

CT_EX 宏集手册

CTEX.ORG

2016/10/25 v2.4.5*

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. 575f038.

第 1 节 介绍

历史

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

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

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

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

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

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

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

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

关于宏集名字の説明

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

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

¹<http://code.google.com/p/ctex-kit/>

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

³比如：pdf_TE_X 引擎下的 CJK、zhmCJK 宏包，X_YT_EX 引擎下的 xeCJK 宏包和 Lua_TE_X 引擎下的 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 CTeX 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 CTeX 宏集提供的四个中文文档类。

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

例 1

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

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

以下是使用 `ctexbeamer` 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

`\ctexset {⟨键值列表⟩}`

New: 2014-03-18

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

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

例 4

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

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

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

表 2 CT_EX 宏集的中文支持方式

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

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

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

不同的编译方式和中文支持方式会在一定程度上影响 $\text{CT}_{\text{E}}\text{X}$ 宏集的行为, 比如宏包对编码的处理。在用户使用 $\text{X}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 、 $\text{Lua}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 及 $\text{up}_{\text{A}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 $\text{up}_{\text{A}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 $\text{up}_{\text{A}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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
$\text{up}_{\text{A}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ + 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, 预设有 `chinese` 和 `plain` 两种方案。

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

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

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

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

Updated: 2014-04-11

设置标点处理格式。预定义好的格式有:

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

`banjiao` 半角式: 所有标点占半个汉字宽度;

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

`CCT` CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

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

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

plain 原样(不调整标点间距)。

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

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。

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

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

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

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

New: 2014-04-23

接受一个浮点数值,设置行距倍数。本选项的初始值与 scheme 有关。

- scheme = chinese 对标准文档类初始值为 1.3,即 1.3 倍行距。此时,相邻两行的基线($\backslash baselineskip$)距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。对 beamer 不改变行距,即使用默认的单倍行距。
- scheme = plain CTeX 宏集默认不调整行距倍数,文档中的行距由所选文档类和其他宏包或用户设置决定。

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

New: 2014-03-13

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

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

- true 等价于设置 autoindent = 2。
- false 禁用自动调整功能,可以设置固定长度的段首缩进。如设置每段缩进 40 点:

例 8

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

linestretch ☆ linestretch = $\langle \text{数值或长度} \rangle$

New: 2014-03-26

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

¹⁵LaTeX 将单个换行视作一个空格。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化,使之以中文的方式显示今天的日期。如编译本文档的日期就是“2016 年 10 月 25 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 10 月 25 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年十月二十五日”。使用全汉字的日期格式。
`old` 效果为“October 25, 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”。

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

6.3 页面格式设置与汉化

当使用了 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}}{}

appendix/numbering ★ numbering = true|false

New: 2015-06-21

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

用法与普通章节 numbering 选项类似。

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

Updated: 2014-03-08

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

用法与普通章节 name 选项类似。

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

表 20 appendix/name 选项的默认设置

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

appendix/number ★ number = {<数字输出命令>}

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

用法与普通章节的 number 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

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

表 21 appendix/number 选项的默认设置

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

我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 21

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

本例的设置效果大致如下:

熟悉 L^AT_EX

第 8 节 实用命令

8.1 字号与间距

`\zihao`

Updated: 2014-03-08

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

表 22 中文字号

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

`\ziju`

Updated: 2014-03-28

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

这个命令会影响 `\ccwd` 的值，但不会影响英文字距。

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

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

8.2 中文数字转换

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

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

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

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

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

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

以中文格式输出数字。这里的数字可以是整数、小数和分数。

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

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

将阿拉伯数字转换为中文数字串。

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

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

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

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

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

8.3 杂项

 $\backslash\text{CTeX}$

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

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

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

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

9.1 Lua \LaTeX 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:

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

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

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

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

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

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

10.1 修改宏包默认选项

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

例 23

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

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

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

10.2 宏包载入后的配置

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

例 24

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

例 25

```
% 略复杂的 ctex.cfg 内容示例: 禁止段末孤字成行。
% 在使用 XeTeX 编译时, 打开 xeCJK 的 CheckSingle 选项。
\sys_if_engine_xetex:T
{
  \xeCJKsetup { CheckSingle }
}
% 在使用 LuaTeX 编译时, 设置 LuaTeX-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}TeX 编译时的依赖项。
- LuaTeX-jan 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_{La}TeX 编译时的依赖项。
- zhmetrics-uptex 宏包。
- ➡ 以上是使用 up_{La}TeX 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 . \@currentext } { 9999/99/99 } }
5 <*class>
6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl
7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl
8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl
9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl
10 </class>
11 <*ctex>
12 \msg_new:nnnn { ctex } { subpackage-loaded }
13 { Package~\#1'~can~not~be~loaded~with~`ctex'. }
14 {
15   \#1'~is~actually~a~part~of~`ctex'.\\
16   It~is~not~necessary~to~load~it~separately.
17 }
18 \@ifpackageloaded { ctexsize }
19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }
20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }
21 \@ifpackageloaded { ctexheading }
22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }
23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }
24 </ctex>
25 </class|ctex>
26 <*class|style>
27 \RequirePackage { xparse , l3keys2e }
28 </class|style>
29 <*class|ctex>
```

检查 expl3 和 l3keys2e 的版本。

```
30 \msg_new:nnnn { ctex } { l3-too-old }
31 { Support~package~\#1'~too~old. }
32 {
33   Please~update~an~up~to~date~version~of~the~bundles\\
34   \l3kernel'~and~\l3packages'\\
35   using~your~TeX~package~manager~or~from~CTAN.
36 }
37 \@ifpackagelater { expl3 } { 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 }
<ltexheading> 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
<ltexheading> 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 ,

```

GBK
UTF8

```

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     adobe-fonts .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     win-fonts .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     no-fonts .value_forbidden:n = true ,
254     win-fonts .value_forbidden:n = true ,
255     adobe-fonts .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 用户设置接口

```

\ctexset 429 <*class|ctex|ctexheading>
430 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
431 </class|ctex|ctexheading>

```

\CTEXsetup
\CTEXoptions 过时命令。

```

432 <*class|ctex>
433 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
434 {
435   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
436   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
437   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
438 }
439 \NewDocumentCommand \CTEXoptions { +o }
440 {
441   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
442   { \ctexset~ {~ #1~ }~ is~ set. }
443   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
444 }
445 </class|ctex>

```

14.4 特定引擎支持与设置

14.4.1 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。

```

446 <pdfTeX|uptex|aptex>
447 \cs_new_protected_nopar:Npn \ctex_set_zhmap:n #1
448 {
449   \AtBeginDvi {#1}
450   \ctex_at_end_package:nn { atbegshi }
451   { \AtBeginShipoutFirst {#1} }
452 }
453 \onlypreamble \ctex_set_zhmap:n
454 <pdfTeX|uptex|aptex>
455 <pdfTeX>

```

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

```

456 \seq_new:N \c__ctex_cmap_encoding_seq
457 \seq_set_from_clist:Nn \c__ctex_cmap_encoding_seq
458 { C19 , C10 , C00 , C09 , C40 , C60 }

```

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

```

459 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2
460 {
461   \cs_if_free:cF { #1 + #2 }
462   {
463     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
464     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
465   }
466 }
467 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
468 \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。

```

469 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
470 {
471   \cs_if_free:NF \CJK@plane
472   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
473 }
474 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
475 {
476   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
477   #1
478 }
479 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
480 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
481 {
482   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
483   \tex_immediate:D \pdfTeX_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
484   \cs_new_protected_nopar:Npx #1
485   {
486     \exp_not:N \pdfTeX_pdffontattr:D \exp_not:N \tex_font:D
487     { /ToUnicode ~ \int_use:N \pdfTeX_pdflastobj:D \c_space_tl 0 ~ R }
488   }
489 }

```

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

```

490 \group_begin:
491 \char_set_catcode_other:N \#

```

```

492 \sys_if_output_pdf:TF
493 {
494   \group_end:
495   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
496     { \CTEX@Family@CMap {#1} {#2} }
497     {
498       \ctex_at_end_package:nn { cmap }
499       { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
500     }
501     { \ctex_patch_failure:N \DeclareFontFamily }
502   }
503 { \group_end: }

```

首先检查选项, 决定是否载入 zhmCJK 宏包。

```

504 \if_bool:N \g__ctex_zhmCJK_bool
505   \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
506   \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

507 \else:
508   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
509     { \RequirePackage { CJK } }
510     { \RequirePackage { CJKutf8 } }
511   \RequirePackage { CJKpunct , CJKspace }

```

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

```

512 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
513 {
514   \tl_set:Nn \CJKrmdefault {#1}
515   \tl_set:Nn \CJKsfdefault {#2}
516   \tl_set:Nn \CJKttdefault {#3}
517   \ctex_set_zhmap:n { \ctex_zhmap_input:n {#4} }
518 }
519 \@onlypreamble \ctex_load_zhmap:nnnn

```

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

```

520 \cs_new_protected_nopar:Npn \ctex_zhmap_input:n #1
521 {
522   \ctex_file_wrapper:nnn
523   {
524     \char_set_catcode_comment:n { 37 } % %
525     \char_set_catcode_escape:n { 92 } % \
526   }
527   { \file_input:n {#1} }
528   {
529     \char_set_catcode:nn { 37 } { \char_value_catcode:n { 37 } }
530     \char_set_catcode:nn { 92 } { \char_value_catcode:n { 92 } }
531   }
532 }

533 \tl_if_exist:NF \CJKfamilydefault
534 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
535 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
536 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
537 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
538 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
539 { \CJKfamily { \CJKrmdefault } }
540 { }
541 { \ctex_patch_failure:N \rmfamily }
542 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
543 { \CJKfamily { \CJKsfdefault } }
544 { }
545 { \ctex_patch_failure:N \sffamily }

```

```

546 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
547   { \CJKfamily { \CJKttdefault } }
548   { }
549   { \ctex_patch_failure:N \ttfamily }
550 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
551   { \CJKfamily { \CJKfamilydefault } }
552   { \cs_set_eq:NN \reset@font \normalfont }
553   { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```
554 \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。

```

555 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
556 {
557   \ctex_file_wrapper:nnn
558   {
559     \char_set_catcode_other:n { 60 } % <
560     \char_set_catcode_math_superscript:n { 94 } % ^
561     \int_set_eq:NN \tex_endlinechar:D \c_minus_one
562   }
563   { \file_input:n {#1} }
564   {
565     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
566     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
567     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
568   }
569 }
570 \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。

```

571 \if_cs_exist:N \CJK@surr
572 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
573 {
574   \tl_set:Nx \l__ctex_tmp_tl {#2}
575   \int_set:Nn \l__ctex_tmp_int
576     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
577   \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
578     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
579     {
580       \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
581       \tl_gset:Nx #1
582       {
583         \int_to_Hex:n
584         { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
585         \int_to_Hex:n
586         { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
587       }
588     }
589 }
590 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
591 \fi:

```

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

```

592 \AtBeginDocument
593 {

```

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

```

594 \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
595 { \punctstyle { \l__ctex_punct_tl } }
596 }

```

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

```

597 \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。

```

598 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
599 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
600 {
601   \CJK@makeActive
602   \CJK@loadBinding { standard }
603 }

```

在导言区结束时调用 `\CJK@envStart` 启用完整的中文功能。

`\CJK@envStart` 的定义是

```

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

```

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

```

604 \exp_args:Nx \ctex_at_end_preamble:n
605 {
606   \exp_not:N \CJK@envStart
607   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
608   \exp_not:N \CJKtilde
609 }

```

`zhmCJK` 判断结束。

```

610 \fi:

```

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

```

611 <@@=>
612 \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。

```

613 <@@=ctex>

```


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

```

614 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
615 {
616   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
617     { \cs_set_eq:NN \CJK@ignorespaces #1 }
618   \ctex_set_ignorespaces:
619 }
620 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
621 { \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。

```

622 \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。

```

623 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
624 {
625   \clist_map_inline:Nn \c__ctex_punct_family_clist
626   {
627     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
628     {
629       \cs_set_eq:cc
630       { CJKpunct@ ##1 @spaces }
631       { c__ctex_ #1 ##1 _punct_spaces_tl }
632     }
633   }
634 }
635 \clist_const:Nn \c__ctex_punct_family_clist
636 {
637   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
638   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
639 }

```

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

```

640 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
641 {
642   \cs_if_free:cF { CJKpunct@ #2 @spaces }
643   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
644 }

```

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

```

645 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2#3
646 {
647   \clist_map_inline:nn {#1}
648   {
649     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
650     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
651   }
652 }
653 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
654 {
655   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
656   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
657   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
658   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
659   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
660   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
661 }

```

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

```

662 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
663 {
664   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
665   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
666   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
667   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
668   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
669   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
670 }

```

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

```

671 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
672 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
673 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

674 \ctex_file_input:n { ctexspa.def }
675 </pdfTeX>

```

14.4.2 ctex-engine-xetex.def

```

676 <*xetex>
677 \RequirePackage { xeCJK }
678 \exp_args:Nx \xeCJKsetup
679 {
680   LoadFandol = false ,
681   PunctStyle = \l__ctex_punct_tl
682 }

```

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`。

```

683 \@ifpackagelater { fontspec } { 2014/05/25 } { }
684 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
685 </xetex>

```

14.4.3 ctex-engine-luatex.def

```

686 <*luatex>

```

LuaTeX-j_a 为了兼容 p_LAT_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。

```

687 \msg_new:nnn { ctex } { luatexja-loaded }
688 {
689   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
690   Loading~file~`#1'~will~abort!
691 }
692 \@ifpackageloaded { luatexja }
693 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
694 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
695 \RequirePackage { luatexja }
696 \@ifpackagelater { luatexja } { 2015/09/21 } { }
697 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
698 \RequirePackage { fontspec }
699 \@ifpackagelater { fontspec } { 2014/05/25 } { }
700 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

14.4.3.1 LuaTeX-j_a 的默认设置

```

701 \ExplSyntaxOff

```

以下设置抄录自 `l1tjdefs.sty`。

```

702 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
703 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
704 \ltjdefcharrange{3}{%
705   "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
706 \ltjdefcharrange{4}{%
707   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
708   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
709   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
710 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
711 \ltjdefcharrange{6}{%
712   "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
713   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
714 \ltjdefcharrange{7}{
715   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
716   "AC00-"D7FF}
717 \ltjdefcharrange{8}{ "A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
718 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, -8}}
719 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 `ltj-latex.sty`。

```

720 \directlua{
721   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
722   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
723 }
724 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
725   xkanjiskip=.25\zw plus 1pt minus 1pt,
726   autospacing, autoxspacing, jacharrange={-1},
727   yalbaselineshift=\z@, yjabaselineshift=\z@,
728   jcharwidowpenalty=500, differentjfm=paverage
729 }
730 \ExplSyntaxOn

```

14.4.3.2 LuaTeX-jā 的补丁

731 `<@@=ctex_l1tj>`

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`。

```

732 \RequirePackage { xunicode-addon }
733 \AtBeginUTFCommand
734 {
735   \group_begin:
736   \lua_now_x:n { tex.globaldefs = 0 }
737   \ltj@allalchar
738 }
739 \AtEndUTFCommand { \group_end: }

```

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

```

740 \cs_set_protected:Npn \fontspec_visible_space:
741 {
742   \font_glyph_if_exist:NnTF \font { "2423 }
743   { \ltjalchar "2423 \exp_stop_f: }
744   { \fontspec_visible_space_fallback: }
745 }

```

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。

```

746 \cs_new_protected:Npn \ctex_l1tj_set_math_letter:NN #1#2
747 {
748   \group_begin:
749   \cs_set_protected:Npn #1 ##1##2##3
750   { \ltjsetmathletter { ##1 } }

```

```

751     #2
752     \group_end:
753 }
754 \ctex_at_end_package:nn { unicode-math }
755 {
756     \cs_if_exist:NTF \um_input_math_symbol_table:
757     {
758         \ctex_ltj_set_math_letter:NN
759         \um_sym:nnn
760         \um_input_math_symbol_table:
761     }
762     {
763         \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
764         \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
765         { \__um_switchto_literal: }
766         \ctex_ltj_set_math_letter:NN
767         \__um_sym:nnn
768         \__um_input_math_symbol_table:
769     }
770 }

```

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

```

771 \ctex_at_end_package:nn { listings }
772 {
773     \use:x
774     {
775         \exp_not:N \RequirePackage { lltjp-listings }
776         \tl_set:Nn \exp_not:N \lstlistingname
777         { \exp_not:o { \lstlistingname } }
778         \tl_set:Nn \exp_not:N \lstlistlistingname
779         { \exp_not:o { \lstlistlistingname } }
780     }
781 }

```

14.4.3.3 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。

```

782 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
783 {
784     \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
785     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
786 }
787 \tl_new:N \CJK@family
788 \tl_new:N \l__ctex_ltj_current_font_tl
789 \tl_set:Nn \l__ctex_ltj_current_font_tl
790 { \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。

