknown_family_seq</code> ...	1086, 1088, 1093	<code>\l__ctex_tmp_tl</code>	
<code>__ctex_ltj_update_family_uid:N</code>	992, 1399, 1438	58, 366, 373, 388, 389, 465, 466, 557, 559,
<code>__ctex_ltj_use_global_options:N</code>	993, 1000, 1381, 1400	1957, 1968, 2347, 2411, 2424, 2434, 3644, 3645, 3733, 3734	
<code>\c__ctex_macosx_file_str</code>	1783, 1787	<code>\c__ctex_upper_case_file_str</code>	1771, 1779
<code>\l__ctex_msyh_suffix_tl</code>		<code>\c__ctex_version_tl</code>	3, 6, 7, 8, 9, 20, 23
.....	4074, 4075, 4079, 4095, 4097, 4131	<code>__ctex_write_family:nn</code>	4791, 4796
<code>\l__ctex_parameter_str</code>	5126, 5145, 5166, 5213	<code>\l__ctex_xkanjiskip_skip</code>	1988
<code>\l__ctex_parameter_tl</code>	5166, 5171, 5182	<code>\l__ctex_xkanjiskip_tl</code>	1995, 2000, 2001
<code>__ctex_parse_name:nnNnNn</code>	5078, 5087	<code>\c__ctex_zero_tl</code>	2088, 2095, 2097
<code>__ctex_parse_name:w</code>	5100, 5106, 5115	<code>\g__ctex_zhmCJK_bool</code>	131, 260, 265, 270, 487, 581
<code>__ctex_patch_cmd:Nnnnnw</code>	5018, 5023, 5028, 5184	<code>\l__ctex_ziju_dim</code> 2164, 2165, 2169, 2180, 2182, 2186, 2190	
<code>__ctex_patch_false:w</code> 5054, 5060, 5199, 5201, 5218, 5244		<code>\l__ctex_ziju_tl</code>	2088, 2096, 2097, 2158, 2164
<code>__ctex_patch_rebuild:Nn</code> 5157, 5163, 5196, 5237, 5241		<code>\ctexbibname</code>	2273, 2293
<code>__ctex_patch_replace:nnTF</code>	5191, 5203	<code>\CTEXcounter</code>	2208
<code>__ctex_patch_rescan:Nn</code>	5165, 5166, 5167, 5175	<code>\CTEXdigits</code>	25, 2218
<code>__ctex_patch_true:w</code>	5053, 5059, 5197, 5230, 5242	<code>\CTEXfilltwosides</code>	30, 1867, 1877
<code>\l__ctex_prefix_str</code>	5126, 5144, 5165, 5222, 5223	<code>\CTEXifname</code>	16, 2457, 2713, 2895, 2903, 2911,
<code>\l__ctex_prefix_tl</code>	5165, 5170, 5181	3137, 3179, 3182, 3186, 3198, 3202, 3210, 3213, 3218, 3221	
<code>\l__ctex_punct_bounds_clist</code> ...	4801, 4807, 4813, 4819	<code>\CTEXindent</code>	30, 2191
<code>\c__ctex_punct_family_clist</code>	608, 618	<code>\CTEXnoindent</code>	30, 2191
<code>\l__ctex_punct_font</code>	4799, 4800	<code>\CTEXnumber</code>	25, 2218
<code>\c__ctex_punct_seq</code>	4778, 4779, 4802	<code>\CTEXnumberline</code>	2909, 2929, 3391
<code>\l__ctex_punct_tl</code>		<code>\CTEXoptions</code>	30, 2052
.....	280, 577, 578, 664, 947, 2035, 2036, 2037, 2038	<code>\ctexset</code>	5, 2049, 2056, 2062
<code>__ctex_rebuild_cmd:w</code> ...	5157, 5158, 5162, 5237, 5238	<code>\CTEXsetfont</code>	30, 2083
<code>\l__ctex_replacement_str</code>		<code>\CTEXsetup</code>	30, 2052
.....	5126, 5146, 5167, 5205, 5206, 5240	<code>\CTEXsout</code>	30
<code>\l__ctex_replacement_tl</code>	5167, 5172, 5183	<code>\ctexspadef</code>	654, 4806
<code>__ctex_restore_selectfont:</code>	1959, 1963, 1970	<code>\CTEXthechapter</code>	16, 2660, 2904, 3072, 3198, 3218
<code>__ctex_right_hook_aux:nn</code>	5234	<code>\CTEXtheparagraph</code>	16
<code>__ctex_save_bounds:n</code>	4803, 4811	<code>\CTEXthepart</code>	16, 2896, 3077
<code>__ctex_save_cmap:Nn</code>	459, 463	<code>\CTEXthesection</code>	16, 3179, 3186, 3202, 3210, 3221
<code>__ctex_save_font_size:nn</code>	3689	<code>\CTEXthesubparagraph</code>	16
<code>\l__ctex_scheme_tl</code>	315, 3316, 3592	<code>\CTEXthesubsection</code>	16, 3182, 3213
<code>\g__ctex_section_depth_flag</code>		<code>\CTEXthesubsubsection</code>	16
.....	94, 309, 311, 3514, 3530, 3546	<code>\CTEXunderdblline</code>	30
<code>\c__ctex_section_headings_seq</code> .	2300, 2307, 2916, 3020	<code>\CTEXunderdot</code>	30
<code>__ctex_set_font_size:nnNn</code>	3761, 3764	<code>\CTEXunderline</code>	30
<code>\g__ctex_spa_iow</code>	4789, 4792, 4794, 4804	<code>\CTEXunderwave</code>	30
<code>\c__ctex_std_class_tl</code> ...	3563, 3578, 3579, 3645, 3668	<code>\CTEXxout</code>	30

\CurrentOption 394

D

declarecharrange 26, 1553

\DeclareFontEncoding 949

\DeclareFontFamily
..... 473, 958, 3963, 3964, 3965, 3966, 3967, 3968

\DeclareFontShape
..... 959, 961, 1666, 1667, 1670, 1671, 3979, 3980,
3981, 3982, 3983, 3984, 3985, 3986, 3987, 3990, 3991,
3992, 3993, 3994, 3995, 3996, 3997, 3998, 4002, 4003,
4004, 4005, 4006, 4007, 4011, 4012, 4013, 4014, 4015,
4016, 4017, 4018, 4019, 4022, 4023, 4024, 4025, 4026,
4027, 4028, 4029, 4030, 4034, 4035, 4036, 4037, 4041,
4042, 4043, 4044, 4045, 4046, 4047, 4048, 4049, 4052,
4053, 4054, 4055, 4056, 4057, 4058, 4059, 4060, 4064, 4065

\DeclareFontSubstitution 952

\DeclareKanjifamily
..... 1664, 1665, 3970, 3971, 3972, 3973, 3974, 3975

\DeclareMathSizes 3726

\DeclareSymbolFont 964, 1316

\def 2521, 2558, 2590, 2608,
2620, 2651, 2691, 2700, 2724, 2743, 2769, 2773, 2808,
2823, 2837, 2838, 2842, 3786, 3797, 3825, 3836, 4470, 4506

\defaultCJKfontfeatures 1223

\defbeamertemplate 3227, 3245, 3263

\defbeamertemplatealias 3281, 3282, 3283

dim commands:

\dim_abs:n 2182
\dim_add:Nn 2165
\dim_compare:nNnTF ... 112, 2115, 2150, 2166, 2173, 2441
\dim_eval:n 2135, 2181
\dim_max:nn 3149
\dim_min:nn 2182
\dim_new:N 61, 1907, 2190
\dim_set:Nn 1898, 1944, 1945,
1946, 2114, 2120, 2151, 2164, 2168, 2195, 2448, 3147, 3856
\dim_sub:Nn 2172
\dim_to_decimal:n 36, 3697, 3698, 3735
\dim_to_decimal_in_unit:nn 4826
\dim_use:N 120, 3856
\dim_zero:N 2201
\c_max_dim 2115, 2441
\c_zero_dim 2123, 2135, 2150, 2166, 2173, 2393, 3547
\dimexpr 2540, 2595, 2706, 2730, 3233, 3251, 3269
\directlua 702, 703
\document 4982, 4985