```

791 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
792 {
793     \group_begin:
794     \tl_set_eq:NN \f@encoding \CJK@encoding
795     \tl_set_eq:NN \f@family \CJK@family
796     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
797     \ctex_ltj_pickup_font:
798     \group_end:
799     \font@name
800     \__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。

```

801 \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
802 { \__ctex_ltj_select_font_aux: }
803 }
804 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
805 {
806   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
807   \cs_gset_nopar:Npx \font@name {#1}
808 }
809 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
810 { \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。

```

811 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
812 {
813   \exp_after:wN \cs_if_exist:NF \font@name
814   {
815     \group_begin:
816     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
817     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
818     \define@newfont
819     \group_end:
820   }
821 }
822 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

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

```

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

```

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

```

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

```

823 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
824 {
825   \get@external@font
826   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
827   {
828     \tl_set:Nx \external@font
829     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
830   }
831   \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。

```

832 \font@name
833 \lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
834 \use:c { \f@encoding + \f@family }
835 \use:c { \curr@fontshape }
836 }

```

`\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。

```
837 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
838   {
839     \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
840     {
841       \group_begin:
842       \tl_set_eq:NN \CJK@family \f@family
843       \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
844       {
845         \cs_gset_protected_nopar:Npx \subst@correction
846         {
847           \cs_new_eq:NN
848           \exp_not:c { \l__ctex_ltj_current_font_tl }
849           \font@name
850         }
851         \group_insert_after:N \group_insert_after:N
852         \group_insert_after:N \subst@correction
853       }
854       \group_end:
855     }
856   }
```

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

```
857 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
858   {
859     \lua_now_x:n { luatexja.jfont.does_alt_set ('lua_escape_x:n {#1}') }
860     \prg_return_true: \else: \prg_return_false: \fi:
861   }
```

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

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

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

```
862 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
863   { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }
```

`\ctex_ltj_select_alternate_font:` 在 `\selectfont` 中更新替代字体。

```
864 \cs_new_protected_nopar:Npn \ctex_ltj_select_alternate_font:
865   {
866     \ctex_ltj_if_alternate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
867     {
868       \lua_now_x:n
869       {
```

```

870         luatexja.jfont.output_alt_font_cmd
871         ('y', '\lua_escape_x:n { \l__ctex_ltj_current_shape_tl }')
872     }
873     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a ('f@size') }
874 }
875 }
876 \tl_new:N \l__ctex_ltj_current_shape_tl
877 \tl_set:Nn \l__ctex_ltj_current_shape_tl
878 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

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

```

879 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
880 {
881     \cs_if_exist:cF { #1/\f@size }
882     {
883         \group_begin:
884         \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
885         \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
886         \ctex_ltj_pickup_font:
887         \group_end:
888         \__ctex_ltj_pop_fontname:
889     }
890 }
891 <@@=>

```

`\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。

```

892 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
893 {
894     \ltj@@getjfontnumber #1
895     \lua_now_x:n
896     {
897         luatexja.jfont.pickup_alt_font_b
898         ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
899     }
900 }

```

14.4.3.4 数学字体族

```

901 <@@=ctex_ltj>

```

以下内容来自 `l1tjfont.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:

$$\backslash\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle$$

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

```

902 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
903 {
904     \lua_now_x:n
905     { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[^/]+') ) }
906     \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
907 }

```

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

```

908 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
909 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
910 \group_begin:
911 \char_set_catcode_other:N M

```



```

912 \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
913 { \ctex_ltj_if_jfont:nTF {#3} }
914 \group_end:
915 <@@=>

```

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`。

```

916 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
917 {
918   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
919   { \ctex_ltj_get_and_define_fonts_je:nN }
920   { \ctex_ltj_get_and_define_fonts_al:nN }
921   {#1} #2
922 }
923 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
924 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
925 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_je:nN #1#2
926 {
927   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
928   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
929   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
930   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
931   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
932   \ctex_ltj_pickup_font:
933   \tl_put_right:Nx \math@fonts
934   {
935     \ltj@setpar@global
936     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
937     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
938     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
939   }
940 }
941 <@@=-ctex_ltj>

```

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

```

942 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
943 {
944   \mode_if_math:T
945   {
946     \math@bgroup
947     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
948     \ctex_ltj_math_group_hook:
949     \ctex_ltj_if_jfont_math:NTF #1
950     { \jfam } { \mathgroup } #2 \scan_stop:
951     \math@egroup
952   }
953 }
954 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
955 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

14.4.3.5 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。

```

956 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
957 {
958   \str_if_eq:nnTF {#1} { plain }
959   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
960   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
961 }
962 \tl_new:N \l__ctex_ltj_jfm_tl
963 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
964 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

```

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

```

\CJK@encoding
 _ctex_ltj_change_encoding:

在 \LaTeX 下, \LuaTeX-j a 依赖字体编码来实现特殊设置。例如上述的 $\text{\ctex_ltj_if_jfont:nTF}$ 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。

```

965 \tl_const:Nn \CJK@encoding { LTJY3 }
966 \DeclareFontEncoding { \CJK@encoding } { } { }
967 \use:x
968 {
969   \exp_not:N \DeclareFontSubstitution
970   { \CJK@encoding } { song } { \mddefault } { \updefault }
971 }
972 \lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
973 \cs_new_protected_nopar:Npn \_ctex_ltj_change_encoding:
974 { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
975 \DeclareFontFamily { \CJK@encoding } { song } { }
976 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
977 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\_ctex_ltj_jfm_tl } { }
978 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
979 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\_ctex_ltj_jfm_tl } { }
980 \tl_const:Nn \c\_ctex_ltj_math_tl { CJKmath }
981 \DeclareSymbolFont { \c\_ctex_ltj_math_tl }
982 { \CJK@encoding } { song } { \mddefault } { \updefault }
983 \SetSymbolFont { \c\_ctex_ltj_math_tl } { bold }
984 { \CJK@encoding } { song } { \bfdefault } { \updefault }
985 \int_const:Nn \c\_ctex_ltj_math_fam_int { \use:c { sym \c\_ctex_ltj_math_tl } }
986 \jfam \c\_ctex_ltj_math_fam_int

```

这是 luatexja-fontspec 中新增的一些字体选项。

```

987 \newfontfeature { CID } { cid = #1 }
988 \newfontfeature { JFM } { jfm = #1 }
989 \newfontfeature { JFM-var } { jfmvar = #1 }

```

在新版本的 fontspec 中, $\text{_fontspec_namewrap:n}$ 变成了私有函数。

```

990 \keys_define:nn { fontspec-preparse-external }
991 {
992   NoEmbed .code:n =
993   { \cs_set_eq:NN \_fontspec_namewrap:n \_ctex_ltj_noembed_wrap:n }
994 }
995 \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。

```

996 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
997 {
998   \group_begin:
999   \clist_clear:N \l__ctex_ltj_char_range_clist
1000   \seq_clear:N \l__ctex_ltj_alternate_seq
1001   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1002   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1003   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1004   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1005   {
1006     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1007     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1008     {#1} { \l__ctex_ltj_font_options_clist }
1009     \_ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1010     \_ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1011     \_ctex_ltj_gset_family_cs:nn {#1} {#3}
1012   }
1013   \group_end:
1014 }
1015 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1016 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1017 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl

```

```

1018 \clist_new:N \l__ctex_ltj_font_options_clist
1019 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1020 {
1021   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1022   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1023 }

```

分别保存 `fontspec` 设置的字体族名、字体名称和字体选项。

```

1024 \prop_new:N \g__ctex_ltj_family_name_prop
1025 \prop_new:N \g__ctex_ltj_family_font_name_prop
1026 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

删除重复的定义,清除替代字体的先前设置。

```

1027 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1028 {
1029   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1030   {
1031     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1032     \cs_undefine:c { \__ctex_ltj_altername_cs:n {#1} }
1033     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1034     {
1035       \use:c { \__ctex_ltj_altername_cs:n { clear / #1 } }
1036       \cs_undefine:c { \__ctex_ltj_altername_cs:n { clear / #1 } }
1037       \cs_undefine:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1038       \prop_gremove:Nn \g__ctex_ltj_reset_altername_prop {#1}
1039     }
1040     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1041   }
1042 }
1043 \tl_new:N \l__ctex_ltj_tmp_tl
1044 \msg_new:nnn { ctex } { redefine-family }
1045 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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

```

1046 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1047 {
1048   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }
1049   {
1050     \group_begin:
1051     \__ctex_ltj_change_encoding:
1052     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1053     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1054     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1055     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1056     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1057     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1058     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1059     \__ctex_ltj_set_altername_family:n {#1}
1060     \group_end:
1061   }
1062 }
1063 \tl_new:N \l__ctex_ltj_base_family_tl
1064 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1065 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1066 \cs_new_protected_nopar:Npn \__ctex_ltj_set_altername_family:n #1
1067 {
1068   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1069   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1070   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1071   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n {#1} }
1072 }
1073 \cs_new:Npn \__ctex_ltj_altername_cs:n #1 { ctex_ltj/altername_family/#1 }

```

\CJKfamily 切换字体。

```

1074 \NewDocumentCommand \CJKfamily { m }
1075 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1076 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1077 {
1078   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1079   {
1080     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1081     \selectfont
1082   }
1083   { \__ctex_ltj_family_unknown_warning:n {#1} }
1084 }
1085 \tl_new:N \l_ctex_ltj_family_tl
1086 \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。

```

1087 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1088 {
1089   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1090   { \prg_return_true: }
1091   {
1092     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1093     {
1094       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1095       \prg_return_true:
1096     }
1097     { \prg_return_false: }
1098   }
1099 }
1100 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

\__ctex_ltj_family_unknown_warning:n 1101 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1102 {
1103   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1104   {
1105     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1106     {
1107       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1108       \msg_warning:nnn { ctex } { family-unknown } {#1}
1109     }
1110   }
1111 }
1112 \seq_new:N \g__ctex_ltj_unknown_family_seq
1113 \msg_new:nnn { ctex } { family-unknown }
1114 {
1115   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1116   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1117 }
1118 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1119 {
1120   \str_case:x:nnF {#1}
1121   {
1122     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1123     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1124     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1125   }
1126   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1127   [...] \{...\}
1128 }
1129 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1130 {
1131   \str_case:x:nnF {#1}
1132   {
1133     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1134     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1135     \CJKttdefault { \token_to_str:N \CJKttdefault }

```

```

1136     }
1137     {#1}
1138 }

\ctex_ltj_fontspec:nn 1139 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1140 {
1141     \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1142     { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1143     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1144     {
1145         \int_gincr:N \g__ctex_ltj_family_int
1146         \__ctex_ltj_fontspec:xnn
1147         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1148         {#1} {#2}
1149     }
1150 }
1151 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1152 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1153 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1154 {
1155     \bool_if:NT \l__ctex_ltj_add_alterate_bool
1156     {
1157         \cs_if_free:cF
1158         { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1159         {
1160             \cs_gset_eq:cc
1161             { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1162             { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1163             \cs_gset_eq:cc
1164             { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1165             { \__ctex_ltj_alterate_cs:n { clear / \l_ctex_ltj_family_tl } }
1166         }
1167         \bool_set_false:N \l__ctex_ltj_add_alterate_bool
1168     }
1169     \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1170     \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1171     \ctex_ltj_switch_family:n {#1}
1172 }
1173 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1174 \prop_new:N \g__ctex_ltj_fontspec_prop

\ctex_ltj_add_font_features:n 1175 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1176 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1177 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1178 {
1179     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1180     {#1} \l__ctex_ltj_tmp_tl
1181     {
1182         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1183         {#1} \l__ctex_ltj_font_options_clist
1184         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1185         \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1186         \ctex_ltj_fontspec:xx
1187         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1188         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1189     }
1190     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1191 }
1192 \bool_new:N \l__ctex_ltj_add_alterate_bool
1193 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1194 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1195 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1196 {
1197     \token_to_str:N \addCJKfontfeature (s)~ignored.\
1198     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1199 }

```

```

\setCJKfamilyfont 1200 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
\newCJKfontfamily 1201 { \ctex_ljt_set_family:xxx {#1} {#2} {#3} }
\CJKfontspec 1202 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
\addCJKfontfeatures 1203 {
1204   \tl_set:Nx \l__ctex_ljt_tmp_tl
1205   { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1206   \cs_new_protected_nopar:Npx #2
1207   { \ctex_ljt_switch_family:n { \l__ctex_ljt_tmp_tl } }
1208   \ctex_ljt_set_family:xxx { \l__ctex_ljt_tmp_tl } {#3} {#4}
1209 }
1210 \NewDocumentCommand \CJKfontspec { O { } m }
1211 {
1212   \ctex_ljt_fontspec:xx {#1} {#2}
1213   \tex_ignorespaces:D
1214 }
1215 \NewDocumentCommand \addCJKfontfeatures { m }
1216 {
1217   \ctex_ljt_add_font_features:x {#1}
1218   \tex_ignorespaces:D
1219 }
1220 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\setCJKmainfont 1221 \NewDocumentCommand \setCJKmainfont { O { } m }
\setCJKsansfont 1222 {
\setCJKmonofont 1223   \ctex_ljt_set_family:xxx { \CJKrmdefault } {#1} {#2}
\setCJKmathfont 1224   \normalfont
\defaultCJKfontfeatures 1225 }
1226 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1227 \NewDocumentCommand \setCJKsansfont { O { } m }
1228 {
1229   \ctex_ljt_set_family:xxx { \CJKsfdefault } {#1} {#2}
1230   \normalfont
1231 }
1232 \NewDocumentCommand \setCJKmonofont { O { } m }
1233 {
1234   \ctex_ljt_set_family:xxx { \CJKttdefault } {#1} {#2}
1235   \normalfont
1236 }
1237 \NewDocumentCommand \setCJKmathfont { O { } m }
1238 { \ctex_ljt_set_family:xxx { \c__ctex_ljt_math_tl } {#1} {#2} }
1239 \NewDocumentCommand \defaultCJKfontfeatures { m }
1240 { \clist_gset:Nn \g__ctex_ljt_default_features_clist {#1} }
1241 \clist_new:N \g__ctex_ljt_default_features_clist
1242 \@onlypreamble \setCJKmainfont
1243 \@onlypreamble \setCJKsansfont
1244 \@onlypreamble \setCJKmonofont
1245 \@onlypreamble \setCJKmathfont
1246 \@onlypreamble \setCJKromanfont
1247 \@onlypreamble \defaultCJKfontfeatures

1248 \tl_if_exist:NF \CJKfamilydefault
1249 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1250 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1251 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1252 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1253 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1254 { \CJKfamily { \CJKrmdefault } }
1255 { }
1256 { \ctex_patch_failure:N \rmfamily }
1257 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1258 { \CJKfamily { \CJKsfdefault } }
1259 { }
1260 { \ctex_patch_failure:N \sffamily }
1261 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1262 { \CJKfamily { \CJKttdefault } }
1263 { }
1264 { \ctex_patch_failure:N \ttfamily }
1265 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }

```

```

1266 { \CJKfamily { \CJKfamilydefault } }
1267 { \cs_set_eq:NN \reset@font \normalfont }
1268 { \ctex_patch_failure:N \normalfont }

```

在导言区末尾更新 \CJKfamilydefault。

```

1269 \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。

```

1270 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1271 {
1272   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1273   {
1274     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1275     {
1276       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1277       { \use:n }
1278       {
1279         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1280         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1281         { \use:n }
1282       }
1283     }
1284     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1285     {
1286       \prop_map_break:n
1287       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1288     }
1289   }
1290   \normalfont
1291   \ctex_ltj_update_mathfont:
1292 }
1293 }
1294 }

```

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

```

1295 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1296 {
1297   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1298   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1299   {
1300     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1301     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1302   }
1303 }
1304 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1305 {
1306   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1307   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1308   { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1309   \cs_if_free:cTF
1310   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1311   {
1312     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1313     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1314   }
1315   {
1316     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1317     { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1318   }
1319 }

```


14.4.3.6 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1320 \keys_define:nn { ctex_ltj / fontspec }
1321 {
1322   AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1323   AlternateFont .value_required:n = true ,
1324   CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1325   CharRange .value_required:n = true
1326 }
```

\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。

```

1327 \group_begin:
1328   \char_set_catcode_other:N \|
1329   \cs_set:Npn \__ctex_ltj_tmp:w #1
1330   {
1331     \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1332     {
1333       \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1334       {
1335         \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1336         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1337         \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1338         \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1339         { ! \tl_if_blank_p:n { ###1 } }
1340         \seq_concat:NNN \l__ctex_ltj_alternate_seq
1341         \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1342       }
1343     }
1344   }
1345   \char_set_catcode_active:N \|
1346   \__ctex_ltj_tmp:w { || }
1347 \group_end:
1348 \seq_new:N \l__ctex_ltj_tmp_seq
1349 \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。

```

1350 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1351 {
1352   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1353   {
1354     \__ctex_ltj_check_family:n {#1}
1355     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1356     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1357     #3
1358   }
1359   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1360 }
```

\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

```

1361 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1362 {
1363   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1364   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1365 }
1366 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1367 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1368 { m m m +O{ } m u{ \q_stop } }
1369 {
1370   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1371   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1372   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
```

```

1373 \tl_if_blank:nTF {#5}
1374 { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1375 {
1376 \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1377 \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1378 }
1379 \use:x
1380 {
1381 \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1382 { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1383 { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1384 { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1385 }
1386 }
1387 \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。

```

1388 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1389 {
1390 \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1391 \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1392 \ctex_ltj_set_alternate_family:coonn
1393 { \__ctex_ltj_alternate_cs:n {#1} }
1394 { \l__ctex_ltj_char_range_clist }
1395 { \l__ctex_ltj_font_options_clist } {#2} {#1}
1396 }
1397 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1398 {
1399 \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1400 { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1401 \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1402 }
1403 \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。

```

1404 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1405 {
1406 \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1407 \cs_gset_protected_nopar:Npx #1
1408 { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1409 }

```

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

```

1410 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1411 {
1412 \group_begin:
1413 \__ctex_ltj_change_encoding:
1414 \cs_set_eq:NN \CJKfamily \use_none:n
1415 \ctex_ltj_swap_cs:NN
1416 \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1417 \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1418 \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1419 \group_end:
1420 }
1421 \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。

```

1422 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1423 {
1424 \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1425 \cs_set_eq:NN #1 #2

```

```

1426 \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1427 \cs_undefine:N \__ctex_ltj_tmp:w
1428 }

```

LTJFONTUID

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。

```

1429 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1430 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1431 {
1432   \int_gincr:N \g__ctex_ltj_family_int
1433   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1434 }
1435 \int_new:N \g__ctex_ltj_family_int

```

\ctex_ltj_declare_altername_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>*。

```

1436 \cs_new_protected:Npn \ctex_ltj_declare_altername_shape:nnnnnn #1#2#3#4#5#6
1437 {
1438   \ctex_ltj_declare_altername_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1439   \ctex_ltj_set_altername_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1440   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1441   { \l_fontspec_family_tl } {#3} {#4}
1442 }

```

\ctex_ltj_set_altername_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。

```

1443 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:Nnnnnnn #1#2#3#4#5#6#7
1444 {
1445   \clist_map_inline:Nn #1
1446   {
1447     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1448     {
1449       \ctex_ltj_set_altername_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1450       \l__ctex_ltj_char_range_tl
1451     }
1452     { \ctex_ltj_set_altername_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1453   }
1454   \__ctex_ltj_save_altername_shape:cnn
1455   { \__ctex_ltj_altername_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1456   { luatexja.jfont.clear_alt_font_latex }
1457   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1458 }

```

\ctex_ltj_set_altername_shape:nnn

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

```

1459 \NewDocumentCommand \ctex_ltj_set_altername_shape:nnn
1460 { m m > { \SplitArgument { \c_one } { -> } } m }
1461 { \ctex_ltj_set_altername_shape:nnnn {#1} {#2} #3 }
1462 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:nnnn #1#2#3#4
1463 {
1464   \ctex_ltj_set_altername_shape:n
1465   {
1466     \IfNoValueTF {#4}
1467     { \int_eval:n {#3} , \int_eval:n {#3} , }
1468     {
1469       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1470       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1471     }

```

```

1472         '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1473         '\lua_escape_x:n { \CJK@encoding/#1 }'
1474     }
1475 }
1476 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1477 {
1478     \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1479     \__ctex_ltj_save_alternate_shape:cnn
1480     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1481     { luatexja.jfont.set_alt_font_latex } {#1}
1482 }

```

\ctex_ltj_set_alternate_shape:nnN

若字符范围预先由 declarecharrange 声明,则可以直接使用。

```

1483 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1484 {
1485     \tl_map_inline:Nn #3
1486     {
1487         \ctex_ltj_set_alternate_shape:n
1488         {
1489             ##1 ,
1490             '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1491             '\lua_escape_x:n { \CJK@encoding/#1 }'
1492         }
1493     }
1494 }

```

__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。

```

1495 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1496 {
1497     \group_begin:
1498     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1499     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1500     \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1501     \cs_gset_protected_nopar:Npx #1
1502     { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1503     \group_end:
1504 }
1505 \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。

```

1506 \keys_define:nn { ctex }
1507 {
1508     clearalternatefont .code:n =
1509     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1510     resetalternatefont .code:n =
1511     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1512     clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1513     resetalternatefont .default:n = \l_ctex_ltj_family_tl
1514 }
1515 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1516 {
1517     \group_begin:
1518     \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1519     {
1520         \cs_if_exist_use:cT { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1521         {
1522             \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1523             {#1} { \l__ctex_ltj_base_family_tl }
1524             \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1525             \selectfont
1526         }
1527     }

```

```

1527     }
1528     { \_ctex_ltj_family_unknown_warning:n {#1} }
1529   \group_end:
1530 }
1531 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1532 {
1533   \group_begin:
1534   \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1535   {
1536     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1537     \use:c { \_ctex_ltj_alternate_cs:n { reset / #1 } }
1538     \selectfont
1539   }
1540   \group_end:
1541 }
1542 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1543 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

declarecharrange 预先声明字符范围。

```

1544 \keys_define:nn { ctex }
1545 {
1546   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1547   declarecharrange .value_required:n = true
1548 }
1549 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1550 { \clist_map_inline:nn {#1} { \_ctex_ltj_declare_char_range:nn ##1 } }
1551 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1552 \cs_new_protected_nopar:Npn \_ctex_ltj_declare_char_range:nn #1#2
1553 { \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
1554 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1555 {
1556   \tl_clear:N \l__ctex_ltj_char_range_tl
1557   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1558   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1559   \ctex_ltj_def_char_range_key:n {#1}
1560   \tl_clear:N \l__ctex_ltj_char_range_tl
1561 }
1562 \tl_new:N \l__ctex_ltj_char_range_tl
1563 \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。

```

1564 \NewDocumentCommand \ctex_ltj_save_char_range:n
1565 { > { \SplitArgument { \c_one } { -> } } m }
1566 { \ctex_ltj_save_char_range:nn #1 }
1567 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1568 {
1569   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1570   { {
1571     \IfNoValueTF {#2}
1572     { \int_eval:n {#1} , \int_eval:n {#1} }
1573     {
1574       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1575       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1576     }
1577   } }
1578 }

```

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

```

1579 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1580 {
1581   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}

```

```

1582     {
1583       \keys_define:nn { ctex_ltj / fontspec }
1584       { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1585     }
1586   }

```

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

```

1587 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1588 {
1589   \tl_if_blank:nTF {#2}
1590   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1591   {
1592     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1593     {
1594       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1595       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1596     }
1597   }
1598 }

```

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

```

1599 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1600 { +o o u { \q_stop } }
1601 {
1602   \exp_args:Nnf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1603   {
1604     \IfNoValueTF {#1} { {#3} }
1605     {
1606       \IfNoValueTF {#2}
1607       { \tl_if_blank:nTF {#3} { { {#1} } } { [ {#1} ] {#3} } }
1608       { [ {#1} ] { {#2} } }
1609     }
1610   }
1611   \seq_put_right:No \l__ctex_ltj_alterate_seq { \l__ctex_ltj_tmp_tl }
1612 }

```

14.4.3.7 其它设置

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

```

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

1613 \AtBeginDocument
1614 {
1615   \ctex_appto_cmd:NnnTF \verbatim@font
1616   { \char_set_catcode_letter:n { 64 } }
1617   { \CTEX@verbatim@font@hook }
1618   { }
1619   { \ctex_patch_failure:N \verbatim@font }

```

```

1620 }
1621 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1622 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1623 <@@= >

```

\@@italiccorr L^AT_EX 的倾斜校正也要重新定义。

```

1624 \cs_set_eq:NN \@@italiccorr \/
1625 <@@=ctex >
1626 </luatex >

```

14.4.4 ctex-engine-uptex.def

```
1627 <*uptex|aptex >
```

按 CJK 的命名习惯模拟部分命令,并设置默认字体。

```

1628 \tl_set:Nn \CJKrmdefault { rm }
1629 \tl_set:Nn \CJKsfdefault { sf }
1630 \tl_set:Nn \CJKttdefault { tt }
1631 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1632 \tl_put_left:Nn \rmfamily { \kanjifamily { \CJKrmdefault } }
1633 \tl_put_left:Nn \sffamily { \kanjifamily { \CJKsfdefault } }
1634 \tl_put_left:Nn \ttfamily { \kanjifamily { \CJKttdefault } }
1635 \NewDocumentCommand \CJKfamily { m }
1636 {
1637   \kanjifamily { #1 }
1638   \selectfont
1639 }
1640 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```

在导言区末尾更新 \CJKfamilydefault。

```
1641 \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 2016/05/07 的定义有所变化,这一行为可以由用户通过 platexrelease 包改变,需要分支处理。

```

1642 \ctex_patch_cmd_once:NnnTF \em
1643 { \ExplSyntaxOff }
1644 { \eminnershape \else \gtfamily \itshape }
1645 { \eminnershape \else \itshape }
1646 { }
1647 {
1648   \ctex_patch_cmd:Nnn \em
1649   { \mcfamily \upshape \else \gtfamily \itshape }
1650   { \eminnershape \else \itshape }
1651 }
1652 \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。

```

1653 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1654 {
1655   \DeclareKanjiFamily{JY2}{#1}{}
1656   \DeclareKanjiFamily{JT2}{#1}{}
1657   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1658   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1659   \tl_if_empty:nF { #3 }
1660   {

```

```

1661         \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{ }
1662         \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{ }
1663     }
1664 }
1665 \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。

```

1666 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1667 {
1668     \ctex_set_zhmap:n
1669     {
1670         \special{ pdf:mapline~ #1-h~    UniGB-UTF16-H~ #2 }
1671         \special{ pdf:mapline~ #1-v~    UniGB-UTF16-V~ #2 }
1672         \tl_if_empty:nF { #3 }
1673         {
1674             \special{ pdf:mapline~ #1b-h~  UniGB-UTF16-H~ #3 }
1675             \special{ pdf:mapline~ #1b-v~  UniGB-UTF16-V~ #3 }
1676         }
1677     }
1678 }
1679 \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。

```

1680 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1681 {
1682     \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1683     \ctex_set_upmap:nnn { upserifit } { #3 } { }
1684     \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1685     \ctex_set_upmap:nnn { upmono } { #6 } { }
1686 }
1687 \onlypreamble \ctex_set_upfonts:nnnnnn

```

`\@EverySelectfont@Init` `everyysel` 宏包 (2011/10/28) 未考虑 upLaTeX 对 `\selectfont` 的修改，因此这里需要对其给出正确的定义。因为不会使用 `CJK`，这里忽略有关 `CJK` 的定义。

```

1688 \cs_set_nopar:Npn \@EverySelectfont@Init
1689 {
1690     \ctex_patch_cmd:Nnn \selectfont
1691     { \enc@update }
1692     {
1693         \@EverySelectfont@EveryHook
1694         \@EverySelectfont@AtNextHook
1695         \gdef \@EverySelectfont@AtNextHook { }
1696         \enc@update
1697     }
1698     \ifpackageloaded { tracefnt }
1699     {
1700         \EverySelectfont
1701         {
1702             \ifnum \tracingfonts > \tw@
1703                 \font@info { Switching~ to~ \font@name }
1704             \fi
1705         }
1706     }
1707     { }
1708     \let \@EverySelectfont@Init \undefined
1709 }
1710 </uptex|aptex>

```


14.4.5 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。

```

1711 <*/pdf|luatex|uptex|aptex>
1712 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1713 {
1714   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1715   {
1716     \group_begin:
1717     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1718     \tl_gset:Nx \CJKfamilydefault
1719     {
1720       \str_case:onF { \familydefault }
1721       {
1722         { \rmdefault } { \exp_not:N \CJKrmdefault }
1723         { \sfdefault } { \exp_not:N \CJKsfdefault }
1724         { \ttdefault } { \exp_not:N \CJKttdefault }
1725       }
1726       { \CJKfamilydefault }
1727     }
1728     \group_end:
1729   }

```

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

```

1730 <*/luatex>
1731   \ctex_ltj_ensure_default_family:
1732 </luatex>
1733 }

```

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

```

1734 \tl_new:N \l__ctex_family_default_init_tl
1735 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1736 \tl_set:Nx \l__ctex_family_default_init_tl
1737 {
1738   \exp_not:N \__ctex_family_default_wrap:n
1739   { \exp_not:o { \CJKfamilydefault } }
1740 }
1741 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1742 </pdf|luatex|uptex|aptex>

```

14.4.6 操作系统的判断

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

```

1743 <*/luatex>
1744 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1745 {
1746   \tl_gset:Nx \g__ctex_fontset_tl
1747   {
1748     \lua_now_x:n
1749     {
1750       if ~ os.name == 'windows' then ~
1751         tex.sprint ( 'windows' )
1752       elseif ~ os.name == 'macosx' then ~
1753         tex.sprint ( 'mac' )
1754       else ~
1755         tex.sprint ( 'fandol' )
1756       end
1757     }
1758   }
1759 }

```

1760 `</luatex>`

ApTeX 可以使用 `\ngostype` 来判断。

1761 `<*aptex>`

1762 `\cs_new_protected_nopar:Npn \ctex_detected_platform:`

```
1763 {
1764   \tl_gset:Nx \g__ctex_fontset_tl
1765   {
1766     \str_case:onF { \ngostype }
1767     {
1768       { Win32 } { windows }
1769       { Win64 } { windows }
1770       { Darwin } { mac }
1771     }
1772     { fandol }
1773   }
1774 }
1775 </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!

1776 `<*xetex|pdfx|uptex>`

1777 `\cs_new_protected_nopar:Npn \ctex_detected_platform:`

```
1778 {
1779   \file_if_exist:nTF { /dev/null }
1780   {
1781     \file_if_exist:nTF { nul: }
1782     {
1783       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1784       { \ctex_if_macosx:TF { mac } { windows } }
1785       { \ctex_if_macosx:TF { mac } { fandol } }
1786     }
1787     { \ctex_if_macosx:TF { mac } { fandol } }
1788   }
1789   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1790 }
1791 \str_const:Nx \c__ctex_upper_case_file_str
1792 { \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。

1793 `\cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2`

```
1794 {
1795   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1796   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1797   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1798 }
1799 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1800 </xetex|pdfx|uptex>
```

14.4.7 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。

1801 `<*class|ctex>`

1802 `\@ifpackageloaded { hyperref }`

```

1803 {
1804   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1805     { \hypersetup {#1} }
1806 }
1807 {
1808   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1809     { \PassOptionsToPackage {#1} { hyperref } }
1810 }
1811 </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。

```

1812 <*pdfTeX>
1813 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1814 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1815 {
1816   \ctex_hypersetup:n { CJKbookmarks = true }
1817   \sys_if_output_pdf:TF
1818     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1819     {
1820       \ctex_at_end_package:nn { hyperref }
1821       {
1822         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1823         {
1824           \AtBeginShipoutFirst
1825             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1826         }
1827         { \RequirePackage { xCJK2uni } }
1828       }
1829     }
1830 }
1831 { \ctex_hypersetup:n { pdfencoding = unicode } }
1832 </pdfTeX>

```

在 XeTeX 下, hyperref 在处理带有非 ASCII 字符和 \% 的书签时有问题¹⁷。事实上, hyperref 在驱动文件 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@if_undefined:StringEncodingConvertTest{%
    }{%
      \EdefUnescapeString\HyPsd@temp#1%
      \ifxetex
        \let\HyPsd@UnescapedString\HyPsd@temp
        \StringEncodingConvertTest\HyPsd@temp\HyPsd@temp
          {utf16be}{ascii-print}{%
          \EdefEscapeString\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:

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

避免 XDVIPDFMX 的警告¹⁸:

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

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

```
1833 <*xetex|luatex>
1834 \ctex_hypersetup:n { pdfencoding = unicode }
1835 </xetex|luatex>
```

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

```
1836 <*uptex|aptex>
1837 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1838 \ctex_at_end_package:nn { hyperref }
1839 { \AtBeginShipoutFirst { \special { pdf:tounicode~ UTF8-UTF16 } } }
1840 </uptex|aptex>

1841 <*pdftex|xetex|luatex|uptex|aptex>
```

14.4.8 CJKfntef、xeCJKfntef 相关设置

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

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

```
1842 <*pdftex>
1843 \RequirePackage { CJKfntef }
1844 \normalem
1845 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1846 { \tl_clear:c { CJK#1color } }
1847 </pdftex>
1848 <*xetex>
1849 \RequirePackage { xeCJKfntef }
1850 \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
1851 {
1852   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1853   { \xeCJKsetup { #1 / format = { } } }
1854 }
1855 {
1856   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1857   { \tl_clear:c { CJK#1color } }
1858 }
1859 </xetex>
1860 <*luatex|uptex|aptex>
1861 \msg_new:nnn { ctex } { fntef-not-available }
<luatex> 1862 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
<uptex> 1863 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
<aptex> 1864 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1865 \msg_warning:nn { ctex } { fntef-not-available }
1866 </luatex|uptex|aptex>
1867 \clist_map_inline:nn
1868 { underdot , underline , underdblline , underwave , sout , xout }
1869 <*pdftex|xetex>
1870 {
```