E

\edef 4489, 4490

\else . 1653, 1654, 1658, 1659, 2493, 2502, 2511, 2527, 2532,
2564, 2569, 2639, 2647, 2663, 2669, 2675, 2687, 2696,
2754, 2759, 2778, 2783, 2802, 2807, 2816, 2836, 2852,
2865, 3355, 3382, 4501, 4502, 4548, 4587, 4618, 4677, 4740

else commands:

\else: 490, 843, 889, 1929, 2983, 3183, 3860
\em 1651

\eminnershape 1653, 1654, 1659, 1661

\end 3241, 3259, 3277, 4871

\endCJKfilltwosides 1873

\endcsname 2761,
2770, 2771, 2772, 2779, 2797, 2798, 2800, 2803, 2812,
2813, 2814, 2817, 4496, 4498, 4499, 4500, 4503, 4505, 4509

\endCTEXfilltwosides 1873, 1878

\endgroup 2799, 2835, 2860, 3243, 3261, 3279, 4489, 4497, 4512

\endinput 4494

\endlinechar 4468, 4469

etex commands:

\etex_dimexpr:D 120, 2129
\etex_glueexpr:D 2121
\everypar 2758, 2855, 2867
\EverySelectfont 1974, 1977

exp commands:

\exp_after:wN
119, 796, 812, 814, 889, 892, 3053, 3679, 3726, 3761, 5135
\exp_args:cc 3291
\exp_args:Nc 3025, 3030, 5089
\exp_args:Nco 3135
\exp_args:Nnc 99
\exp_args:NNf 1611
\exp_args:NNNo 3855
\exp_args:Nno 3303
\exp_args:No 559, 1780, 3049, 4803, 5151
\exp_args:Nx 293, 587, 661, 2155, 3675, 3854
\exp_last_unbraced:Nn 5100
\exp_last_unbraced:NNn 1967
\exp_not:N 124, 125, 469, 589,
590, 591, 758, 759, 761, 831, 867, 952, 1036, 1037, 1038,
1039, 1390, 1511, 1710, 1711, 1712, 1726, 1862, 1863,
1864, 1891, 2321, 2322, 2323, 2328, 2330, 2331, 2332,
2333, 2336, 2337, 2340, 2352, 2353, 2354, 2355, 2356,
2357, 2358, 2359, 2360, 2361, 2362, 2364, 2365, 2366,
2368, 2369, 2371, 2374, 2379, 2381, 2383, 2385, 2387,
2389, 2921, 2923, 2924, 2925, 2926, 3037, 3038, 3047,
3048, 3085, 3087, 3088, 3695, 3897, 5080, 5081, 5082, 5170
\exp_not:n 69,
388, 760, 762, 1033, 1034, 1035, 1168, 1169, 1391,
1392, 1393, 1417, 1511, 1705, 1727, 1950, 2375, 2434,
3056, 3292, 4961, 4985, 5098, 5170, 5171, 5172, 5227, 5228
\exp_stop_f: 726
\expandafter
4471, 4472, 4496, 4497, 4498, 4499, 4500, 4505, 4509, 4510

\ExplSyntaxOff 66, 72, 478, 521, 525, 529, 533, 684,
1262, 1266, 1270, 1274, 1652, 1968, 3158, 3226, 4832, 5031

\ExplSyntaxOn 71, 713, 3284, 4771

F

\familydefault 1708

fancyhdr 29, 341

\fangsong 4454

\fi .. 2484, 2489, 2495, 2504, 2513, 2516, 2531, 2536, 2544,
2553, 2568, 2573, 2583, 2603, 2623, 2631, 2632, 2635,
2639, 2649, 2667, 2673, 2679, 2690, 2699, 2703, 2711,

2712, 2721, 2727, 2739, 2744, 2750, 2756, 2763, 2765,
2787, 2788, 2804, 2818, 2821, 2840, 2849, 2868, 2869,
3178, 3181, 3185, 3195, 3196, 3201, 3209, 3212, 3217,
3220, 3237, 3255, 3273, 3355, 3382, 4501, 4502, 4511, 4764

fi commands:

\fi: 407, 537,
574, 593, 843, 889, 1929, 1972, 2986, 3187, 3203, 3770, 3860
\figurename 2254
figurename 11

file commands:

\g_file_current_name_tl 676, 1780
\file_if_exist:nTF
..... 48, 1767, 1769, 1771, 1783, 3890, 4069, 4076, 4078
\file_if_exist_input:nTF 86
\file_input:n 78, 87, 89, 510, 546
fntef 30, 341
\font 725, 3963, 3964, 3965, 3966, 3967, 3968

font commands:

\font_glyph_if_exist:NnTF 725
fontset 7, 225, 3910
\fontsize 3679

fontspec commands:

\g_fontspec_encoding_tl 957
\l_fontspec_family_tl 1450
\fontspec_set_family:Nnn 1034, 1427
\fontspec_visible_space: 723
\fontspec_visible_space_fallback: 727

fontspec internal commands:

__fontspec_namewrap:n 55, 976
\footnotesep 3855
\footnotesize 3792, 3831, 3855

fp commands:

\fp_compare:nNnTF 3639, 3666, 3852
\fp_eval:n 2158, 4822
\fp_set:Nn 3640, 3667
\fp_use:N 2095, 3854
\c_nan_fp 182
\c_zero_fp 2095, 3639, 3666, 3852

G

GBK 6, 221
\gdef 2947, 2948, 2949, 2950, 2953, 2954, 2955, 2956
\global 2643, 2854, 2857
\globaljfont 814
\glueexpr 2794, 2809,
2830, 2837, 2838, 2864, 3232, 3242, 3250, 3260, 3268, 3278

group commands:

\group_begin: 473,
718, 731, 776, 798, 824, 866, 893, 981, 1031, 1336,
1421, 1506, 1526, 1542, 1704, 1954, 2327, 2335, 2409,
2972, 3094, 3153, 3855, 4798, 4979, 5055, 5063, 5074, 5129
\group_end: 477, 486,
722, 735, 781, 802, 837, 870, 897, 996, 1041, 1356, 1428,
1512, 1538, 1549, 1716, 1967, 2338, 2342, 2433, 2987,
3117, 3157, 3855, 4809, 4979, 5077, 5124, 5152, 5186, 5210
\group_insert_after:N 834, 835

\gtfamily 1653, 1658

H

hbox commands:

\hbox_set:Nn 1897, 3146
heading 9, 304
\heiti 4453
\hskip ... 2794, 2809, 2830, 2837, 2838, 2863, 2864, 3209, 3212
\hspace 2526, 2563, 2896, 2904, 3077
\Huge 2584, 2614, 2715, 2733, 3359, 3389, 3809, 3848
\huge 2545,
2581, 2597, 2710, 3341, 3358, 3362, 3387, 3395, 3808, 3847
hyperref 30, 341
\hypersetup 358, 359, 1793
\hyphenchar 3963, 3964, 3965, 3966, 3967, 3968

I

if commands:

\if_bool:N 487, 581
\if_case:w 403, 3766
\if_cs_exist:N 554, 1953
\if_false: 107
\if_meaning:w 1928
\if_true: 106
\iftexpdf 105
\ifdim 2748, 2790, 2826, 2844
\ifincsname 3860
\IfNoValueTF
... 1183, 1204, 1475, 1580, 1613, 1615, 2057, 2063, 2405
\ifnum 2522,
2541, 2559, 2580, 2652, 2708, 2774, 2802, 2816, 3178,
3181, 3185, 3192, 3193, 3201, 3209, 3212, 3217, 3220, 4501
\ifodd 2489, 2491, 2516, 2523, 2541, 2553, 2560,
2580, 2603, 2623, 2645, 2656, 2703, 2721, 2727, 2739,
2752, 2763, 2779, 2791, 2827, 2845, 2849, 3235, 3253, 3271
\ifx 4499, 4500, 4505
\ifzhmappdf 4517, 4584, 4615, 4646, 4713
\ignorespaces 2870
\immediate 4508
indent 29, 201
\indexname 2257
indexname 12
\input 4770, 4835
\insertcontinuationtext 2266, 2275, 2282
\insertpart 3239, 3240
\insertromanpartnumber 3230, 3328
\insertsection 3257, 3258
\insertsectionnumber 3248, 3426
\insertsubsection 3275, 3276
\insertsubsectionnumber 3266, 3455
int commands:
\c_five 3698
\c_four 311, 567, 569
\int_compare:nNnTF
..... 560, 2131, 2175, 3514, 3530, 3546, 3637, 3664
\int_const:Nn 968
\int_div_truncate:nn 78, 567