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

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

```

1871 \__ctex_clear_fntef_color:n {#1}
1872 \cs_new_protected_nopar:cpx { CTEX#1 }
1873 {
1874   \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1875   { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1876   \exp_not:c { CJK#1 }
1877 }
1878 }
1879 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1880 {
1881   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1882   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1883   \CJKfilltwosides
1884 }
1885 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1886 </pdfTeX|xetex>
1887 <*/luatex|uptex|aptex>
1888 { \cs_new_eq:cN { CTEX#1 } \use:n }
1889 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1890 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1891 </luatex|uptex|aptex>
1892 <*/pdfTeX>
1893 \clist_map_inline:nn
1894 {
1895   underdotbasesep , underdotsep , underlinebasesep ,
1896   underlinesep , underdbllinesep , underdbllinebasesep ,
1897   underwavebasesep , underwavesep , southeight ,
1898   underdotcolor , underwavecolor , underlinecolor ,
1899   underdbllinecolor , soutcolor , xoutcolor
1900 }
1901 {
1902   \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1903   \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1904 }
1905 </pdfTeX>

```

14.4.9 \ccwd 的更新

```

\ctex_update_ccwd: 1906 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
\ccwd 1907 <*/pdfTeX|xetex>
1908 {
1909   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1910   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1911 }
1912 </pdfTeX|xetex>
1913 <*/luatex>
1914 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1915 </luatex>
1916 <*/uptex|aptex>
1917 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1918 </uptex|aptex>
1919 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1920 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1921 <*/pdfTeX>
1922 {
1923   \cs_set_protected_nopar:Npn \CJKglue
1924   { \skip_horizontal:N \l__ctex_ccglue_skip }
1925 }
1926 </pdfTeX>
1927 <*/xetex>
1928 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1929 </xetex>
1930 <*/luatex>

```

```

1931 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1932 </luatex>
1933 <*<u>ptex</u>|aptex>
1934 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1935 </u>ptex|aptex>
1936 \skip_new:N \l__ctex_ccglue_skip

```

检查用户是否修改过汉字间距。

\ctex_if_ccglue_touched_p:
\ctex_if_ccglue_touched:TF

```

1937 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1938 {
1939 <*<pdf<u>tex</u>|xetex>
1940   \if_meaning:w \CJKglue \__ctex_ccglue:
1941   \prg_return_false: \else: \prg_return_true: \fi:
1942 </pdf<u>tex</u>|xetex>
1943 <*<u>luatex</u>
1944   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1945   { \prg_return_false: } { \prg_return_true: }
1946 </luatex>
1947 <*<u>ptex</u>|aptex>
1948   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1949   { \prg_return_false: } { \prg_return_true: }
1950 </u>ptex|aptex>
1951 }

```

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

```

1952 <*<pdf<u>tex</u>|xetex>
1953 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1954 </pdf<u>tex</u>|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 pdfT_EX 和 X_YT_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_YT_EX 可以用 \fontcharwd 来改进。而在 pdfT_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 LuaT_EX 下, LuaT_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。upT_EX 可以直接使用原生的长度单位 zw。

```

1955 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
<pdf<u>tex</u>|xetex> 1956 { \dim_set:Nn \ccwd { \f@size \p@ } }
<u>luatex</u> 1957 { \dim_set:Nn \ccwd { \zw } }
<u>ptex</u>|aptex> 1958 { \dim_set:Nn \ccwd { \zw } }

```

14.4.10 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 LuaT_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。everyysel 宏包会用 \CheckCommand 来检查 \selectfont 是否为标准定义。我们修改了 \selectfont, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。

```

1959 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1960 {
1961   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1962   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1963 }
1964 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1965 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1966 \ctex_preto_cmd:NnnTF \selectfont { \ExplSyntaxOff }
1967 { \CTEX@selectfont@hook }
1968 {
1969   \tl_put_left:Nn \@EverySelectfont@Init
1970   { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }

```

```

1971 }
1972 { \ctex_patch_failure:N \selectfont }

```

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

```

1988 <*\luatex|uptex|aptex>
1989 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
1990 {
1991     \skip_if_eq:nnT
1992     { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
1993     { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
1994     {
1995         \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
1996         \ltjsetxkanjiskip \l__ctex_xkanjiskip_skip
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
2002 { .25\zw plus 1pt minus 1pt }
2003 { .25zw plus 1pt minus 1pt }
2004 \skip_new:N \l__ctex_xkanjiskip_skip
2005 \skip_set:Nn \l__ctex_xkanjiskip_skip
2006 { \ltjgetparameter { xkanjiskip } }
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 =
2015     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
2016 <*\xetex>
2017     { \xeCJKsetup { CJKspace = true } } ,
2018     space / auto .code:n =

```

```

2018 { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
2019 { \xeCJKsetup { CJKspace = false } } ,
2020 space / false .code:n =
2021 { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
2022 { \xeCJKsetup { CJKspace = false } } ,
2023 space .default:n = { true } ,
2024 space .initial:n = { auto }
2025 </pdf|tex|xetex>
2026 <*|uatex|uptex|aptex>
2027 space .code:n =
2028 { \msg_warning:nn { ctex } { invalid-option } }
2029 </uatex|uptex|aptex>
2030 }

```

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-j_a 设置的是字体的默认 JFM, 只会影响到之后设置的字体。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_qLaTeX、LuaLaTeX 和 upLaTeX 总是使用 UTF8 编码。

```

2043 <*xetex|uatex|uptex|aptex>
2044 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2045 </xetex|uatex|uptex|aptex>
2046 </pdf|tex|xetex|uatex|uptex|aptex>
2047 <*class|ctex>

```

14.4.11 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 }

```

14.5 字距与缩进

autoindent 也是可以用在正文中的选项, 意义与宏包选项 option/autoindent 相同。

```

2049 \keys_define:nn { ctex }
2050 {
2051   autoindent .choice: ,
2052   autoindent .default:n = { true } ,
2053   autoindent / true .code:n =
2054   {
2055     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2056     \ctex_select_size:
2057   } ,
2058   autoindent / false .code:n =
2059   { \tl_clear:N \l__ctex_autoindent_tl } ,
2060   autoindent / unknown .code:n =
2061   {
2062     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2063     \ctex_select_size:

```



```

2064     }
2065 }

```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2066 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2067 \cs_new_protected_nopar:Npn \ctex_select_size:
2068 { \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。

```

2069 \cs_new_protected_nopar:Npn \ctex_update_size:
2070 {
2071     \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2072     {
2073         \ctex_update_stretch:
2074         \ctex_update_parindent:
2075     }
2076     { \ctex_update_ziju: }
2077 }
2078 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2079 \tl_new:N \l__ctex_ziju_tl
2080 \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。

```

2081 \ctex_add_to_selectfont:n
2082 { \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`。

```

2083 \keys_define:nn { ctex }
2084 {
2085     linestretch .code:n =
2086     {
2087         \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2088         \ctex_select_size:
2089     } ,
2090     linestretch .value_required:n = true
2091 }
2092 \tl_new:N \l__ctex_line_stretch_tl
2093 \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。

```

2094 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2095 {
2096     \ctex_update_em_unit:
2097     \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2098     \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2099     {
2100         \ctex_if_ccglue_touched:TF
2101         { \ctex_update_ccwd: }
2102         {
2103             \dim_set:Nn \l__ctex_tmp_dim

```

```

2104         { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2105         \skip_set:Nn \l__ctex_ccglue_skip
2106         { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2107         \ctex_update_ccglue:
2108     }
2109 }
2110 {
2111     \int_set:Nn \l__ctex_tmp_int
2112     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2113     \tex_divide:D \l__ctex_tmp_int \ccwd
2114     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2115     {
2116         \skip_set:Nn \l__ctex_ccglue_skip
2117         {
2118             \c_zero_dim plus \dim_eval:n
2119             {
2120                 ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2121                 \l__ctex_tmp_int
2122             }
2123         }
2124     }
2125     { \skip_zero:N \l__ctex_ccglue_skip }
2126     \ctex_update_ccglue:
2127 }
2128 }

```

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

```

2129 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2130 {
2131     \tl_if_empty:NF \l__ctex_autoindent_tl
2132     {
2133         \dim_compare:nNnF \parindent = \c_zero_dim
2134         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2135     }
2136 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2137 \NewDocumentCommand \ziju { m }
2138 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2139 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2140 {
2141     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2142     \ctex_select_size:
2143 }

```

`\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。

```

2144 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2145 {
2146     \ctex_update_em_unit:
2147     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2148     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2149     \dim_compare:nNnTF \ccwd > \c_zero_dim

```

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

```

2150     {
2151         \dim_set:Nn \l__ctex_tmp_dim
2152         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2153         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2154         \tex_divide:D \l__ctex_tmp_int \ccwd
2155         \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。

```

2156      \dim_compare:nNnF \parindent = \c_zero_dim
2157      {
2158          \int_compare:nNnF \l__ctex_tmp_int < \c_three
2159          { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2160      }
2161      \skip_set:Nn \l__ctex_ccglue_skip
2162      {
2163          \l__ctex_ziju_dim
2164          plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2165          minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2166          { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2167      }
2168  }
2169  { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2170  \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 \TeX 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`。

```

2171      \ctex_update_parindent:
2172  }
2173  \dim_new:N \l__ctex_ziju_dim

```

`\CTEXindent`
`\CTEXnoindent`

过时命令。

```

2174 \NewDocumentCommand \CTEXindent { }
2175 {
2176     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2177     { \parindent is~ set~ to~ 2\ccwd. }
2178     \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2179 }
2180 \NewDocumentCommand \CTEXnoindent { }
2181 {
2182     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2183     { \parindent is~ set~ to~ 0pt. }
2184     \dim_zero:N \parindent
2185 }

```

14.6 中文数字与日期

```

2186 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2187 \RequirePackage { zhnumber }

\chinese 2188 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2189 \cs_new_eq:NN \@chinese \@zhnum
2190 \cs_new_eq:NN \Chinese \chinese
2191 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

```

2192 \ctex_at_end_package:nn { enumitem }
2193 {
2194     \cs_if_free:NF \AddEnumerateCounter
2195     {
2196         \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2197         \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2198         \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2199     }
2200 }

```

```

\CTEXnumber 2201 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 2202 { \protected@edef #1 { \zhnumber {#2} } }

```

```

2203 \NewDocumentCommand \CTEXdigits { m m }
2204 { \protected@edef #1 { \zhdigits {#2} } }

today 2205 \cs_set_eq:NN \CTEX@todayold \today
2206 \keys_define:nn { ctex }
2207 {
2208     today .choice: ,
2209     today / old .code:n =
2210     { \cs_set_eq:NN \today \CTEX@todayold } ,
2211     today / small .code:n =
2212     {
2213         \cs_set_eq:NN \today \zhtoday
2214         \zhnumsetup { time = Arabic }
2215     } ,
2216     today / big .code:n =
2217     {
2218         \cs_set_eq:NN \today \zhtoday
2219         \zhnumsetup { time = Chinese }
2220     } ,
2221     today / unknown .code:n =
2222     { \msg_error:nxx { ctex } { today-undef } {#1} }
2223 }
2224 \msg_new:nnnn { ctex } { today-undef }
2225 { Today~format~`#1'~is~undefined. }
2226 { 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。

```

2227 \tl_if_exist:NF \proofname
2228 {
2229     \tl_new:N \proofname
2230     \tl_set:Nn \proofname { Proof }
2231 }

2232 \keys_define:nn { ctex }
2233 {
2234     contentsname .tl_set:N = \contentsname ,
2235     listfigurename .tl_set:N = \listfigurename ,
2236     listtablename .tl_set:N = \listtablename ,
2237     figurename .tl_set:N = \figurename ,
2238     tablename .tl_set:N = \tablename ,
2239     abstractname .tl_set:N = \abstractname ,
2240     indexname .tl_set:N = \indexname ,
2241     appendixname .tl_set:N = \appendixname ,
2242     proofname .tl_set:N = \proofname ,
2243     bibname .tl_set:N = \refname
2244     bibname .tl_set:N = \bibname
2245 <*beamer>
2246     algorithmname .tl_set:N = \algorithmname ,
2247     bibname .tl_set:N = \bibname ,
2248     refname .tl_set:N = \refname ,
2249     continuation .tl_set:N = \insertcontinuationtext
2250 </beamer>
2251 }

2252 <*ctex>
2253 \msg_new:nnn { ctex } { ctexbibname }
2254 {
2255     Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2256     The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2257 }
2258 \tl_if_exist:NTF \insertcontinuationtext
2259 {
2260     \keys_define:nn { ctex }
2261     {

```

```

2262     algorithmname .tl_set:N = \algorithmname ,
2263     bibname       .tl_set:N = \bibname ,
2264     refname       .tl_set:N = \refname ,
2265     continuation .tl_set:N = \insertcontinuationtext
2266   }
2267 }
2268 {
2269   \tl_if_exist:NTF \bibname
2270   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2271   {
2272     \tl_if_exist:NTF \refname
2273     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2274     {
2275       \msg_warning:nn { ctex } { ctexbibname }
2276       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2277     }
2278   }
2279 }
2280 </ctex>
2281 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 CTeX 文档类或打开 heading 选项下生效。

```
2282 <*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。

```

2283 <*article|book|report>
2284 \seq_new:N \c__ctex_section_headings_seq
2285 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2286 { section , subsection , subsubsection , paragraph , subparagraph }
2287 </article|book|report>

```

```

\c__ctex_headings_seq 2288 \seq_new:N \c__ctex_headings_seq
2289 <*article|book|report>
2290 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2291 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2292 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2293 </article|book|report>
2294 <*beamer>
2295 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2296 { part , section , subsection }
2297 </beamer>

```

```

\__ctex_initial_heading:n 2298 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2299 {
2300   \tl_new:c { CTEX@pre#1 }
2301   \tl_new:c { CTEX@post#1 }
2302   \tl_const:cx { CTEXthe#1 }
2303   {
2304     \exp_not:c { CTEX@pre#1 }
2305     \exp_not:c { CTEX@the#1 }
2306     \exp_not:c { CTEX@post#1 }
2307   }
2308   \tl_const:cx { CTEX@#1name }
2309   {
2310     \group_begin:
2311     \exp_not:c { CTEX@#1@nameformat }
2312     {

```

```

2313         \exp_not:c { CTEX@pre#1 }
2314         \exp_not:N \tl_if_empty:NTF
2315         \exp_not:c { CTEX@#1@numberformat }
2316         { \exp_not:c { CTEX@the#1 } }
2317         {
2318             \group_begin:
2319             \exp_not:c { CTEX@#1@numberformat }
2320             \exp_not:c { CTEX@the#1 }
2321             \group_end:
2322         }
2323         \exp_not:c { CTEX@post#1 }
2324     }
2325     \group_end:
2326 }
2327 }

```

```

__ctex_def_heading_keys:n 2328 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2329 {
2330     \tl_put_right:Nx \l__ctex_tmp_tl
2331     {
2332         #1 .meta:nn = { ctex / #1 } { #####1 } ,
2333         #1 / name .code:n =
2334         { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2335         #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2336         #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2337         #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2338         #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2339         #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2340         #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2341         #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2342         #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
2343         #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2344         #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2345         #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2346         <article|book|report>
2347         #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2348         #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2349         #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2350         #1 / hang .initial:n = true ,
2351         #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2352         #1 / break .tl_set:N = \exp_not:c { CTEX@#1@break } ,
2353         #1 / break+ .code:n =
2354         { \tl_put_right:Nn \exp_not:c { CTEX@#1@break } { #####1 } } ,
2355         #1 / tocline .code:n =
2356         {
2357             \cs_set:Npn \exp_not:c { CTEX@#1@tocline}
2358             \exp_not:n { #####1####2 } { #####1 }
2359         } ,
2360         </article|book|report>
2361         #1 / format+ .code:n =
2362         { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2363         #1 / nameformat+ .code:n =
2364         { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2365         #1 / numberformat+ .code:n =
2366         { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2367         #1 / aftername+ .code:n =
2368         { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2369         #1 / titleformat+ .code:n =
2370         { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2371         #1 / aftertitle+ .code:n =
2372         { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2373         #1 / numbering .initial:n = true ,
2374         #1 / beforeskip .initial:n = \c_zero_skip ,
2375         #1 / afterskip .initial:n = \c_zero_skip ,
2376         #1 / indent .initial:n = \c_zero_dim ,
2377         #1 / beforeskip .value_required:n = true ,
2378         #1 / afterskip .value_required:n = true ,

```

```

2379         #1 / indent .value_required:n = true ,
2380     }
2381 }

```

`\ctex_assign_heading_name:nn` 的 `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。

```

2382 \NewDocumentCommand \ctex_assign_heading_name:nn
2383 { m > { \SplitArgument { \c_one } { , } } +m }
2384 { \__ctex_assign_heading_name:nnn {#1} #2 }
2385 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2386 {
2387     \tl_set:cn { CTEX@pre#1 } {#2}
2388     \IfNoValueTF {#3}
2389     { \tl_clear:c { CTEX@post#1 } }
2390     { \tl_set:cn { CTEX@post#1 } {#3} }
2391 }

```

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

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2392 \group_begin:
2393 <*book|report>
2394 \tl_set:Nn \l__ctex_tmp_tl
2395 {
2396     part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2397     chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2398     chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2399     chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2400     chapter / lofskip .initial:n = \c_zero_skip ,
2401     chapter / lotskip .initial:n = \c_zero_skip ,
2402     chapter / lofskip .value_required:n = true ,
2403     chapter / lotskip .value_required:n = true ,
2404 }
2405 </book|report>
2406 <*article|beamer>
2407 \tl_clear:N \l__ctex_tmp_tl
2408 </article|beamer>