- \int_eval:n 1476, 1478, 1479, 1581, 1583, 1584
- \int_from_hex:n 559
- \int_gincr:N 1126, 1441
- \int_incr:N 2918
- \int_mod:nn 569
- \int_new:N 59, 1444
- \int_set:Nn 550, 558, 2128, 2170
- \int_set_eq:NN 544
- \int_sub:Nn 563, 2176
- \int_to_Hex:n 561, 566, 568
- \int_use:N 470, 550, 881, 1128, 1442, 2922, 4803
- \int_zero:N 2915
- \c_one 162, 1469, 1574, 2183, 2400, 4815
- \c_six 3698
- \c_three 309, 2175, 3530, 4816
- \c_two 95, 163, 377, 381, 2176, 3150, 3514, 3546
- \c_zero 161, 2131, 3638, 3665
- \interlinepenalty 2538, 2577, 2593, 2611, 2714, 2732, 2795, 2796, 2831, 2832
- iow commands:**
 - \iow_close:N 4792
 - \iow_indent:n 4966
 - \iow_new:N 4794
 - \iow_now:Nn 4804
 - \iow_open:Nn 4789
- \itemsep 3789, 3800, 3828, 3839
- \itshape 1653, 1654, 1658, 1659
- J**
- \jfam 933, 969
- K**
- \kaishu 4455
- \kanjifamily 1641, 1642, 1643, 1646
- \kanjifamilydefault 1649
- kernel internal commands:**
 - \l_kernel_expl_bool 70
- keys commands:**
 - \l_keys_choice_tl 294
 - \keys_define:nn 155, 388, 391, 973, 1329, 1438, 1515, 1553, 1592, 2010, 2031, 2066, 2100, 2223, 2249, 2277, 2287, 2290, 2293, 2434, 2931, 3863, 3910
 - \keys_if_exist:nnTF 1590
 - \l_keys_key_tl 141, 143, 145
 - \keys_set:nn 28, 171, 177, 233, 242, 251, 277, 287, 294, 301, 331, 337, 2050, 2057, 2063, 3319, 3374, 3402, 3434, 3469, 3486, 3500, 3516, 3524, 3532, 3540, 3547, 3548, 3550, 3616, 3618, 3952
 - \keys_set_known:nn 3595
 - \keys_set_known:nnN 985
- L**
- \labelformat 3303
- \languagealias 3624, 3629, 3650, 3655
- \LARGE 3807, 3846
- \Large 2542, 3339, 3345, 3413, 3417, 3806, 3845
- \large 3438, 3805, 3844
- \lastbox 2858
- \leavevmode 2484, 2744
- left commands:**
 - \c_left_brace_str 5121
 - \leftmargin 3786, 3797, 3825, 3836
 - \leftmargini 3786, 3797, 3825, 3836
 - \let 1962, 2777, 2786, 3779, 3818, 4492, 4498
 - \linespread 3854
 - linespread 10, 181
 - linestretch 10, 2100
 - \linewidth 2129, 2137, 2169
 - \lishu 4457
 - \listfigurename 2252
 - listfigurename 11
 - \listtablename 2253
 - listtablename 11
 - \LoadClass 411, 416, 421, 426
 - \long 4506
 - \lstlistingname 759, 760
 - \lstlistlistingname 761, 762
 - \ltjchar 726
 - \ltjdefcharrange 685, 686, 687, 689, 693, 694, 697, 700
 - LTJFONTUID 1438
 - \ltjgetparameter 1902, 1932, 1992, 2006
 - \ltjsetkanjiskip 1919
 - \ltjsetmathletter 733
 - \ltjsetparameter 701, 707, 1631
 - \ltjsetxkanjiskip 1996
- lua commands:**
 - \lua_escape_x:n 842, 854, 881, 888, 1466, 1481, 1482, 1499, 1500, 1509
 - \lua_now_x:n 719, 816, 842, 846, 851, 856, 878, 887, 955, 1487, 1511, 1736
- M**
- \MAKESPA 4795, 4837
- \markboth 2456, 2547, 2575
- \mathgroup 933
- \mcfamily 1658
- \mddefault 953, 959, 965, 1317, 1322
- mode commands:**
 - \mode_if_math:TF 927
- msg commands:**
 - \msg_critical:nnn 53, 676, 3879
 - \msg_error:nn 3930
 - \msg_error:nnn 19, 22, 38, 41, 667, 680, 683, 2239, 3680, 3762
 - \msg_error:nnnn 3896, 3925
 - \msg_fatal:nn 4776
 - \msg_new:nn 4774
 - \msg_new:nnn 140, 142, 144, 146, 148, 670, 1025, 1094, 1176, 1849, 2270, 3569, 3880, 3939, 4962, 5039
 - \msg_new:nnnn 12, 30, 45, 99, 2241, 3682, 3904, 3933
 - \msg_warning:nn 1171, 1853, 2028, 2039, 2292
 - \msg_warning:nnn 169, 175, 203, 213, 228, 237, 246, 275, 285, 299, 329, 335, 343, 348, 354, 1089, 3588, 3867, 3917, 4968, 5038

\msg_warning:nnnn 1021, 1862, 1869, 2055, 2061, 2193, 2199

N

\newCJKfontfamily 1195
 \NewDocumentCommand
 1055, 1190, 1195, 1201, 1211, 1217, 1223,
 1230, 1236, 1242, 1248, 1376, 1468, 1573, 1608, 1644,
 2050, 2053, 2059, 2083, 2154, 2191, 2197, 2218, 2220,
 2399, 3674, 3859, 4452, 4453, 4454, 4455, 4457, 4458, 4460
 \newfontfeature 970, 971, 972
 \newpage 2625, 2630
 \ngostype 44, 1754
 \nobreak 2542,
 2550, 2600, 2710, 2715, 2718, 2733, 2736, 2846, 3340, 3388
 \nobreakspace 2542, 2581
 nocap 29
 \noexpand 4490, 4492, 4493, 4494
 nofonts 29
 \noindent 2876
 noindent 29
 nopunct 29
 \normalem 1832
 \normalfont 533, 535, 536, 1227,
 1234, 1240, 1274, 1276, 1277, 1300, 2539, 2540, 2578,
 2579, 2594, 2595, 2612, 2613, 2705, 2706, 2729, 2730, 2926
 \normalsize 3472, 3489, 3503, 3773, 3812, 3850, 3858
 nospace 29
 nozhmap 29
 \null 2514, 2628
 \numberline 2662, 2803, 2817, 2904, 2912, 3088, 3144

O

one commands:

\c_minus_one 152, 544, 3637, 3664
 \onecolumn 2509

or commands:

\or: 405, 3768

P

\pagestyle 3615
 \par 2447, 2485, 2542, 2581, 2584, 2597,
 2614, 2710, 2715, 2733, 2745, 2846, 3231, 3232, 3239,
 3242, 3249, 3250, 3257, 3260, 3267, 3268, 3275, 3278,
 3321, 3340, 3351, 3369, 3377, 3388, 3429, 3431, 3463, 3465
 paragraph/afterskip 31
 paragraph/beforeskip 31
 paragraph/hang 21
 paragraph/numbering 15
 paragraph/runin 19
 \parindent
 2150, 2151, 2173, 2194, 2195, 2200, 2201, 2539, 2540,
 2594, 2595, 2705, 2706, 2729, 2730, 3233, 3251, 3269, 3548
 \parsep 3788, 3789, 3799, 3800, 3827, 3828, 3838, 3839
 \part 2482
 part/beforeskip 31
 part/fixbeforeskip 31
 part/numbering 15

part/pagestyle 21, 2409
 \partmark 2455, 2548, 2576
 \partname 2542, 2581, 3230, 3323
 \PassOptionsToClass 410, 415, 420, 425
 \PassOptionsToPackage 488, 1797, 2203, 4970
 \pdfmapline 4518,
 4519, 4520, 4521, 4522, 4523, 4524, 4526, 4527, 4528,
 4529, 4530, 4531, 4532, 4534, 4535, 4536, 4537, 4538,
 4539, 4541, 4542, 4543, 4544, 4545, 4546, 4647, 4648,
 4649, 4650, 4651, 4652, 4653, 4655, 4656, 4657, 4658,
 4659, 4660, 4661, 4663, 4664, 4665, 4666, 4667, 4668,
 4670, 4671, 4672, 4673, 4674, 4675, 4714, 4715, 4716,
 4717, 4718, 4719, 4721, 4722, 4723, 4724, 4725, 4726,
 4728, 4729, 4730, 4731, 4732, 4734, 4735, 4736, 4737, 4738
 \pdfoutput 4501
 \pdfstringdefDisableCommands 3862
 pdftex commands:
 \pdftex_pdffontattr:D 469
 \pdftex_pdflastobj:D 470
 \pdftex_pdfobj:D 466
 prg commands:
 \prg_do_nothing:
 937, 1415, 1507, 1878, 1952, 2015, 2888, 5162
 \prg_new_conditional:Npnn 840, 1925
 \prg_new_protected_conditional:Npnn 1068, 5132
 \prg_return_false: ... 843, 1078, 1929, 1933, 1937, 5142
 \prg_return_true: 843, 1071, 1076, 1929, 1933, 1937, 5147
 \primitive 4871
 \ProcessKeysOptions 400
 \proofname 2244, 2259
 proofname 12
 prop commands:
 \prop_get:NnN 1163
 \prop_get:NnNTF
 ... 1070, 1081, 1122, 1160, 1408, 1456, 3678, 3733, 3760
 \prop_gpop:NnNTF 1010, 1014, 1543
 \prop_gput:Nnn ... 989, 990, 1036, 1150, 1531, 1567, 3695
 \prop_gremove:Nn 1019
 \prop_if_empty:NTF 1084, 1281
 \prop_map_break:n 1295
 \prop_map_inline:Nn 1293
 \prop_new:N 1005, 1006, 1007, 1155, 1551, 1572, 3689
 \protect 2469, 2475, 2662,
 2682, 2683, 2803, 2817, 2904, 2912, 3087, 3099, 3101,
 3105, 3107, 3111, 3113, 3121, 3126, 3138, 3139, 5093, 5094
 \ProvideDocumentCommand 2455
 \ProvidesFile 4506, 4514, 4581, 4612, 4643, 4710
 \providetranslation
 4876, 4877, 4878, 4879, 4880, 4881, 4882,
 4883, 4884, 4885, 4886, 4887, 4888, 4889, 4890, 4891,
 4892, 4893, 4894, 4895, 4896, 4897, 4898, 4899, 4900,
 4901, 4902, 4903, 4904, 4905, 4906, 4907, 4908, 4909,
 4910, 4911, 4912, 4913, 4914, 4915, 4916, 4917, 4918,
 4919, 4920, 4921, 4922, 4923, 4924, 4925, 4926, 4927,
 4928, 4929, 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

ptex commands:

\ptex_kanjiskip:D 1905, 1922, 1936
\ptex_xkanjiskip:D 1993, 1997, 2007
punct 9, 29, 280, 2031
\punctstyle 578, 2036

Q

\quad 2770, 3178, 3179, 3181, 3182, 3185, 3186, 3198,
3210, 3213, 3218, 3346, 3396, 3406, 3439, 3473, 3490, 3504

quark commands:

\q_mark 5136, 5139
\q_stop 120, 125, 892,
895, 1373, 1377, 1604, 1609, 5101, 5106, 5115, 5136, 5139
\quotation 3872

R

\raggedright 2539, 2594, 2705, 2729, 3338, 3386
\refname 2260, 2265, 2272, 2281, 2289, 2290
refname 12
\refstepcounter ... 2525, 2562, 2658, 2781, 3286, 3308, 3309
\relax 2540, 2559, 2580, 2595, 2706, 2730, 2746,
2782, 2789, 2794, 2809, 2825, 2830, 2837, 2838, 2843,
2862, 2864, 3209, 3212, 3232, 3233, 3242, 3250, 3251,
3260, 3268, 3269, 3278, 4464, 4468, 4471, 4499, 4500, 4505
\renewcommand 2483, 2499, 2638
\RenewDocumentCommand 3309
\RequirePackage 27,
56, 57, 350, 361, 489, 492, 493, 494, 660, 678, 681,
715, 758, 1697, 1806, 1815, 1831, 1837, 2204, 3669, 5015
\RequirePackageWithOptions 4971
resetalternatefont 27, 1515
reverse commands:
\reverse_if:N 581
\rmdefault 1710
\rmfamily 521, 524, 1262, 1265, 1641

S

scan commands:

\scan_stop:
... 64, 120, 814, 933, 1508, 1509, 2121, 2129, 4799, 4981
scheme 9, 315
\scriptsize 3803, 3842
\secdef 2496, 2518, 2650
section/afterskip 31
section/beforeskip 31
section/hang 21
section/numbering 15
section/runin 19
\sectionname 3248, 3423
\selectfont 1062, 1534, 1547, 1647, 1966, 1971, 1985, 2085, 3679

seq commands:

\seq_clear:N 983
\seq_concat:NNN 1349
\seq_gput_left:Nn 2308, 2309
\seq_gput_right:Nn 1088, 3701

\seq_gset_eq:NN 2307
\seq_gset_from_clist:Nn 2302, 2312, 2962, 4779
\seq_if_empty:NTF 1364
\seq_if_in:NnTF 446, 1086
\seq_map_function:NN 3020, 3142
\seq_map_inline:Nn
... 1372, 2426, 2916, 2965, 2975, 2984, 3079, 3297, 4802
\seq_new:N
... 439, 1093, 1357, 1358, 2301, 2305, 2961, 3690, 4778
\seq_put_right:Nn 1620
\seq_set_filter:NNn 1347
\seq_set_from_clist:Nn 440
\seq_set_split:Nnn 1346
\seq_use:Nnnn 3687
\setbox 2858
\setCJKfamilyfont 1107,
1195, 4096, 4102, 4103, 4104, 4105, 4106, 4107, 4150,
4151, 4159, 4160, 4162, 4163, 4164, 4165, 4167, 4189,
4191, 4193, 4195, 4234, 4235, 4236, 4237, 4261, 4266,
4271, 4273, 4311, 4313, 4315, 4316, 4330, 4331, 4332,
4333, 4346, 4347, 4348, 4349, 4350, 4351, 4392, 4393,
4394, 4395, 4396, 4397, 4410, 4411, 4412, 4445, 4446, 4447
\setCJKmainfont 1103, 1223, 4088, 4146,
4154, 4181, 4227, 4249, 4306, 4327, 4342, 4387, 4406, 4441
\setCJKmathfont 1223
\setCJKmonofont 1105, 1223, 4101, 4149,
4158, 4188, 4233, 4260, 4310, 4329, 4345, 4391, 4409, 4444
\setCJKromanfont 1229, 1255
\setCJKsansfont 1104, 1223, 4091, 4095, 4148,
4156, 4187, 4232, 4255, 4309, 4328, 4344, 4389, 4408, 4443
\SetSymbolFont 966, 1321, 1325
\sfdefault 1711
\sffamily 525, 528, 1266, 1269, 1642
skip commands:
\skip_horizontal:N 1912, 1916
\skip_if_eq:nnTF 1932, 1936, 1991, 2466, 2472
\skip_new:N 1924, 2004, 2451
\skip_set:Nn 1902,
1905, 1995, 2005, 2122, 2133, 2178, 2186, 2453, 2465, 2471
\skip_set_eq:NN 1922, 1997
\skip_sub:Nn 2442, 2449
\skip_use:N 2469, 2475
\skip_zero:N 2142
\c_zero_skip 2391, 2392, 2417, 2418, 2466, 2472
\small 3781, 3820
\songti 4452
\space 2659, 2710, 3323, 3384, 3423, 3454, 3556
space 10, 29, 290, 2010
\special 1679, 1680, 1683,
1684, 1813, 1827, 4549, 4550, 4551, 4552, 4553, 4554,
4555, 4557, 4558, 4559, 4560, 4561, 4562, 4563, 4565,
4566, 4567, 4568, 4569, 4570, 4572, 4573, 4574, 4575,
4576, 4577, 4588, 4589, 4590, 4591, 4592, 4594, 4595,
4596, 4597, 4598, 4600, 4601, 4602, 4603, 4605, 4606,
4607, 4608, 4619, 4620, 4621, 4622, 4623, 4625, 4626,
4627, 4628, 4629, 4631, 4632, 4633, 4634, 4636, 4637,