```

定义标题键值选项。

```

2409 \seq_map_inline:Nn \c__ctex_headings_seq
2410 {
2411     \__ctex_initial_heading:n {#1}
2412     \__ctex_def_heading_keys:n {#1}
2413 }
2414 \use:x
2415 {
2416     \group_end:
2417     \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2418 }
2419 <*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。

```

2420 <*book|report>
2421 \cs_new_protected_nopar:Npn \CTEX@fixtopskip
2422 {
2423     \CTEX@fixheadingskip
2424     \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim

```

```

2425     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2426   }
2427 </book|report>

```

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

```

2428 \cs_new_protected_nopar:Npn \CTEX@fixheadingskip
2429 {
2430   \par
2431   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2432   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2433 }
2434 \skip_new:N \l__ctex_heading_skip
2435 \cs_new_protected_nopar:Npn \CTEX@setheadingskip
2436 { \skip_set:Nn \l__ctex_heading_skip }
2437 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\partmark` 提供 `\partmark`。

```

2438 \ProvideDocumentCommand \partmark { m }
2439 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2440 \cs_new_eq:NN \CTEXifname \use_i:nn
2441 \cs_new_protected_nopar:Npn \CTEX@ifnametrue
2442 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2443 \cs_new_protected_nopar:Npn \CTEX@ifnamefalse
2444 { \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。

```

2445 <*book|report>
2446 \cs_new_protected_nopar:Npn \CTEX@addloflotskip #1
2447 {
2448   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2449   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2450   {
2451     \addtocontents { lof }
2452     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2453   }
2454   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2455   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2456   {
2457     \addtocontents { lot }
2458     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2459   }
2460 }
2461 </book|report>

```

```

\CTEX@addtocline
2462 \cs_new_protected:Npn \CTEX@addtocline #1#2
2463 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

2464 <@@=>
\part
2465 <*article>
2466 \renewcommand\part{%
2467   \if@noskipsec \leavevmode \fi
2468   \par

```



```

2469 \CTEX@part@break
2470 % \addvspace{4ex}%
2471 \CTEX@setheadingskip \CTEX@part@beforeskip
2472 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2473 \addvspace \CTEX@headingskip
2474 \ifodd \CTEX@part@afterindent
2475 \@afterindenttrue
2476 \else
2477 \@afterindentfalse
2478 \fi
2479 \secdef\@part\@spart}
2480 </article>
2481 <*book|report>
2482 \renewcommand\part{%
2483 % \if@openright
2484 % \cleardoublepage
2485 % \else
2486 % \clearpage
2487 % \fi
2488 \CTEX@part@break
2489 % \thispagestyle{plain}%
2490 \thispagestyle{\CTEX@part@pagestyle}%
2491 \if@twocolumn
2492 \onecolumn
2493 \@tempswatrue
2494 \else
2495 \@tempswafalse
2496 \fi
2497 % \null\vfil
2498 \CTEX@setheadingskip \CTEX@part@beforeskip
2499 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2500 \vspace*{\CTEX@headingskip}%
2501 \secdef\@part\@spart}
2502 </book|report>

\@part 2503 <*article>
2504 \def\@part[#1]#2{%
2505 \ifnum \c@secnumdepth >\m@ne
2506 \ifodd \CTEX@part@numbering
2507 \CTEX@ifnametrue
2508 \refstepcounter{part}%
2509 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2510 \else
2511 \CTEX@ifnamefalse
2512 \CTEX@makeanchor{part*}%
2513 % \addcontentsline{toc}{part}{#1}%
2514 \fi
2515 \else
2516 \CTEX@ifnamefalse
2517 \CTEX@makeanchor{part*}%
2518 % \addcontentsline{toc}{part}{#1}%
2519 \fi
2520 \CTEX@addtocline{part}{#1}%
2521 {\interlinepenalty \@M
2522 % \normalfont \parindent \z@ \raggedright
2523 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2524 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2525 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2526 \CTEX@partname \CTEX@part@aftername
2527 \fi \fi
2528 % \huge\bfseries #2%
2529 \CTEX@part@titleformat{#2}%
2530 % \markboth{}{}%
2531 \partmark{#1}%
2532 \CTEX@part@aftertitle}%
2533 \nobreak
2534 % \vskip 3ex

```

```

2535 \CTEX@setheadingskip \CTEX@part@afterskip
2536 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2537 \vskip \CTEX@headingskip
2538 \@afterheading}
2539 </article>
2540 <*book|report>
2541 \def\@part[#1]#2{%
2542 \ifnum \c@secnumdepth >-2\relax
2543 \ifodd \CTEX@part@numbering
2544 \CTEX@ifnametrue
2545 \refstepcounter{part}%
2546 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2547 \else
2548 \CTEX@ifnamefalse
2549 \CTEX@makeanchor{part*}%
2550 % \addcontentsline{toc}{part}{#1}%
2551 \fi
2552 \else
2553 \CTEX@ifnamefalse
2554 \CTEX@makeanchor{part*}%
2555 % \addcontentsline{toc}{part}{#1}%
2556 \fi
2557 \CTEX@addtocline{part}{#1}%
2558 % \markboth{}{}%
2559 \partmark{#1}%
2560 {\interlinepenalty \@M
2561 % \normalfont \centering
2562 \normalfont \CTEX@part@format
2563 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2564 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2565 \CTEX@partname \CTEX@part@aftername
2566 \fi \fi
2567 % \Huge\bfseries #2\par}%
2568 \CTEX@part@titleformat{#2}%
2569 \CTEX@part@aftertitle}%
2570 \@endpart}
2571 </book|report>

\@spart 2572 <*article>
2573 \def\@spart#1{%
2574 \CTEX@ifnamefalse
2575 \CTEX@makeanchor@spart{part*}%
2576 {\interlinepenalty \@M
2577 % \normalfont \parindent \z@ \raggedright
2578 \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2579 \CTEX@part@format
2580 % \huge \bfseries #1\par}%
2581 \CTEX@part@titleformat{#1}%
2582 \CTEX@part@aftertitle}%
2583 \nobreak
2584 % \vskip 3ex
2585 \CTEX@setheadingskip \CTEX@part@afterskip
2586 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2587 \vskip \CTEX@headingskip
2588 \@afterheading}
2589 </article>
2590 <*book|report>
2591 \def\@spart#1{%
2592 \CTEX@ifnamefalse
2593 \CTEX@makeanchor@spart{part*}%
2594 {\interlinepenalty \@M
2595 % \normalfont \centering
2596 \normalfont \CTEX@part@format
2597 % \Huge \bfseries #1\par}%
2598 \CTEX@part@titleformat{#1}%
2599 \CTEX@part@aftertitle}%
2600 \@endpart}

```

```

2601 </book|report>

\endpart 2602 <*book|report>
2603 \def\endpart{%
2604 % \vfil
2605 \CTEX@setheadingskip \CTEX@part@afterskip
2606 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2607 \vskip \CTEX@headingskip
2608 \newpage
2609 \if@twoside
2610 \if@openright
2611 \null
2612 \thispagestyle{empty}%
2613 \newpage
2614 \fi
2615 \fi
2616 \if@tempwa
2617 \twocolumn
2618 \fi}
2619 </book|report>

```

14.8.2.2 chapter 的标题

```

2620 <*book|report>

\chapter 2621 \renewcommand\chapter{%
2622 % \if@openright\cleardoublepage\else\clearpage\fi
2623 % \thispagestyle{plain}%
2624 \CTEX@chapter@break
2625 \thispagestyle{\CTEX@chapter@pagestyle}%
2626 \global\@topnum\z@
2627 % \@afterindentfalse
2628 \ifodd \CTEX@chapter@afterindent
2629 \afterindenttrue
2630 \else
2631 \afterindentfalse
2632 \fi
2633 \secdef\@chapter\@schapter}

\@chapter 2634 \def\@chapter[#1]#2{%
2635 \ifnum \c@secnumdepth >\m@ne
2636 <*book>
2637 \if@mainmatter
2638 </book>
2639 \ifodd \CTEX@chapter@numbering
2640 \CTEX@ifnametrue
2641 \refstepcounter{chapter}%
2642 % \typeout{\@chapapp\space\thechapter.}%
2643 \typeout{\CTEXthechapter}%
2644 % \addcontentsline{toc}{chapter}
2645 % {\protect\numberline{\thechapter}#1}%
2646 \else
2647 \CTEX@ifnamefalse
2648 \CTEX@makeanchor{\Hy@chapapp*}%
2649 % \addcontentsline{toc}{chapter}{#1}%
2650 \fi
2651 <*book>
2652 \else
2653 \CTEX@ifnamefalse
2654 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2655 % \addcontentsline{toc}{chapter}{#1}%
2656 \fi
2657 </book>
2658 \else
2659 \CTEX@ifnamefalse
2660 \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2661 % \addcontentsline{toc}{chapter}{#1}%
2662 \fi

```

```

2663 \CTEX@addtocline{chapter}{#1}%
2664 \chaptermark{#1}%
2665 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2666 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2667 \CTEX@addloflotskip{chapter}%
2668 \if@twocolumn
2669 \topnewpage[\@makechapterhead{#2}]%
2670 \else
2671 \makechapterhead{#2}%
2672 \@afterheading
2673 \fi}

\@schapter 2674 \def\@schapter#1{%
2675 \CTEX@ifnamefalse
2676 \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2677 \if@twocolumn
2678 \topnewpage[\@makeschapterhead{#1}]%
2679 \else
2680 \makeschapterhead{#1}%
2681 \afterheading
2682 \fi}

\@makechapterhead 2683 \def\@makechapterhead#1{%
2684 % \vspace*{50\p@}%
2685 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2686 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2687 \vspace*{\CTEX@headingskip}%
2688 % {\normalfont \parindent \z@ \raggedright
2689 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2690 \CTEX@chapter@format
2691 % \ifnum \c@secnumdepth >\m@ne
2692 % \if@mainmatter
2693 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2694 % \fi
2695 % \fi
2696 \CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}%
2697 \interlinepenalty\@M
2698 % \Huge \bfseries #1\par\nobreak
2699 \CTEX@chapter@titleformat{#1}%
2700 \CTEX@chapter@aftertitle
2701 \nobreak
2702 % \vskip 40\p@
2703 \CTEX@setheadingskip \CTEX@chapter@afterskip
2704 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2705 \vskip \CTEX@headingskip
2706 }}

\@makeschapterhead 2707 \def\@makeschapterhead#1{%
2708 % \vspace*{50\p@}%
2709 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2710 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2711 \vspace*{\CTEX@headingskip}%
2712 % {\normalfont \parindent \z@ \raggedright
2713 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2714 \CTEX@chapter@format
2715 \interlinepenalty\@M
2716 % \Huge \bfseries #1\par\nobreak
2717 \CTEX@chapter@titleformat{#1}
2718 \CTEX@chapter@aftertitle
2719 \nobreak
2720 % \vskip 40\p@
2721 \CTEX@setheadingskip \CTEX@chapter@afterskip
2722 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2723 \vskip \CTEX@headingskip
2724 }}

2725 </book|report>

```

14.8.2.3 section 类的标题

```

\@startsection 2726 \def\@startsection#1#2#3#4#5#6{%
2727   \if@noskipsec \leavevmode \fi
2728   \par
2729   % \@tempskipa #4\relax
2730   % \@afterindenttrue
2731   % \ifdim \@tempskipa <\z@
2732   %   \@tempskipa -\@tempskipa \@afterindentfalse
2733   % \fi
2734   \CTEX@update@sectionformat@n{#1}%
2735   \ifodd \CTEX@afterindent
2736     \@afterindenttrue
2737   \else
2738     \@afterindentfalse
2739   \fi
2740   \if@nobreak
2741     \everypar{}%
2742   \else
2743     % \addpenalty\@secpenalty\addvspace\@tempskipa
2744     \csname CTEX@#1@break\endcsname
2745     \CTEX@setheadingskip{#4}%
2746     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2747     \addvspace \CTEX@headingskip
2748   \fi
2749   \@ifstar
2750     {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2751     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}

\@secntformat 2752 \def\@secntformat#1{%
2753   % \csname the#1\endcsname\quad}%
2754   \csname CTEX@#1name\endcsname
2755   \csname CTEX@#1@aftername\endcsname}

\@sect 2756 \def\@sect#1#2#3#4#5#6[#7]#8{%
2757   \ifnum #2>\c@secnumdepth
2758     \CTEX@ifnamefalse
2759     \CTEX@makeanchor@ssect{#1*}%
2760     \let\@svsec\@empty
2761   \else
2762     \ifodd \csname CTEX@#1@numbering\endcsname
2763       \CTEX@ifnametrue
2764       \refstepcounter{#1}%
2765       \protected@edef\@svsec{\@secntformat{#1}\relax}%
2766     \else
2767       \CTEX@ifnamefalse
2768       \CTEX@makeanchor{#1*}%
2769       \let\@svsec\@empty
2770     \fi
2771   \fi
2772   % \@tempskipa #5\relax
2773   % \ifdim \@tempskipa>\z@
2774   \unless \ifodd \CTEX@runin
2775     \begingroup
2776       #6{%
2777         \CTEX@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2778         % \interlinepenalty \@M #8\@par}%
2779         \interlinepenalty \@M
2780         \csname CTEX@#1@titleformat\endcsname{#8}%
2781         \csname CTEX@#1@aftertitle\endcsname}%
2782       \endgroup
2783       \csname #1mark\endcsname{#7}%
2784     % \addcontentsline{toc}{#1}{%
2785     %   \ifnum #2>\c@secnumdepth \else
2786     %     \protect\numberline{\csname the#1\endcsname}%
2787     %   \fi
2788     %   #7}%

```

```

2789 \CTEX@addtocline{#1}{#7}%
2790 \else
2791 \def\@svsechd{%
2792 #6{\hskip\glueexpr #3\relax
2793 % \@svsec #8}%
2794 \@svsec
2795 \csname CTEX@#1@titleformat\endcsname{#8}%
2796 \csname CTEX@#1@aftertitle\endcsname}%
2797 \csname #1mark\endcsname{#7}%
2798 % \addcontentsline{toc}{#1}{%
2799 % \ifnum #2>\c@secnumdepth \else
2800 % \protect\numberline{\csname the#1\endcsname}%
2801 % \fi
2802 % #7}%
2803 \CTEX@addtocline{#1}{#7}}%
2804 \fi
2805 \@xsect{#5}}

\@ssect 2806 \def\@ssect#1#2#3#4#5{%
2807 \CTEX@ifnamefalse
2808 % \@tempskipa #3\relax
2809 % \ifdim \@tempskipa>\z@
2810 \unless \ifodd \CTEX@runin
2811 \begingroup
2812 #4{%
2813 \CTEX@hangfrom{\hskip\glueexpr #1\relax}%
2814 % \interlinepenalty \@M #5\@par}%
2815 \interlinepenalty \@M
2816 \CTEX@titleformat@n{#5}%
2817 \CTEX@aftertitle}%
2818 \endgroup
2819 \else
2820 % \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2821 \def\@svsechd{#4{\hskip\glueexpr #1\relax
2822 \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2823 \fi
2824 \@xsect{#3}}

\@xsect 2825 \def\@xsect#1{%
2826 % \@tempskipa #1\relax
2827 % \ifdim \@tempskipa>\z@
2828 \unless \ifodd \CTEX@runin
2829 \par \nobreak
2830 % \vskip \@tempskipa
2831 \CTEX@setheadingskip{#1}%
2832 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2833 \vskip \CTEX@headingskip
2834 \afterheading
2835 \else
2836 \@nobreakfalse
2837 \global\@noskipsectrue
2838 \everypar{%
2839 \if@noskipsec
2840 \global\@noskipsecfalse
2841 {\setbox\z@\lastbox}%
2842 \clubpenalty\@M
2843 \begingroup \@svsechd \endgroup
2844 \unskip
2845 % \@tempskipa #1\relax
2846 % \hskip -\@tempskipa
2847 \hskip\glueexpr #1\relax
2848 \else
2849 \clubpenalty \@clubpenalty
2850 \everypar{}}%
2851 \fi}%
2852 \fi
2853 \ignorespaces}

```

2854 <@@=-ctex>

\CTEX@hangfrom hang 选项控制是否采用悬挂缩进。

```
2855 \cs_new_protected_nopar:Npn \CTEX@hangfrom
2856 {
2857   \bool_if:NTF \CTEX@hang
2858     { \@hangfrom }
2859     { \noindent \use:n }
2860 }
```

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

```
2861 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2862 {
2863   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2864   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2865   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2866   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2867   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2868   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2869 }
2870 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2871 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2872 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2873 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2874 \cs_new_eq:NN \CTEX@hang \c_true_bool
2875 \cs_new_eq:NN \CTEX@runin \c_false_bool
```

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

```
2876 \cs_new:Npn \CTEX@part@tocline #1#2
2877 {
2878   \CTEXifname
2879   { \CTEXthepart \hspace { 1em } }
2880   { }
2881   #2
2882 }
2883 \cs_new:Npn \CTEX@chapter@tocline #1#2
2884 {
2885   \CTEXifname
2886   { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2887   { }
2888   #2
2889 }
```

\CTEXnumberline

```
2890 \cs_new_nopar:Npn \CTEXnumberline #1
2891 {
2892   \CTEXifname
2893   { \protect \numberline { \use:c { CTEXthe #1 } } }
2894   { }
2895 }

2896 \int_zero:N \l__ctex_tmp_int
2897 \seq_map_inline:Nn \c__ctex_section_headings_seq
2898 {
2899   \int_incr:N \l__ctex_tmp_int
2900   \cs_gset_protected_nopar:cpx {#1}
2901   {
2902     \exp_not:N \@startsection {#1}
2903     { \int_use:N \l__ctex_tmp_int }
2904     { \exp_not:c { CTEX@#1@indent } }
2905     { \exp_not:c { CTEX@#1@beforeskip } }
2906     { \exp_not:c { CTEX@#1@afterskip } }
2907     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2908   }
2909   \cs_new:cpn { CTEX@#1@tocline } ##1##2
2910   { \CTEXnumberline { ##1 } ##2 }
2911 }
```

14.8.2.4 附录标题

```

appendix/name 2912 \keys_define:nn { ctex }
appendix/number 2913 {
appendix/numbering 2914     appendix .meta:nn = { ctex / appendix } {#1} ,
2915     appendix / name .code:n =
2916     { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2917     appendix / number .tl_set:N = \CTEX@appendix@number ,
2918     appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2919     appendix / numbering .initial:n = true
2920 }
2921 \tl_new:N \CTEX@preappendix
2922 \tl_new:N \CTEX@postappendix

\appendix 2923 \cs_new_eq:NN \CTEX@save@appendix \appendix
2924 \cs_gset_protected_nopar:Npn \appendix
2925 {
2926     \CTEX@save@appendix
2927     <*article>
2928     \gdef \CTEX@presection { \CTEX@preappendix }
2929     \gdef \CTEX@thesection { \CTEX@appendix@number }
2930     \gdef \CTEX@postsection { \CTEX@postappendix }
2931     \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2932     </article>
2933     <*book|report>
2934     \gdef \CTEX@prechapter { \CTEX@preappendix }
2935     \gdef \CTEX@thechapter { \CTEX@appendix@number }
2936     \gdef \CTEX@postchapter { \CTEX@postappendix }
2937     \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2938     </book|report>
2939 }

```

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。

```

2940 \cs_new_protected_nopar:Npn \CTEX@makeanchor #1
2941 { }

```

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

```

2942 \seq_new:N \c__ctex_headings_cs_seq
2943 \seq_gset_from_clist:Nn \c__ctex_headings_cs_seq
<article> 2944 { part , spart , sect , ssect }
<book|report> 2945 { part , spart , chapter , schapter , sect , ssect }
2946 \seq_map_inline:Nn \c__ctex_headings_cs_seq
2947 {
2948     \cs_new_eq:cc { CTEx@ #1 } { @ #1 }
2949     \cs_new_eq:cN { CTEx@makeanchor@ #1 } \CTEX@makeanchor
2950 }

```

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

```

2951 \cs_new_protected_nopar:Npn \CTEX@hyperheadinghook
2952 {
2953     \group_begin:
2954     \ifHy@implicit
2955         \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
2956         \seq_map_inline:Nn \c__ctex_headings_cs_seq
2957         {
2958             \cs_if_eq:ccT { H@old@ ##1 } { CTEx@ ##1 }
2959             {
2960                 \cs_gset_eq:cc { @ ##1 } { CTEx@ ##1 }
2961                 \cs_gset_eq:cN { CTEx@makeanchor@ ##1 } \CTEX@makeanchor

```



```

2962     }
2963   }
2964   \else:
2965     \seq_map_inline:Nn \c__ctex_headings_cs_seq
2966     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
2967   \fi:
2968   \group_end:
2969 }

2970 \ctex_at_end_package:nn { hyperref }
2971 {
2972   \cs_gset_protected_nopar:Npn \CTEX@makeanchor #1
2973   {
2974     \Hy@MakeCurrentHrefAuto {#1}
2975     \Hy@raisedlink
2976     {
2977       \hyper@anchorstart { \@currentHref }
2978       \hyper@anchorend
2979     }
2980   }
2981   \CTEX@hyperheadinghook
2982 }

```

14.8.2.6 兼容 titlesec 宏包

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

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

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

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

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

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

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

```

2983 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
2984 {
2985   \ifpackagewith { titlesec } { explicit }
2986   {
2987     \cs_set_eq:NN \__ctex_titlesec_format:Nn
2988     \__ctex_titlesec_format_explicit:Nn
2989   }
2990   { }
2991   \clist_map_inline:nn
2992   { indentafter , noindentafter , indentfirst , nonindentfirst }
2993   {
2994     \ifpackagewith { titlesec } { ##1 }
2995     {
2996       \clist_map_break:n
2997       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
2998     }

```

```

2999     { }
3000   }
3001   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3002 }
3003 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
3004 {
3005   \__ctex_titlesec_format:n {#1}
3006   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3007 }
3008 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
3009 {
3010   \cs_if_free:cF { ttlf@#1 }
3011   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3012 }
3013 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
3014 {
3015   \tl_set:Nx #1
3016   {
3017     \bool_if:cTF { CTEX@#2@runin }
3018     { \exp_not:N \ttlh@runin }
3019     { \exp_not:N \ttlh@hang }
3020     \tl_tail:N #1
3021   }
3022 }
3023 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3024 {
3025   \cs_set_nopar:Npx #1 ##1
3026   {
3027     \bool_if:cTF { CTEX@#2@runin }
3028     { \exp_not:N \ttlh@runin }
3029     { \exp_not:N \ttlh@hang }
3030     \exp_args:No \tl_tail:n { #1 { } }
3031   }
3032 }
3033 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
3034 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3035 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3036 {
3037   \exp_not:n { {#1} {#2} {#3} {#4} }
3038   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3039 }
3040 \@ifpackageloaded { titlesec }
3041 {
3042   {
3043     \ctex_at_end_package:nn { titlesec }
3044     {
3045       \@ifpackagewith { titlesec } { loadonly }
3046       {
3047         { \ctex_titlesec_hook: }
3048       }
3049     }
3050   }

```

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

```

3050 \ctex_at_end_package:nn { titlesec }
3051 {
3052   <*book|report>
3053   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3054   </book|report>
3055   \cs_if_free:NF \ttl@tocpart
3056   {
3057     \cs_set_protected_nopar:Npn \ttl@tocpart
3058     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3059   }
3060   \seq_map_inline:Nn \c__ctex_headings_seq
3061   {
3062     \cs_if_exist:cF { ttl@toc #1 }

```

```

3063     {
3064         \cs_new_protected_nopar:cpx { ttl@toc #1 }
3065         {
3066             \tl_set:Nn \exp_not:N \ttl@a
3067             {
3068                 \exp_not:N \protect
3069                 \exp_not:N \numberline { \exp_not:c { CTEXthe #1 } }
3070             }
3071         }
3072     }
3073 }
3074 }

```

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。

```

3075 \group_begin:
3076 \char_set_catcode_other:N \#
3077 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
3078 {
3079     \ctex_patch_cmd:Nnn \ttl@settopmark
3080     { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3081     {
3082         \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3083         \CTEX@titlepslabel@set {#1}
3084     }
3085     \ctex_patch_cmd:Nnn \ttl@setsubmark
3086     { \protect \@namedef { the#1 } { } }
3087     {
3088         \protect \@namedef { the#1 } { }
3089         \CTEX@titlepslabel@clear {#1}
3090     }
3091     \ctex_patch_cmd:Nnn \ttl@setsubmark
3092     { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3093     {
3094         \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3095         \CTEX@titlepslabel@set {#2}
3096     }
3097 }
3098 \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。

```

3099 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
3100 {
3101     \cs_if_free:cF { CTEXthe#1 }
3102     { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3103 }
3104 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
3105 {
3106     \cs_if_free:cF { CTEXthe#1 }
3107     { \protect \@namedef { CTEXthe#1 } { } }
3108 }

```

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

```

3109 \ctex_at_end_package:nn { titlesec }

```

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

```

3110 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3111 \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。

```

3112 \ctex_at_end_package:nn { titleps }
3113 {
3114   \cs_set_protected_nopar:Npn \ttl@setifthe #1
3115   {
3116     \exp_args:Nco \cs_set_nopar:Npn { ifthe #1 }
3117     {
3118       \CTEXifname
3119       { \protect \@firstoftwo }
3120       { \protect \@secondoftwo }
3121     }
3122   }
3123 }

```

14.8.3 目录标签的宽度

```

\numberline 3124 \cs_new_protected:Npn \CTEX@toc@width@n #1
3125 {
3126   \hbox_set:Nn \l__ctex_tmp_box {#1}
3127   \dim_set:Nn \@tempdima
3128   {
3129     \dim_max:nn { \@tempdima }
3130     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
3131   }
3132 }
3133 \group_begin:
3134 \char_set_catcode_other:N \#
3135 \use:n
3136 {
3137   \group_end:
3138   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3139   { \CTEX@toc@width@n {#1} }
3140   { }
3141   { \ctex_patch_failure:N \numberline }
3142   \@ifpackageloaded { tocloft }
3143   { }
3144   {
3145     \ctex_at_end_package:nn { tocloft }
3146     {
3147       \ctex_preto_cmd:NnnTF \numberline
3148       { \char_set_catcode_letter:n { 64 } }
3149       { \CTEX@toc@width@n {#1} }
3150       { }
3151       { \ctex_patch_failure:N \numberline }
3152     }
3153   }
3154 }

```

14.8.4 页眉信息的修改

```

\ps@headings 3155 <*article>
3156 \if@twoside
3157   \ctex_patch_cmd:Nnn \ps@headings
3158   { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3159   { \CTEXifname { \CTEXthesection \quad } { } }

```

```

3160 \ctex_patch_cmd:Nnn \ps@headings
3161 { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3162 { \CTEXifname { \CTEXthesubsection \quad } { } }
3163 \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。

```

3164 \ctex_patch_cmd:Nnn \ps@headings
3165 { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3166 { \CTEXifname { \CTEXthesection \quad } { } }
3167 \fi:
3168 </article>
3169 <*book|report>
3170 \ctex_patch_cmd:Nnn \ps@headings
3171 {
3172   \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3173     \chapapp \thechapter . ~ \
3174   \fi \fi
3175 }
3176 { \CTEXifname { \CTEXthechapter \quad } { } }
3177 \if@twoside
3178   \ctex_patch_cmd:Nnn \ps@headings
3179   { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3180   { \CTEXifname { \CTEXthesection } { } }
3181 \fi:
3182 </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。

```

3183 \ctex_at_end_package:nn { fancyhdr }
3184 {
3185 <*article>
3186   \ctex_patch_cmd:Nnn \ps@fancy
3187   { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3188   { \CTEXifname { \CTEXthesection \quad } { } }
3189   \ctex_patch_cmd:Nnn \ps@fancy
3190   { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3191   { \CTEXifname { \CTEXthesubsection \quad } { } }
3192 </article>
3193 <*book|report>
3194   \ctex_patch_cmd:Nnn \ps@fancy
3195   { \ifnum \c@secnumdepth > \m@ne \chapapp \thechapter . ~ \ \fi }
3196   { \CTEXifname { \CTEXthechapter \quad } { } }
3197   \ctex_patch_cmd:Nnn \ps@fancy
3198   { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3199   { \CTEXifname { \CTEXthesection } { } }
3200 </book|report>
3201 }
3202 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

```

3203 <*beamer>
3204 \ExplSyntaxOff

    对应 \partpage。

3205 \defbeamertemplate*{part page}{CTEX}[1][]{
3206   \beginngroup
3207   % \centering
3208   % {\usebeamerfont{part name}\usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3209   % \vskip1em\par
3210   \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3211   \parindent \dimexpr \CTEX@part@indent \relax
3212   \CTEX@part@format

```

```

3213 \ifodd \CTEX@part@numbering
3214 \CTEX@partname \CTEX@part@aftername
3215 \fi
3216 \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3217 % \usebeamerfont{part title}\insertpart\par
3218 \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3219 \end{beamercolorbox}
3220 \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3221 \endgroup
3222 }

```

对应 \sectionpage。

```

3223 \defbeamer-template*{section page}{CTEX}[1][]{
3224 \begin{group
3225 % \centering
3226 % {\usebeamerfont{section name}\usebeamercolor[fg]{section name}\sectionname~\insertsectionname}
3227 % \vskip1em\par
3228 \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3229 \parindent \dimexpr \CTEX@section@indent \relax
3230 \CTEX@section@format
3231 \ifodd \CTEX@section@numbering
3232 \CTEX@sectionname \CTEX@section@aftername
3233 \fi
3234 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3235 % \usebeamerfont{section title}\insertsection\par
3236 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3237 \end{beamercolorbox}
3238 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3239 \endgroup
3240 }

```

对应 \subsectionpage。

```

3241 \defbeamer-template*{subsection page}{CTEX}[1][]{
3242 \begin{group
3243 % \centering
3244 % {\usebeamerfont{subsection name}\usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionname}
3245 % \vskip1em\par
3246 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3247 \parindent \dimexpr \CTEX@subsection@indent \relax
3248 \CTEX@subsection@format
3249 \ifodd \CTEX@subsection@numbering
3250 \CTEX@subsectionname \CTEX@subsection@aftername
3251 \fi
3252 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3253 % \usebeamerfont{subsection title}\insertsubsection\par
3254 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3255 \end{beamercolorbox}
3256 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3257 \endgroup
3258 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3259 \defbeamer-templatealias{part page}{default}{CTEX}
3260 \defbeamer-templatealias{section page}{default}{CTEX}
3261 \defbeamer-templatealias{subsection page}{default}{CTEX}
3262 \ExplSyntaxOn
3263 </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。

```

3264 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel:n #1
3265 {
3266 \protected@edef \@currentlabel
3267 {

```

```

3268      \cs_if_exist:CTF { CTEX@the#1 }
3269      { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3270      { \exp_not:o { \@currentlabel } }
3271    }
3272  }

```

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

```

3273 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3274 {
3275   \seq_map_inline:Nn \c__ctex_headings_seq
3276   { \ctex_fix_varioref_label:n { ##1 } }
3277 }

```

`__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。

```

3278 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3279 {
3280   \tl_if_empty:cT { p@#1 }
3281   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3282 }

```

如果 `varioref` 已经被载入,则使用它来设置。

```

3283 \@ifpackageloaded { varioref }
3284 { \ctex_varioref_hook: }
3285 {
3286   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3287   \RenewDocumentCommand \refstepcounter { m }
3288   {
3289     \CTEX@save@refstepcounter {#1}
3290     \CTEX@setcurrentlabel@n {#1}
3291   }
3292   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3293 }

```

14.8.7 载入 *<scheme>* 文件

```

3294 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3295 </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。

```

3296 <*scheme&(article|book|report|beamer)>
3297 \keys_set:nn { ctex / part }
3298 {
3299   aftertitle = \par ,
3300 <*plain>
3301   name       = \partname \space ,
3302 <*article|book|report>
3303   number     = \thepart ,

```

```

3304 </article|book|report>
3305 <*beamer>
3306     number      = \insertromanpartnumber ,
3307 </beamer>
3308 </plain>
3309 <*chinese>
3310     number      = \chinese { part } ,
3311 </chinese>
3312 <*article>
3313     beforeskip   = 4ex ,
3314     afterskip    = 3ex ,
3315 <*plain>
3316     format       = \raggedright ,
3317     nameformat   = \Large \bfseries ,
3318     aftername    = \par \nobreak ,
3319     titleformat  = \huge \bfseries ,
3320     afterindent  = false
3321 </plain>
3322 <*chinese>
3323     format       = \Large \bfseries \centering ,
3324     aftername    = \quad ,
3325     afterindent  = true
3326 </chinese>
3327 </article>
3328 <*book|report>
3329     aftername    = \par \vskip 20 \p@ ,
3330     beforeskip   = Opt \@plus 1fil ,
3331     afterskip    = Opt \@plus 1fil ,
3332     pagestyle    = plain ,
3333     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3334 <*plain>
3335     format       = \centering ,
3336     nameformat   = \huge \bfseries ,
3337     titleformat  = \Huge \bfseries
3338 </plain>
3339 <*chinese>
3340     format       = \huge \bfseries \centering
3341 </chinese>
3342 </book|report>
3343 <*beamer>
3344     format       = \centering ,
3345     nameformat   = \usebeamerfont { part ~ name }
3346                  \usebeamercolor [fg] { part ~ name } ,
3347     aftername    = \vskip 1em \par ,
3348     titleformat  = \usebeamerfont { part ~ title }
3349 </beamer>
3350 }

3351 <*book|report>
3352 \keys_set:nn { ctex / chapter }
3353 {
3354     pagestyle    = plain ,
3355     aftertitle   = \par ,
3356     beforeskip   = 50 \p@ ,
3357     afterskip    = 40 \p@ ,
3358     lofskip      = 10 \p@ ,
3359     lotskip      = 10 \p@ ,
3360     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3361 <*plain>
3362     name         = \chaptername \space ,
3363     number       = \thechapter ,
3364     format       = \raggedright ,
3365     nameformat   = \huge \bfseries ,
3366     aftername    = \par \nobreak \vskip 20 \p@ ,
3367     titleformat  = \Huge \bfseries ,
3368     afterindent  = false ,
3369     tocline      = \CTEXnumberline {#1} #2