- 4638, 4639, 4678, 4679, 4680, 4681, 4682, 4683, 4684,
4686, 4687, 4688, 4689, 4690, 4691, 4692, 4694, 4695,
4696, 4697, 4698, 4699, 4701, 4702, 4703, 4704, 4705,
4706, 4741, 4742, 4743, 4744, 4745, 4747, 4748, 4749,
4750, 4751, 4753, 4754, 4755, 4756, 4758, 4759, 4760, 4761
- \SplitArgument 1469, 1574, 2400
- str commands:
- \c_backslash_str 5120
 - \c_colon_str 919, 920, 921
 - \str_case:nnTF 1708, 1754, 5091, 5106
 - \str_case_x:nnTF 1101, 1112
 - \str_const:Nn 43, 50, 1779, 1787
 - \str_if_empty:NTF 5177, 5213, 5222
 - \str_if_eq:nnTF 491,
582, 941, 1802, 1810, 3620, 3647, 3661, 3919, 3936, 3943
 - \str_if_eq_x:nnTF 577, 1285, 3916, 5097
 - \str_lower_case:n 465
 - \str_new:N 5015, 5126, 5127, 5128
 - \str_set:Nn 5144, 5145, 5146
 - \str_upper_case:n 1780
- \strutbox 3856
- sub3section 9, 307
- sub4section 9, 307
- 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 3266, 3454
- 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_pdf:TF
130, 4083, 4174, 4242, 4321, 4338, 4402
 - \sys_if_engine_uptex:TF
4127, 4215, 4294, 4324, 4371, 4430
 - \sys_if_engine_xetex:TF 4390, 4772
 - \sys_if_output_pdf:TF 105, 475, 1805, 4176, 4244
- T**
- \tablename 2255
- tablename 12
- \TeX 3860
- TeX and L^AT_EX₂_ε commands:
- \@italiccorr 1633
 - \@par 2795, 2831, 3407, 3440, 3474, 3518, 3534
 - \@afterheading 2555, 2605, 2689, 2698, 2851
 - \@afterindentfalse 2494, 2644, 2648, 2749, 2755
 - \@afterindenttrue 2492, 2646, 2747, 2753
 - \@Alph 3552, 3557
 - \@chapapp 2659, 2710, 3194, 3217
 - \@chapter 2650, 2651
 - \@chinese 2206, 2215
 - \@clubpenalty 2866
 - \@currentHref 2996
 - \@currentlabel 3288, 3292
 - \@currentxt 4, 4961, 5010
 - \@currname 4, 4961, 5011
 - \@dblarg 2768
 - \@defaultunits 36
 - \@empty 99, 2777, 2786
 - \@endpart 2587, 2617, 2619
 - \@EverySelectfont@Init 1953, 1961
 - \@firstoftwo 3138
 - \@hangfrom 2875
 - \@ifclassloaded 3565
 - \@ifpackagelater 37, 40, 666, 679, 682, 1838
 - \@ifpackageloaded 18, 21, 675, 1790, 3059, 3162, 3305, 4992
 - \@ifpackagewith 3004, 3013, 3064
 - \@ifstar 2766
 - \@listI 3779, 3818
 - \@listi 3779, 3786, 3797, 3818, 3825, 3836
 - \@M 2538,
2577, 2593, 2611, 2714, 2732, 2795, 2796, 2831, 2832, 2859
 - \@makechapterhead 2686, 2688, 2700
 - \@makeschapterhead 2695, 2697, 2724
 - \@minus 3408, 3441, 3475, 3491, 3505, 3775,
3777, 3783, 3785, 3787, 3788, 3794, 3796, 3798, 3799,
3814, 3816, 3822, 3824, 3826, 3827, 3833, 3835, 3837, 3838
 - \@namedef 3099, 3101, 3105, 3107, 3111, 3113, 3121, 3126
 - \@nameuse 3099, 3101, 3111, 3113, 3121
 - \@ne 3057, 3181, 3212
 - \@nil 867
 - \@nobreakfalse 2853
 - \@noskipsecfalse 2857
 - \@noskipsectrue 2854
 - \@onlypreamble 36, 436, 502,
1251, 1252, 1253, 1254, 1255, 1256, 1674, 1688, 1696, 3903
 - \@part 2496, 2518, 2520
 - \@pkgextension 6, 7, 8, 9, 20, 23, 677, 5010
 - \@plus 3352,
3353, 3408, 3409, 3441, 3442, 3475, 3476, 3491, 3505,
3519, 3535, 3775, 3776, 3777, 3783, 3784, 3785, 3787,
3788, 3794, 3795, 3796, 3798, 3799, 3814, 3815, 3816,
3822, 3823, 3824, 3826, 3827, 3833, 3834, 3835, 3837, 3838
 - \@popfilename 130, 5008
 - \@schapter 2650, 2691
 - \@seccntformat 2769, 2782
 - \@secondoftwo 3139
 - \@secpenalty 2760, 3411, 3444, 3478, 3492, 3506
 - \@sect 2768, 2773
 - \@setfontsize 3765
 - \@spart 2496, 2518, 2589

\@ssect	2767, 2823	\CJK@surr	45, 45, 554
\@startsection	91, 93, 100, 2743, 2921	\CJK@upperReset	46
\@svsec	2777, 2782, 2786, 2794, 2810, 2811	\CJKfamily	56
\@svsechd	2808, 2837, 2838, 2860	\CJKfamilydefault ..	45, 46, 60, 60, 67, 68, 68, 68, 69, 69
\@tempdima	3147, 3149	\CJKglue	78, 78, 109
\@tempskipa	2746, 2748, 2749, 2760, 2789, 2790, 2825, 2826, 2843, 2844, 2847, 2862, 2863	\CJKhook	46, 46
\@tempswafalse	2512	\CJKrmdefault	43
\@tempswatrue	2510	\CJKsymbol	75
\@topnewpage	2686, 2695	\CJKunderdotbasesep	30
\@topnum	2643	\CJKunderline	30
\@xsect	2822, 2841, 2842	\clearpage	46
\@zhdig	2214	\contentsname	11, 11
\@zhnum	2206, 2213	\CS	51, 51
\abstractname	12	\csname	109
\addCJKfontfeature	64	\CTEX@addloflotskip	2462, 2684
\AfterEndPreamble	130	\CTEX@addtocline ...	2479, 2537, 2574, 2680, 2806, 2820
\algorithmname	12	\CTEX@afterindent	2752, 2882, 2889
\appendixname	12, 23, 23, 23	\CTEX@aftertitle	2834, 2839, 2881, 2888
\AtBeginDocument	36, 45	\CTEX@appendix@number	2936, 2948, 2954
\AtBeginDvi	42	\CTEX@appendix@numbering	2937, 2950, 2956
\AtBeginShipoutFirst	42	\CTEX@chapter@afterindent	2645
\AtEndOfClass	37	\CTEX@chapter@aftername	2713
\AtEndOfPackage	37	\CTEX@chapter@afterskip	2720, 2738
\AtEndOfPackageFile*	130	\CTEX@chapter@aftertitle	2717, 2735
\AtEndPreamble	130	\CTEX@chapter@beforeskip	2702, 2726
\baselineskip	10, 11, 75, 78, 78	\CTEX@chapter@break	2641
\bfseries	47	\CTEX@chapter@fixskip	2703, 2721, 2727, 2739
\bibname	12	\CTEX@chapter@format	2707, 2731
\c@chapter	3557	\CTEX@chapter@indent	2706, 2730
\c@secnumdepth	2522, 2541, 2559, 2580, 2652, 2708, 2774, 2802, 2816, 3178, 3181, 3185, 3192, 3193, 3201, 3209, 3212, 3217, 3220	\CTEX@chapter@lofskip	2415
\c@section	3552	\CTEX@chapter@lotskip	2416
\caption	30	\CTEX@chapter@numbering	2656, 2956
\catcode	35, 35, 44, 44, 44, 121, 131	\CTEX@chapter@pagestyle	2414, 2642
\ccwd ..	10, 10, 11, 24, 25, 30, 30, 36, 73, 74, 74, 77, 77, 78, 78	\CTEX@chapter@titleformat	2716, 2734
\chapter	9, 15, 16, 21, 23, 23, 84, 104	\CTEX@chapter@tocline	2893
\chaptermark	13	\CTEX@chaptername	2713
\chaptername	15	\CTEX@document@left@hook	4978, 4984
\char	49	\CTEX@document@right@hook	4980, 4986
\CheckCommand	74	\CTEX@Family@CMap	451, 479, 482
\Chinese	30, 30	\CTEX@fixheadingskip	2440, 2445, 2489, 2553, 2603, 2623, 2721, 2739, 2763, 2849
\chinese	25, 25, 30, 30, 80	\CTEX@fixskip	2763, 2849, 2883, 2890
\CJK@@ignorespaces	46, 595	\CTEX@fixtopskip	2437, 2516, 2703, 2727
\CJK@encoding	54, 56, 63, 773, 777, 861, 948, 1316, 1319, 1321, 1325, 1466, 1481, 1482, 1499, 1500	\CTEX@hang	2874, 2884, 2891
\CJK@envStart	45, 45, 45, 46, 589	\CTEX@hangfrom	2794, 2830, 2872
\CJK@family	50, 765, 778, 825, 861, 1059, 1533, 1543, 1545, 1985, 1986	\CTEX@headingskip	2454, 2490, 2517, 2554, 2604, 2624, 2704, 2722, 2728, 2740, 2764, 2850
\CJK@ignorespaces	46, 46, 600, 604	\CTEX@hyperheadinghook	2970, 3000
\CJK@input	45, 538	\CTEX@ifnamefalse	2457, 2528, 2533, 2565, 2570, 2591, 2609, 2664, 2670, 2676, 2692, 2775, 2784, 2824
\CJK@loadBinding	44, 45, 585	\CTEX@ifnametrue	2457, 2524, 2561, 2657, 2780
\CJK@loadEncoding	44	\CTEX@makeanchor	2529, 2534, 2566, 2571, 2665, 2785, 2959, 2968, 2980, 2985, 2991
\CJK@makeActive	45, 584	\CTEX@makeanchor@chapter	2671, 2677
\CJK@plane	45, 75, 454, 455, 465, 1974	\CTEX@makeanchor@schapter	2693
		\CTEX@makeanchor@sect	2776

<code>\CTEX@makeanchor@spart</code>	2592, 2610	<code>\CTEX@verbatim@font@hook</code>	1626, 1630
<code>\CTEX@makeanchor@ssect</code>	2767	<code>\ctex@zhmap@endinput</code>	4766
<code>\CTEX@part@afterindent</code>	2491	<code>\CTEXdigits</code>	25, 25
<code>\CTEX@part@aftername</code>	2543, 2582, 3236	<code>\CTEXifname</code>	16, 16
<code>\CTEX@part@afterskip</code>	2552, 2602, 2622, 3242	<code>\CTEXnumber</code>	25, 25, 25, 25, 25
<code>\CTEX@part@aftertitle</code> ...	2549, 2586, 2599, 2616, 3240	<code>\CTEXnumberline</code>	22
<code>\CTEX@part@beforeskip</code>	2488, 2515, 3232	<code>\CTEXoptions</code>	30, 30, 76
<code>\CTEX@part@break</code>	2486, 2505	<code>\ctexset</code> .	5, 5, 5, 5, 5, 5, 5, 7, 11, 14, 14, 26, 27, 30, 30, 111
<code>\CTEX@part@fixskip</code>	2489, 2516, 2553, 2603, 2623	<code>\CTEXsetup</code>	30
<code>\CTEX@part@format</code>	2540, 2579, 2596, 2613, 3234	<code>\CTEXthechapter</code>	12, 13, 13, 16, 95
<code>\CTEX@part@indent</code>	2540, 2595, 3233	<code>\CTEXthesection</code>	12, 13, 16, 97
<code>\CTEX@part@numbering</code> ...	2523, 2541, 2560, 2580, 3235	<code>\CTEXunderdotbasesep</code>	30
<code>\CTEX@part@pagestyle</code>	2413, 2507	<code>\CTEXunderline</code>	30
<code>\CTEX@part@titleformat</code> ..	2546, 2585, 2598, 2615, 3240	<code>\curr@fontshape</code>	779, 809, 818, 822, 868
<code>\CTEX@part@tocline</code>	2893	<code>\DeclareAlternateKanjiFont</code>	63
<code>\CTEX@partname</code>	2543, 2582, 3236	<code>\DeclareFontEncoding</code>	53
<code>\CTEX@postappendix</code>	2941, 2949, 2955	<code>\DeclareFontFamily</code>	42, 51
<code>\CTEX@postchapter</code>	2955	<code>\DeclareFontShape</code>	51
<code>\CTEX@postsection</code>	2949	<code>\DeclareFontShape@</code>	63, 1425
<code>\CTEX@preappendix</code>	2940, 2947, 2953	<code>\DeclareRobustCommand</code>	132
<code>\CTEX@prechapter</code>	2953	<code>\define@newfont</code>	50, 801
<code>\CTEX@presection</code>	2947	<code>\do@subst@correction</code>	50, 51, 800
<code>\CTEX@runin</code>	2791, 2827, 2845, 2885, 2892	<code>\em</code>	67, 67, 67
<code>\CTEX@save@appendix</code>	2942, 2945	<code>\escapechar</code>	53
<code>\CTEX@save@refstepcounter</code>	3308, 3311	<code>\EverySelectfont</code>	74
<code>\CTEX@section@aftername</code>	3254	<code>\external@font</code>	811, 812, 814
<code>\CTEX@section@afterskip</code>	3260	<code>\extract@font</code>	50, 799
<code>\CTEX@section@aftertitle</code>	3258	<code>\f@baselineskip</code>	2121
<code>\CTEX@section@beforeskip</code>	3250	<code>\f@encoding</code>	777, 817, 930
<code>\CTEX@section@format</code>	3252	<code>\f@family</code>	50, 778, 817, 825
<code>\CTEX@section@indent</code>	3251	<code>\f@series</code>	773, 861
<code>\CTEX@section@numbering</code>	2950, 3253	<code>\f@shape</code>	773, 861
<code>\CTEX@section@titleformat</code>	3258	<code>\f@size</code>	773, 779, 856, 864, 867, 868, 1898, 1944, 3150
<code>\CTEX@sectionname</code>	3254	<code>\fangsong</code>	8, 8
<code>\CTEX@selectfont@hook</code>	1947, 1974, 1977	<code>\figurename</code>	11
<code>\CTEX@selectfont@save</code>	1958, 1962	<code>\font</code>	51, 51
<code>\CTEX@setcurrentlabel@n</code>	3286, 3312	<code>\font@name</code>	50, 51, 782, 789, 790, 793, 796, 814, 815, 832, 910, 911, 912, 913, 914, 921
<code>\CTEX@setheadingskip</code>	2452, 2488, 2515, 2552, 2602, 2622, 2702, 2720, 2726, 2738, 2762, 2848	<code>\fontcharwd</code>	74, 74
<code>\CTEX@subsection@aftername</code>	3272	<code>\footnotesep</code>	109, 109, 109
<code>\CTEX@subsection@afterskip</code>	3278	<code>\footnotesize</code>	109
<code>\CTEX@subsection@aftertitle</code>	3276	<code>\get@external@font</code>	808
<code>\CTEX@subsection@beforeskip</code>	3268	<code>\getanddefine@fonts</code>	899
<code>\CTEX@subsection@format</code>	3270	<code>\globaljfont</code>	51
<code>\CTEX@subsection@indent</code>	3269	<code>\H@old@chapter</code>	2974
<code>\CTEX@subsection@numbering</code>	3271	<code>\hbox</code>	84
<code>\CTEX@subsection@titleformat</code>	3276	<code>\heiti</code>	8
<code>\CTEX@subsectionname</code>	3272	<code>\hrule</code>	84
<code>\CTEX@thechapter</code>	2954	<code>\Hy@chapapp</code>	2665, 2671, 2677, 2693
<code>\CTEX@thesection</code>	2948	<code>\Hy@driver</code>	1810
<code>\CTEX@titleformat@n</code>	91, 2833, 2839, 2880, 2887	<code>\Hy@MakeCurrentHrefAuto</code>	2993
<code>\CTEX@titleslabel@clear</code>	3108, 3118	<code>\Hy@org@chapter</code>	2974
<code>\CTEX@titleslabel@set</code>	3102, 3114, 3118	<code>\Hy@raisedlink</code>	2994
<code>\CTEX@toc@width@n</code>	3144, 3159, 3169	<code>\Hy@unicodetrue</code>	71
<code>\CTEX@todayold</code>	2222, 2227	<code>\hyper@anchorend</code>	2997
<code>\CTEX@update@sectionformat@n</code>	2751, 2878	<code>\hyper@anchorstart</code>	2996