```



```

3370 </plain>
3371 <*chinese>
3372     number      = \chinese { chapter } ,
3373     format       = \huge \bfseries \centering ,
3374     aftername    = \quad ,
3375     afterindent  = true
3376 </chinese>
3377 }
3378 </book|report>

3379 <@@=>

3380 \keys_set:nn { ctex / section }
3381 {
3382 <*article|book|report>
3383     number      = \thesection ,
3384     aftername    = \quad ,
3385     aftertitle   = \@@par ,
3386     beforekip    = 3.5ex \@plus 1ex \@minus .2ex ,
3387     afterskip    = 2.3ex \@plus .2ex ,
3388     runin        = false ,
3389     break        = \addpenalty \@secpenalty ,
3390 <*plain>
3391     format       = \Large \bfseries ,
3392     afterindent  = false
3393 </plain>
3394 <*chinese>
3395     format       = \Large \bfseries \centering ,
3396     afterindent  = true
3397 </chinese>
3398 </article|book|report>
3399 <*beamer>
3400 <*plain>
3401     name         = \sectionname \space ,
3402 </plain>
3403     format       = \centering ,
3404     number       = \insertsectionnumber ,
3405     nameformat    = \usebeamerfont { section ~ name }
3406                   \usebeamerfont { section ~ name } ,
3407     aftername    = \vskip 1em \par ,
3408     titleformat  = \usebeamerfont { section ~ title } ,
3409     aftertitle   = \par
3410 </beamer>
3411 }

3412 \keys_set:nn { ctex / subsection }
3413 {
3414 <*article|book|report>
3415     number      = \thesubsection ,
3416     format       = \large \bfseries ,
3417     aftername    = \quad ,
3418     aftertitle   = \@@par ,
3419     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3420     afterskip    = 1.5ex \@plus .2ex ,
3421     runin        = false ,
3422     break        = \addpenalty \@secpenalty ,
3423 <*plain>
3424     afterindent  = false
3425 </plain>
3426 <*chinese>
3427     afterindent  = true
3428 </chinese>
3429 </article|book|report>
3430 <*beamer>
3431 <*plain>
3432     name         = \subsectionname \space ,
3433     number       = \insertsubsectionnumber ,
3434 </plain>

```

```

3435 <*chinese>
3436     number      = \arabic { section } . \arabic { subsection } ,
3437 </chinese>
3438     format      = \centering ,
3439     nameformat   = \usebeamerfont { subsection ~ name }
3440                  \usebeamerfontcolor [fg] { subsection ~ name } ,
3441     aftername    = \vskip 1em \par ,
3442     titleformat  = \usebeamerfont { subsection ~ title } ,
3443     aftertitle   = \par
3444 </beamer>
3445 }

3446 <*article|book|report>

3447 \keys_set:nn { ctex / subsubsection }
3448 {
3449     number      = \thesubsubsection ,
3450     format      = \normalsize \bfseries ,
3451     aftername    = \quad ,
3452     aftertitle   = \@@par ,
3453     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3454     afterskip    = 1.5ex \@plus .2ex ,
3455     runin       = false ,
3456     break       = \addpenalty \@secpenalty ,
3457 <*plain>
3458     afterindent = false
3459 </plain>
3460 <*chinese>
3461     afterindent = true
3462 </chinese>
3463 }

3464 \keys_set:nn { ctex / paragraph }
3465 {
3466     number      = \theparagraph ,
3467     format      = \normalsize \bfseries ,
3468     aftername    = \quad ,
3469     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3470     break       = \addpenalty \@secpenalty ,
3471 <*plain>
3472     afterindent = false
3473 </plain>
3474 <*chinese>
3475     afterindent = true
3476 </chinese>
3477 }

3478 \keys_set:nn { ctex / subparagraph }
3479 {
3480     number      = \thesubparagraph ,
3481     format      = \normalsize \bfseries ,
3482     aftername    = \quad ,
3483     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3484     break       = \addpenalty \@secpenalty ,
3485 <*plain>
3486     afterindent = false
3487 </plain>
3488 <*chinese>
3489     afterindent = true
3490 </chinese>
3491 }

```

处理 sub3section 与 sub4section 的格式。

```

3492 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3493 {
3494     \keys_set:nn { ctex / paragraph }
3495     {
3496         aftertitle = \@@par ,

```

```

3497     afterskip    = 1ex \@plus .2ex ,
3498     runin        = false
3499   }
3500 }
3501 {
3502   \keys_set:nn { ctex / paragraph }
3503   {
3504     afterskip    = 1em ,
3505     runin        = true
3506   }
3507 }
3508 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3509 {
3510   \keys_set:nn { ctex / subparagraph }
3511   {
3512     aftertitle   = \@@par ,
3513     afterskip    = 1ex \@plus .2ex ,
3514     runin        = false
3515   }
3516 }
3517 {
3518   \keys_set:nn { ctex / subparagraph }
3519   {
3520     afterskip    = 1em ,
3521     runin        = true
3522   }
3523 }
3524 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3525 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3526 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3527 <@@=ctex>

```

处理附录的格式。

```

3528 \keys_set:nn { ctex / appendix }
3529 <*article>
3530 { number        = \@Alph \c@section }
3531 </article>
3532 <*book|report>
3533 {
3534   name          = \appendixname \space ,
3535   number        = \@Alph \c@chapter
3536 }
3537 </book|report>
3538 </article|book|report>
3539 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

```
3540 <*ctex|ctexheading>
```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3541 \clist_map_inline:nn { article , book , report , beamer }
3542 {
3543   \ifclassloaded {#1}
3544   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3545   { }
3546 }

```

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。

```

3547 \msg_new:nnn { ctex } { not-standard-class }
3548 {

```

```

3549     None~of~the~standard~document~classes~was~loaded.\\
3550     Heading~`#1'~is~selected.\\
3551     ctex~may~not~work~as~expected.
3552 }
<ctex> 3553 \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 3554 \use:n
3555 {
3556     \tl_if_exist:NTF \c__ctex_std_class_tl
3557     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3558     {
3559         \cs_if_exist:NTF \chapter
3560         {
3561             \cs_if_exist:NF \if@mainmatter
3562             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3563             \tl_const:Nn \c__ctex_class_tl { book }
3564         }
3565         { \tl_const:Nn \c__ctex_class_tl { article } }
3566         \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3567     }
3568     \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3569 }
<ctex> 3570 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3571 </ctex|ctexheading>

```

14.8.10 标题配置文件

```

3572 <*name>
3573 \keys_set_known:nn { ctex }
3574 {
3575     contentsname = 目录 ,
3576     listfigurename = 插图 ,
3577     listtablename = 表格 ,
3578     figurename = 图 ,
3579     tablename = 表 ,
3580     abstractname = 摘要 ,
3581     indexname = 索引 ,
3582     bibname = 参考文献 ,
3583     appendixname = 附录 ,
3584     proofname = 证明 ,
3585     algorithmname = 算法 ,
3586     refname = 参考文献 ,
3587     continuation = (续) ,
3588     part / name = { 第 , 部分 } ,
3589     chapter / name = { 第 , 章 }
3590 }
3591 </name>

```

14.9 chinese 方案的其他设置

```
3592 <*scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
<article|book|report> 3593 \pagestyle { headings }
```

日期格式。

```
3594 \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。

```
3595 \ctex_if_autoindent_touched:F
3596 { \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。

```
3597 <!*generic>
3598 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
```

```

3599 <*beamer>
3600 {
3601   \uselanguage { ChineseGBK }
3602   \languagealias { chinese } { ChineseGBK }
3603   \ctex_file_input:n { ctex-name-gbk.cfg }
3604 }
3605 {
3606   \uselanguage { ChineseUTF8 }
3607   \languagealias { chinese } { ChineseUTF8 }
3608   \ctex_file_input:n { ctex-name-utf8.cfg }
3609 }

```

让 translator 包优先查找中文翻译。

```

3610 \clist_put_left:Nn \trans@languagepath { chinese }
3611 </beamer>
3612 <!*beamer>
3613 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3614 { \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。

```

3615 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3616 { \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。

```

3617 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3618 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3619 </!beamer>
3620 </!generic>

```

不使用标题定义时的通用设置。

```

3621 <*generic>
3622 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3623 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3624 {
3625   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3626   {
3627     \uselanguage { ChineseGBK }
3628     \languagealias { chinese } { ChineseGBK }
3629     \ctex_file_input:n { ctex-name-gbk.cfg }
3630   }
3631   {
3632     \uselanguage { ChineseUTF8 }
3633     \languagealias { chinese } { ChineseUTF8 }
3634     \ctex_file_input:n { ctex-name-utf8.cfg }
3635   }
3636   \clist_put_left:Nn \trans@languagepath { chinese }
3637 }
3638 {
3639   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3640   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3641   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3642   \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3643   { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3644   \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3645   { \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。

```

3646   \tl_if_exist:NT \c__ctex_std_class_tl
3647   { \RequirePackage { indentfirst } }
3648 }
3649 </generic>

```

```
3650 </scheme&chinese>
```

14.10 中文字号

```
3651 <*class|ctex|ctexsize>
```

```
\zihao 3652 \NewDocumentCommand \zihao { m }
3653 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

\ctex_zihao:n 3654 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3655 {
3656   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3657   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3658   { \msg_error:nnn { ctex } { fontsize } {#1} }
3659 }
3660 \msg_new:nnnn { ctex } { fontsize }
3661 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3662 {
3663   The~old~font~size~is~used~if~you~continue.\\
3664   The~available~font~sizes~are~listed~as~follow.\\
3665   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3666 }
```

14.10.1 定义中文字号

```
\c__ctex_font_size_prop
\__ctex_save_font_size:nn
```

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

```
3667 \prop_new:N \c__ctex_font_size_prop
3668 \seq_new:N \c__ctex_font_size_seq
3669 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3670 {
3671   \use:x
3672   {
3673     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3674     {
3675       { \dim_to_decimal:n {#2} }
3676       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3677     }
3678   }
3679   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3680 }
3681 \clist_map_inline:nn
3682 {
3683   { 8 } { 5 bp } ,
3684   { 7 } { 5.5 bp } ,
3685   { -6 } { 6.5 bp } ,
3686   { 6 } { 7.5 bp } ,
3687   { -5 } { 9 bp } ,
3688   { 5 } { 10.5 bp } ,
3689   { -4 } { 12 bp } ,
3690   { 4 } { 14 bp } ,
3691   { -3 } { 15 bp } ,
3692   { 3 } { 16 bp } ,
3693   { -2 } { 18 bp } ,
3694   { 2 } { 22 bp } ,
3695   { -1 } { 24 bp } ,
3696   { 1 } { 26 bp } ,
3697   { -0 } { 36 bp } ,
3698   { 0 } { 42 bp }
3699 }
3700 { \__ctex_save_font_size:nn #1 }
```

```
\ctex_declare_math_sizes:nnnn 3701 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3702 {
```

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

```

3703 \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3704 \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3705 }

```

```

\l__ctex_get_font_sizes:Nn 3706 \cs_new_protected_nopar:Npn \l__ctex_get_font_sizes:Nn #1#2
3707 {
3708   \tl_clear:N #1
3709   \tl_map_inline:nn {#2}
3710   {
3711     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3712     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3713     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3714   }
3715 }

3716 \clist_map_inline:nn
3717 {
3718   { 8 }{ 8 }{ 5pt }{ 5pt } ,
3719   { 7 }{ 7 }{ 5pt }{ 5pt } ,
3720   { -6 }{ -6 }{ 5pt }{ 5pt } ,
3721   { 6 }{ 6 }{ 5pt }{ 5pt } ,
3722   { -5 }{ -5 }{ 6pt }{ 5pt } ,
3723   { 5 }{ 5 }{ 7pt }{ 5pt } ,
3724   { -4 }{ -4 }{ 8pt }{ 6pt } ,
3725   { 4 }{ 4 }{ 5 }{ 6 } ,
3726   { -3 }{ -3 }{ -4 }{ -5 } ,
3727   { 3 }{ 3 }{ 4 }{ 5 } ,
3728   { -2 }{ -2 }{ -3 }{ -4 } ,
3729   { 2 }{ 2 }{ 3 }{ 4 } ,
3730   { -1 }{ -1 }{ -2 }{ -3 } ,
3731   { 1 }{ 1 }{ 2 }{ 3 } ,
3732   { -0 }{ -0 }{ -1 }{ -2 } ,
3733   { 0 }{ 0 }{ 1 }{ 2 }
3734 }
3735 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3736 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3737 {
3738   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3739   { \exp_after:wN \l__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3740   { \msg_error:nnn { ctex } { fontsize } {#2} }
3741 }
3742 \cs_new_protected:Npn \l__ctex_set_font_size:nnNn #1#2#3#4
3743 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

3744 \if_case:w \g__ctex_font_size_flag
3745 \ctex_file_input:n { ctex-c5size.clo }
3746 \or:
3747 \ctex_file_input:n { ctex-cs4size.clo }
3748 \fi:

3749 </class|ctex|ctexsize>
3750 <*c5size>
3751 \ctex_set_font_size:Nnn \normalsize { 5 }
3752 {
3753   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3754   \abovedisplayshortskip \z@ \@plus3\p@
3755   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3756   \belowdisplayskip \abovedisplayskip
3757   \let\@listi\@listI
3758 }
3759 \ctex_set_font_size:Nnn \small { -5 }
3760 {
3761   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3762   \abovedisplayshortskip \z@ \@plus2\p@

```

```

3763 \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3764 \def\@listi{\leftmargin\leftmargini
3765 \topsep 4\p@ \@plus2\p@ \@minus2\p@
3766 \parsep 2\p@ \@plus\p@ \@minus\p@
3767 \itemsep \parsep}
3768 \belowdisplayskip \abovedisplayskip
3769 }
3770 \ctex_set_font_size:Nnn \footnotesize { 6 }
3771 {
3772 \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3773 \abovedisplayshortskip \z@ \@plus\p@
3774 \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3775 \def\@listi{\leftmargin\leftmargini
3776 \topsep 3\p@ \@plus\p@ \@minus\p@
3777 \parsep 2\p@ \@plus\p@ \@minus\p@
3778 \itemsep \parsep}
3779 \belowdisplayskip \abovedisplayskip
3780 }
3781 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3782 \ctex_set_font_size:Nnn \tiny { 7 } { }
3783 \ctex_set_font_size:Nnn \large { -4 } { }
3784 \ctex_set_font_size:Nnn \Large { -3 } { }
3785 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3786 \ctex_set_font_size:Nnn \huge { 2 } { }
3787 \ctex_set_font_size:Nnn \Huge { 1 } { }
3788 </cs5size>
3789 <*cs4size>
3790 \ctex_set_font_size:Nnn \normalsize { -4 }
3791 {
3792 \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3793 \abovedisplayshortskip \z@ \@plus3\p@
3794 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3795 \belowdisplayskip \abovedisplayskip
3796 \let\@listi\@listI
3797 }
3798 \ctex_set_font_size:Nnn \small { 5 }
3799 {
3800 \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3801 \abovedisplayshortskip \z@ \@plus3\p@
3802 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3803 \def\@listi{\leftmargin\leftmargini
3804 \topsep 9\p@ \@plus3\p@ \@minus5\p@
3805 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3806 \itemsep \parsep}
3807 \belowdisplayskip \abovedisplayskip
3808 }
3809 \ctex_set_font_size:Nnn \footnotesize { -5 }
3810 {
3811 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3812 \abovedisplayshortskip \z@ \@plus3\p@
3813 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3814 \def\@listi{\leftmargin\leftmargini
3815 \topsep 6\p@ \@plus2\p@ \@minus2\p@
3816 \parsep 3\p@ \@plus2\p@ \@minus\p@
3817 \itemsep \parsep}
3818 \belowdisplayskip \abovedisplayskip
3819 }
3820 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3821 \ctex_set_font_size:Nnn \tiny { -6 } { }
3822 \ctex_set_font_size:Nnn \large { -3 } { }
3823 \ctex_set_font_size:Nnn \Large { -2 } { }
3824 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3825 \ctex_set_font_size:Nnn \huge { -1 } { }
3826 \ctex_set_font_size:Nnn \Huge { 1 } { }
3827 </cs4size>
<ctexsize> 3828 \normalsize

```



```
3829 <*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`。

```
3830 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3831 {
3832   \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。

```
3833   \group_begin: \footnotesize \exp_args:NNNo \group_end:
3834   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3835 }
```

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