\HyPsd@ConvertToUnicode	71	3381, 3388, 3775, 3776, 3777, 3783, 3784, 3785, 3787,	
\HyPsd@LoadUnicode	71	3788, 3794, 3795, 3796, 3798, 3799, 3814, 3815, 3816,	
\HyPsd@pdfencoding	71, 71	3822, 3823, 3824, 3826, 3827, 3833, 3834, 3835, 3837, 3838	
\HyPsd@ToBigChars	71	\pagenumbering	25
\if@mainmatter	2654, 2709, 3192, 3583, 3584	\pagestyle	9
\if@nobreak	2757	\paragraph	9, 9, 9, 9, 9, 19, 35, 35
\if@noskipsec	2484, 2744, 2856	\parindent	10, 30, 77, 79, 109
\if@openright	2500, 2627, 2639, 3355, 3382	\parskip	84, 84, 84
\if@tempswa	2633	\part	14, 20, 21, 84
\if@twocolumn	2508, 2685, 2694	\partmark	84
\if@twoside	2626, 3176, 3199	\partname	15
\iffalse	53	\partpage	14, 98
\ifHy@implicit	2973	\PassOptionsToClass	42
\ifin@	53, 889	\PassOptionsToPackage	70
\iftrue	53	\pdfmapline	115, 118
\indexname	12	\pdfstringdef	71
\insertcontinuationtext	12	\pickup@font	50
\itdefault	51	\pickup@jfont	805
\itshape	47	\ProcessKeysOptions	28
\kaishu	8	\proofname	12, 80
\kanjiskip	75	\protected	96
\labelformat	99, 99	\protected@edef	2219, 2221, 2782, 3288
\leftmark	13	\ProvidesFile	122
\linewidth	78, 78	\ps@fancy	3205
\lishu	8, 8	\ps@headings	3175
\listfigurename	11	\refname	12, 12
\listtablename	11	\refstepcounter	99
\ltj@does@alt@set	52	\relax	45, 77
\ltj@getjfontnumber	53, 877	\reset@font	535, 1276
\ltj@set@stackfont	919, 920, 921	\rightmark	13
\ltj@allalchar	720	\rmfamily	48
\ltj@curjfnt	51, 75	\roman	25
\ltj@pickup@altfont@auxy	862	\scantokens	132
\ltj@pickup@altfont@copy	875	\scriptfont@name	913, 920
\ltj@setpar@global	918	\section 9, 9, 14, 15, 19, 20, 20, 21, 21, 22, 23, 23, 31, 81, 97	
\ltj@tempcntc	53, 881	\section*	15
\ltjalchar	49	\sectionmark	13, 13
\m@ne	2522, 2541, 2652, 2708,	\sectionname	15
	3185, 3192, 3193, 3217, 3963, 3964, 3965, 3966, 3967, 3968	\sectionpage	14, 98
\markboth	13	\selectfont	52, 68, 74, 74, 74, 75, 77
\markright	13	\setCJKfamilyfont	26, 26, 26
\math@bgroup	929	\setCJKmainfont	26
\math@egroup	934	\sf@size	912
\math@fonts	916	\sffamily	48
\mathbb	109	\size@update	77, 78, 2085, 2099
\maxdimen	11, 77	\sldefault	51
\meaning	132, 133, 133, 134	\songti	7
\newcommand	132	\special	7, 70
\newrobustcmd	132	\split@name	867
\newtitlemark	96, 96	\SplitArgument	83
\nfss@catcodes	54	\ssf@size	914
\ngostype	69	\strutbox	109
\normalsize	8, 8, 105	\subparagraph	9, 9, 9, 9, 19, 35
\nouppercase	45	\subsectionname	15
\p@	1898, 1944, 2581, 2682, 2683, 2701,	\subsectionpage	14, 98
	2710, 2719, 2725, 2737, 3150, 3351, 3378, 3379, 3380,	\subst@correction	828, 835

<code>\tablename</code>	12	<code>\theparagraph</code>	3488
<code>\textfont@name</code>	911, 919	<code>\thepart</code>	2526, 2542, 2563, 2581, 3325
<code>\tf@size</code>	910	<code>\thesection</code>	3178, 3185, 3201, 3209, 3220, 3405
<code>\thechapter</code>	13, 16, 23	<code>\thesubparagraph</code>	3502
<code>\thesection</code>	13, 15, 16, 23, 97	<code>\thesubsection</code>	3181, 3212, 3437
<code>\titleformat</code>	93	<code>\thesubsubsection</code>	3471
<code>\titlespacing</code>	93	<code>\thispagestyle</code>	2506, 2507, 2629, 2640, 2642
<code>\today</code>	11, 11	<code>\tiny</code>	3804, 3843
<code>\trans@languagepath</code>	3632, 3658	tl commands:	
<code>\ttfamily</code>	48	<code>\c_space_tl</code>	470
<code>\ttl@a</code>	3077, 3085	<code>\tl_clear:N</code>	193, 366, 389, 1565, 1569, 1834, 1845, 1986, 2076, 2406, 2424, 3730, 5178
<code>\ttl@chapterout</code>	3072	<code>\tl_const:Nn</code> 3, 409, 414, 419, 424, 517, 655, 677, 948, 963, 1258, 1259, 1260, 1261, 1315, 2095, 2319, 2325, 3566, 3585, 3587
<code>\ttl@extract</code>	94	<code>\tl_gput_right:Nn</code>	447, 4975, 4977, 4999
<code>\ttl@setifthe</code>	3131	<code>\tl_gset:Nn</code>	561, 564, 910, 912, 914, 1289, 1706, 1734, 1752, 1777, 1784, 1785, 3921
<code>\ttl@setsubmark</code>	95, 3104, 3110	<code>\tl_gset_eq:NN</code>	1038, 1729
<code>\ttl@settopmark</code>	95, 3098, 3129	<code>\tl_gset_rescan:Nnn</code>	1296
<code>\ttl@tocpart</code>	3074, 3076	<code>\tl_head:N</code>	3734
<code>\ttlh@hang</code>	3038, 3048	<code>\tl_if_blank:nTF</code>	1382, 1478, 1479, 1583, 1584, 1598, 1616
<code>\ttlh@runin</code>	3037, 3047	<code>\tl_if_blank_p:n</code>	1348
<code>\uppercase</code>	45, 45	<code>\tl_if_empty:NnTF</code>	768, 2148, 2331, 3302, 3884
<code>\use@mathgroup</code>	746, 925	<code>\tl_if_empty:nTF</code>	1668, 1681, 5141
<code>\usepackage</code>	29, 30	<code>\tl_if_eq:NNTF</code>	1702, 2088, 3645
<code>\verbatim@font</code>	1624, 1628	<code>\tl_if_exist:NnTF</code>	84, 516, 518, 519, 520, 1257, 1259, 1260, 1261, 2244, 2275, 2286, 2289, 3578, 3668, 4998
<code>\wrong@fontshape</code>	50	<code>\tl_if_in:NnTF</code>	5205
<code>\x@protect</code>	5094	<code>\tl_map_inline:Nn</code>	1494
<code>\xdef</code>	96	<code>\tl_map_inline:nn</code>	3731
<code>\XeTeXglyphbounds</code>	126	<code>\tl_new:N</code>	58, 128, 132, 518, 519, 520, 770, 771, 859, 945, 998, 1024, 1044, 1045, 1066, 1430, 1571, 1722, 2000, 2096, 2109, 2246, 2317, 2318, 2940, 2941, 4074, 4988, 4989, 4998, 5059, 5060, 5181, 5182, 5183
<code>\xkanjiskip</code>	75, 75, 75, 75	<code>\tl_put_left:Nn</code>	1641, 1642, 1643, 1961, 5008
<code>\yahei</code>	8, 8	<code>\tl_put_right:Nn</code>	373, 916, 1578, 1611, 2347, 2371, 2379, 2381, 2383, 2385, 2387, 2389, 3734, 3735
<code>\youyuan</code>	8, 8	<code>\tl_replace_all:Nnn</code>	1345, 1386
<code>\z@</code>	707, 710, 2539, 2594, 2643, 2705, 2729, 2748, 2790, 2826, 2844, 2858, 3057, 3178, 3201, 3209, 3220, 3776, 3784, 3795, 3815, 3823, 3834	<code>\tl_rescan:nn</code>	134, 5223, 5232
<code>\zhdig</code>	80	<code>\tl_set:Nn</code>	114, 115, 129, 188, 221, 222, 465, 497, 498, 499, 557, 759, 761, 772, 811, 860, 942, 943, 984, 1049, 1061, 1203, 1344, 1383, 1385, 1426, 1599, 1603, 1637, 1638, 1639, 1640, 1649, 1724, 1957, 1985, 2001, 2035, 2044, 2072, 2110, 2158, 2247, 2404, 2407, 2411, 3034, 3053, 3072, 3077, 3085, 3644, 3862, 4075, 4079, 4121, 4122, 4123, 5053, 5054
<code>\zhdigits</code>	25	<code>\tl_set_eq:NN</code>	777, 778, 825, 911, 913, 957, 1050, 1075, 1533, 1545, 2097
<code>\zhnum</code>	80	<code>\tl_set_rescan:Nnn</code>	5179
<code>\zhnumber</code>	25, 25, 25, 25	<code>\tl_tail:N</code>	3039
<code>\zihao</code>	24, 24	<code>\tl_tail:n</code>	3049
<code>\ziju</code>	24	<code>\tl_to_str:n</code>	125, 126, 5122, 5151, 5192, 5193
<code>\zw</code>	74	<code>\tl_trim_spaces:n</code>	1562
tex commands:		<code>\today</code>	2222, 2227, 2230, 2235
<code>\tex_def:D</code>	5170	<code>today</code>	11, 2222
<code>\tex_divide:D</code>	78, 2130, 2171		
<code>\tex_edef:D</code>	5224		
<code>\tex_endlinechar:D</code>	544, 550		
<code>\tex_font:D</code>	469, 4799		
<code>\tex_iftrue:D</code>	3584		
<code>\tex_ignorespaces:D</code>	1056, 1215, 1220, 2021, 2155, 3675, 4981		
<code>\tex_immediate:D</code>	466		
<code>\tex_pagegoal:D</code>	2441		
<code>\tex_parskip:D</code>	2449		
<code>\tex_prevdepth:D</code>	2448		
<code>\tex_topskip:D</code>	2442		
<code>\the</code>	4468, 4471, 4472, 4491, 4495		
<code>\thechapter</code>	2659, 2662, 2710, 3194, 3217, 3385		

token commands:

`\token_get_replacement_spec:N` 5079, 5101
`\token_to_meaning:N` 5136
`\token_to_str:N`
 .. 53, 892, 901, 910, 912, 914, 1103, 1104, 1105, 1107,
 1114, 1115, 1116, 1178, 2272, 2273, 3683, 4806, 4966, 5038
`\toks` 4468, 4469, 4471, 4472, 4491, 4495
`\topsep` 3787, 3798, 3826, 3837
`\TrimSpaces` 2053
`\ttdefault` 1712
`\ttfamily` 529, 532, 1270, 1273, 1643
 two commands:
`\c_two_hundred_fifty_six` 560, 563
`\twocolumn` 2634
`\typeout` 2659, 2660, 3072

U

um commands:

`\um_input_math_symbol_table:` 739, 743
`\um_sym:nnn` 742
 um internal commands:
`__um_input_math_symbol_table:` 751
`__um_switchto_literal:` 748
`__um_sym:nnn` 750
`\undefined` 4493
`\unless` 2791, 2827, 2845
`\unskip` 2861
`\updefault` ... 953, 959, 961, 965, 967, 1317, 1319, 1322, 1326
`\upshape` 1658, 1661

use commands:

`\use:N` 779, 817, 818, 868, 910, 912, 914,
 968, 1016, 1546, 2465, 2471, 2480, 2912, 5206, 5226, 5240
`\use:n` 64, 122, 133, 388, 756, 867, 950, 1133,
 1186, 1192, 1286, 1290, 1388, 1562, 1723, 1876, 2431,
 2876, 2887, 3155, 3576, 3693, 3893, 5075, 5117, 5168, 5189
`\use:nn` 5234
`\use_i:nn` 108, 889, 2459, 5019, 5024, 5044, 5049
`\use_i:nnn` 261
`\use_ii:nn` ... 109, 889, 2457, 2461, 5019, 5024, 5044, 5049
`\use_ii:nnn` 134, 266
`\use_iii:nnn` 271
`\use_none:n` .. 189, 194, 199, 1033, 1289, 1423, 1877, 2208
`\use_none:nn` 482
`\usebeamercolor` 3230, 3248, 3266, 3368, 3428, 3462

`\usebeamerfont` 3230, 3239,
 3248, 3257, 3266, 3275, 3367, 3370, 3427, 3430, 3461, 3464
`\uselanguage` 3623, 3628, 3649, 3654
`\usepackage` 4966
 UTF8 6, 221

V

`\verse` 3872
`\vfil` 2514, 2621
`\vskip` 2551, 2554,
 2581, 2601, 2604, 2624, 2710, 2719, 2722, 2737, 2740,
 2847, 2850, 3231, 3249, 3267, 3351, 3369, 3388, 3429, 3463
`\vspace` 2517, 2701, 2704, 2725, 2728

W

`winfonts` 29
`\write` 4508

X

`\x` 4470, 4473,
 4474, 4475, 4476, 4477, 4478, 4479, 4480, 4481, 4482,
 4483, 4484, 4485, 4486, 4487, 4488, 4489, 4496, 4506, 4510
`\xdef` 4509
`\xeCJKsetup` 661, 1841, 1916, 2016, 2019, 2022, 2037
 xetex commands:
`\xetex_charglyph:D` 4803
`\xetex_glyphbounds:D` 4827

Y

`\yahei` 4460
`\youyuan` 4458

Z

`\zhdig` 2214
`\zhdigits` 25, 2221
`zhmap` 7, 29, 256
`\zhnum` 2213
 zhnum commands:
`\zhnum_counter:n` 2205
`\zhnumber` 25, 2219
`\zhnumsetup` 2231, 2236
`\zhtoday` 2230, 2235
`\zihao` 24, 3674, 3683
`zihao` 8, 158
`\ziju` 24, 2154
`\zw` 708, 1902, 1945, 2002