```
3836 \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。

```
3837 \NewDocumentCommand \CTeX { }
3838 { \ifincsname CTeX \else: C \TeX \fi: }
3839 \ctex_at_end_package:n { hyperref }
3840 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

`captiondelimater` 过时选项。

```
3841 \keys_define:nn { ctex }
3842 {
3843   captiondelimater .code:n =
3844   {
3845     \msg_warning:nnn { ctex } { deprecated-option }
3846     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3847   }
3848 }
```

```
3849 </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`

```
3850 <*scheme&chinese&(article|book|report)>
3851 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3852 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3853 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3854 </scheme&chinese&(article|book|report)>
3855 <*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。

```

3856 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3857 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3858 \msg_new:nnn { ctex } { fontset-unavailable }
3859 { 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。

```

3860 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3861 {
3862   \tl_if_empty:NTF \g__ctex_fontset_tl
3863   {
3864     \ctex_detected_platform:
3865     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3866   }
3867   {
3868     \file_if_exist:NTF { ctex-fontset- \g__ctex_fontset_tl .def }
3869     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3870     {
3871       \use:x
3872       {
3873         \ctex_detected_platform:
3874         \msg_error:nxxx { ctex } { fontset-not-found }
3875         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3876       }
3877       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3878     }
3879   }
3880 }
3881 \@onlypreamble \ctex_load_fontset:
3882 \msg_new:nnnn { ctex } { fontset-not-found }
3883 {
3884   CTeX~fontset~`#1'~could~not~be~found.\\
3885   Fontset~`#2'~will~be~used~instead.
3886 }
3887 { You~may~run~`mktexlsr'~firstly. }

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

3888 \keys_define:nn { ctex }
3889 {
3890   fontset .code:n =
3891   {
3892     \ctex_if_preamble:TF
3893     {
3894       \str_if_eq_x:nnTF {#1} { none }
3895       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3896       {
3897         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3898         {
3899           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3900           \ctex_load_fontset:
3901         }
3902         {
3903           \msg_error:nxxx { ctex } { fontset-loaded }
3904           { \g__ctex_fontset_tl } {#1}
3905         }
3906       }
3907     }
3908     { \msg_error:nn { ctex } { fontset-only-preamble } }
3909   }
3910 }
3911 \msg_new:nnnn { ctex } { fontset-loaded }
3912 {

```

```

3913 CTeX~fontset~`#1'~has~been~loaded.
3914 \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
3915 }
3916 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3917 \msg_new:nnn { ctex } { fontset-only-preamble }
3918 {
3919 The~ `fontset'~ option~ can~ be~ used~ only~ in~ preamble.
3920 }

```

载入中文字库。

```

3921 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3922 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

3923 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3924 </class|ctex>
3925 <*config>
3926 %%
3927 </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。

```

3928 <*ctexopts>
3929 %%
3930 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3931 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

3932 <*c19|c70>
3933 %%
3934 %% Chinese characters
3935 %%
<c19> 3936 %% character set: GBK (extension of GB 2312)
<c70> 3937 %% character set: Unicode
      3938 %% font encoding: Unicode
      3939 %%
3940 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm&c19> 3941 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm&c70> 3942 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf&c19> 3943 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 3944 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 3945 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt&c70> 3946 \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。

```

3947 <*uptex|aptex>
3948 \DeclareKanjiFamily{JY2}{rm}{-}
3949 \DeclareKanjiFamily{JY2}{sf}{-}
3950 \DeclareKanjiFamily{JY2}{tt}{-}
3951 \DeclareKanjiFamily{JT2}{rm}{-}

```

```

3952 \DeclareKanjiFamily{JT2}{sf}{-}
3953 \DeclareKanjiFamily{JT2}{tt}{-}
3954 </uptex|aptex>

3955 <*rm>
3956 <*c19>
3957 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3958 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3959 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3960 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3961 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3962 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3963 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3964 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3965 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3966 </c19>
3967 <*c70>
3968 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3969 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3970 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3971 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3972 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3973 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3974 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3975 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3976 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3977 </c70>
3978 </rm>
3979 <*uptex|aptex>
3980 \DeclareFontShape{JY2}{rm}{m}{n}{<-> upzhserif-h}{-}
3981 \DeclareFontShape{JY2}{rm}{m}{it}{<-> upzhserifit-h}{-}
3982 \DeclareFontShape{JY2}{rm}{bx}{n}{<-> upzhserifb-h}{-}
3983 \DeclareFontShape{JT2}{rm}{m}{n}{<-> upzhserif-v}{-}
3984 \DeclareFontShape{JT2}{rm}{m}{it}{<-> upzhserifit-v}{-}
3985 \DeclareFontShape{JT2}{rm}{bx}{n}{<-> upzhserifb-v}{-}
3986 </uptex|aptex>

3987 <*sf>
3988 <*c19>
3989 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3990 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3991 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3992 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3993 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3994 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3995 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3996 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3997 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3998 </c19>
3999 <*c70>
4000 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
4001 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
4002 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4003 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4004 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4005 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4006 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4007 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4008 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4009 </c70>
4010 </sf>
4011 <*uptex|aptex>
4012 \DeclareFontShape{JY2}{sf}{m}{n}{<-> upzhsans-h}{-}
4013 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{-}
4014 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{-}
4015 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{-}
4016 </uptex|aptex>

4017 <*tt>

```

```

4018 <*c19>
4019 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbksfs}{\CJKnormal}
4020 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbksfs}{\CJKbold}
4021 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbksfs}{\CJKbold}
4022 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbksfssl}{\CJKnormal}
4023 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbksfssl}{\CJKbold}
4024 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbksfssl}{\CJKbold}
4025 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbksfs}{\CJKnormal}
4026 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbksfs}{\CJKbold}
4027 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbksfs}{\CJKbold}
4028 </c19>
4029 <*c70>
4030 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifss}{\CJKnormal}
4031 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifss}{\CJKbold}
4032 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifss}{\CJKbold}
4033 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifsssl}{\CJKnormal}
4034 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifsssl}{\CJKbold}
4035 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifsssl}{\CJKbold}
4036 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifss}{\CJKnormal}
4037 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifss}{\CJKbold}
4038 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifss}{\CJKbold}
4039 </c70>
4040 </tt>
4041 <*uptex|aptex>
4042 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{ }
4043 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{ }
4044 </uptex|aptex>
4045 <*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。)

```

4046 <*windows>
4047 \file_if_exist:nTF { C:/bootfont.bin }
4048 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
4049 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
4050 </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。

```

4051 <*windowsnew>
4052 \tl_new:N \l__ctex_msyh_suffix_tl
4053 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
4054 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
4055 {
4056   \file_if_exist:nF { msyh.ttc }
4057   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
4058 }
4059 </windowsnew>
4060 <*windowsnew|windowsold>
4061 \sys_if_engine_pdftex:TF
4062 {
4063   \ctex_zhmap_case:nnn
4064   {
4065     \ctex_punct_set:n { windows }
4066     \setCJKmainfont
4067     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
4068 <*windowsold>
4069     \setCJKsansfont { simhei.ttf }
4070     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4071 </windowsold>

```

```

4072 <*windowsnew>
4073     \setCJKsansfont [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4074     \setCJKfamilyfont { zhyahei }
4075         [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
4076     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4077     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
4078 </windowsnew>
4079     \setCJKmonofont { simfang.ttf }
4080     \setCJKfamilyfont { zhkai } { simkai.ttf }
4081     \setCJKfamilyfont { zhfs } { simfang.ttf }
4082     \setCJKfamilyfont { zhsong } { simsun.ttc }
4083     \setCJKfamilyfont { zhhei } { simhei.ttf }
4084     \setCJKfamilyfont { zhli } { simli.ttf }
4085     \setCJKfamilyfont { zhyou } { simyou.ttf }
4086     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4087     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4088     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4089     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4090 }
4091 {
4092     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
4093     \ctex_punct_set:n { windows }
4094     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4095     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4096     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4097 }
4098 {
4099     \tl_set:Nn \CJKrmdefault { rm }
4100     \tl_set:Nn \CJKsfdefault { sf }
4101     \tl_set:Nn \CJKttdefault { tt }
4102 }
4103 }
4104 {
4105     \sys_if_engine_uptex:TF
4106     {
4107         \ctex_set_upfonts:nnnnnn
4108             {simsun.ttc} {simhei.ttf} {simkai.ttf}
4109             {msyh\l__ctex_msyh_suffix_tl} {msyhbd\l__ctex_msyh_suffix_tl}
4110             {simhei.ttf} {simhei.ttf}
4111             {simfang.ttf}
4112         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4113         \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4114         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4115         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4116 <windowsnew> \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4117         \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4118         \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4119         \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4120         \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4121     }
4122     {
4123 <*windowsold>
4124         \setCJKmainfont
4125             [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
4126         \setCJKsansfont { SimHei }
4127         \setCJKmonofont { FangSong_GB2312 }
4128         \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
4129         \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
4130 </windowsold>
4131 <*windowsnew>
4132         \setCJKmainfont
4133             [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
4134         \setCJKsansfont
4135             [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4136         \setCJKmonofont { FangSong }
4137         \setCJKfamilyfont { zhkai } { KaiTi }
4138         \setCJKfamilyfont { zhfs } { FangSong }

```

```

4139 </windowsnew>
4140     \setCJKfamilyfont { zhsong } { SimSun }
4141     \setCJKfamilyfont { zhhei } { SimHei }
4142     \setCJKfamilyfont { zhli } { LiSu }
4143     \setCJKfamilyfont { zhyou } { YouYuan }
4144 <*windowsnew>
4145     \setCJKfamilyfont { zhyahei }
4146     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
4147 </windowsnew>
4148 }
4149 }
4150 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```
4151 <*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。

```

4152 \sys_if_engine_pdftex:TF
4153 {
4154     \sys_if_output_pdf:TF
4155     { \ctex_fontset_error:n { adobe } }
4156     {
4157         \ctex_zhmap_case:nnn
4158         {
4159             \setCJKmainfont
4160             [
4161                 cmap = UniGB-UTF16-H ,
4162                 BoldFont = AdobeHeitiStd-Regular.otf ,
4163                 ItalicFont = AdobeKaitiStd-Regular.otf
4164             ] { AdobeSongStd-Light.otf }
4165             \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4166             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4167             \setCJKfamilyfont { zhsong }
4168             [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
4169             \setCJKfamilyfont { zhhei }
4170             [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
4171             \setCJKfamilyfont { zhkai }
4172             [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
4173             \setCJKfamilyfont { zhfs }
4174             [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
4175             \ctex_punct_set:n { adobe }
4176             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4177             \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4178             \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4179             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4180             \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4181         }
4182         {
4183             \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
4184             \ctex_punct_set:n { adobe }
4185             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4186             \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4187             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4188         }
4189         { \ctex_fontset_error:n { adobe } }
4190     }
4191 }
4192 {
4193     \sys_if_engine_uptex:TF
4194     {
4195         \ctex_set_upfonts:nnnnnn
4196         {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
4197         {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
4198         {AdobeFangsongStd-Regular.otf}

```



```

4199     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4200     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4201     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4202     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4203   }
4204   {
4205     \setCJKmainfont
4206     [
4207       BoldFont = AdobeHeitiStd-Regular ,
4208       ItalicFont = AdobeKaitiStd-Regular
4209     ] { AdobeSongStd-Light }
4210     \setCJKsansfont { AdobeHeitiStd-Regular }
4211     \setCJKmonofont { AdobeFangsongStd-Regular }
4212     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4213     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4214     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4215     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4216   }
4217 }
4218 </adobe>
4219 <*fandol>

```

14.15.4 ctex-fontset-fandol.def

```

4220 \sys_if_engine_pdftex:TF
4221 {
4222   \sys_if_output_pdf:TF
4223   { \ctex_fontset_error:n { fandol } }
4224   {
4225     \ctex_zhmap_case:nnn
4226     {
4227       \setCJKmainfont
4228       [
4229         cmap = UniGB-UTF16-H ,
4230         BoldFont = FandolSong-Bold.otf ,
4231         ItalicFont = FandolKai-Regular.otf
4232       ] { FandolSong-Regular.otf }
4233       \setCJKsansfont
4234       [
4235         cmap = UniGB-UTF16-H ,
4236         BoldFont = FandolHei-Bold.otf
4237       ] { FandolHei-Regular.otf }
4238       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4239       \setCJKfamilyfont { zhsong }
4240       [
4241         cmap = UniGB-UTF16-H ,
4242         BoldFont = FandolSong-Bold.otf
4243       ] { FandolSong-Regular.otf }
4244       \setCJKfamilyfont { zhhei }
4245       [
4246         cmap = UniGB-UTF16-H ,
4247         BoldFont = FandolHei-Bold.otf
4248       ] { FandolHei-Regular.otf }
4249       \setCJKfamilyfont { zhfs }
4250       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4251       \setCJKfamilyfont { zhkai }
4252       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4253       \ctex_punct_set:n { fandol }
4254       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4255       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4256       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4257       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4258       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4259       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4260     }
4261     {
4262       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }

```



```

4263         \ctex_punct_set:n { fandol }
4264         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4265         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4266         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4267     }
4268     { \ctex_fontset_error:n { fandol } }
4269 }
4270 }
4271 {
4272     \sys_if_engine_uptex:TF
4273     {
4274         \ctex_set_upfonts:nnnnnn
4275         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4276         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4277         {FandolFang-Regular.otf}
4278         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4279         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4280         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4281         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4282     }
4283     {
4284         \setCJKmainfont
4285         [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
4286         { FandolSong-Regular }
4287         \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4288         \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4289         \setCJKfamilyfont { zhsong }
4290         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4291         \setCJKfamilyfont { zhhei }
4292         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4293         \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4294         \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4295     }
4296 }
4297 </fandol>
4298 <*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

```

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

```

/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。

```

4299 \sys_if_engine_pdftex:TF
4300 { \ctex_fontset_error:n { mac } }
4301 {
4302   \sys_if_engine_uptex:TF
4303   { \ctex_fontset_error:n { mac } }
4304   {
4305     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4306     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4307     \setCJKmonofont { STFangsong }
4308     \setCJKfamilyfont { zhsong } { STSong }
4309     \setCJKfamilyfont { zhhei } { STHeiti }
4310     \setCJKfamilyfont { zhfs } { STFangsong }
4311     \setCJKfamilyfont { zhkai } { STKaiti }
4312   }
4313 }
4314 </mac>
4315 <*founder>

```

14.15.6 ctex-fontset-founder.def

```

4316 \sys_if_engine_pdftex:TF
4317 {
4318   \ctex_zhmap_case:nnn
4319   {
4320     \setCJKmainfont
4321     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4322     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXHK.TTF }
4323     \setCJKmonofont { FZFSK.TTF }

```

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>

```

4324 \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4325 \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4326 \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4327 \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4328 \setCJKfamilyfont { zhli } { FZLSK.TTF }
4329 \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4330 \ctex_punct_set:n { founder }
4331 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4332 \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4333 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4334 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4335 \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4336 \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4337 \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4338 }
4339 {
4340 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4341 \ctex_punct_set:n { founder }
4342 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4343 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4344 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4345 }
4346 { \ctex_fontset_error:n { founder } }
4347 }
4348 {
4349 \sys_if_engine_uptex:TF
4350 {
4351 \ctex_set_upfonts:nnnnnn
4352 { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4353 { FZXH1K.TTF } { FZHTK.TTF }
4354 { FZFSK.TTF }
4355 \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4356 \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4357 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4358 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4359 \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4360 \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4361 \ctex_set_upfamily:nnn { zhyou } { upschtg } {}
4362 \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4363 }
4364 {
4365 \setCJKmainfont
4366 [ 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），Xe_{La}TeX 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。

```

4367 \setCJKsansfont [ BoldFont = FZHei-B01 ]
4368 { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4369 \setCJKmonofont { FZFangSong-Z02 }
4370 \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4371 \setCJKfamilyfont { zhhei } { FZHei-B01 }
4372 \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4373 \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4374 \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4375 \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4376 }
4377 }
4378 </founder>
4379 <*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。

```
4380 \sys_if_engine_pdftex:TF
4381 {
4382   \ctex_zhmap_case:nnn
4383   {
4384     \setCJKmainfont
4385       [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4386     \setCJKsansfont { wqy-zenhei.ttc }
4387     \setCJKmonofont { uming.ttc }
4388     \setCJKfamilyfont { zhsong } { uming.ttc }
4389     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4390     \setCJKfamilyfont { zhkai } { ukai.ttc }
4391     \ctex_punct_set:n { ubuntu }
4392     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4393     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4394     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4395     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4396     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4397   }
4398   {
4399     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4400     \ctex_punct_set:n { ubuntu }
4401     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4402     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4403     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4404   }
4405   { \ctex_fontset_error:n { ubuntu } }
4406 }
4407 {
4408   \sys_if_engine_uptex:TF
4409   {
4410     \ctex_set_upfonts:nnnnnn
4411       { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4412       { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4413       { uming.ttc }
4414     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4415     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4416     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4417   }
4418   {
4419     \setCJKmainfont
4420       [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4421     \setCJKsansfont { WenQuanYi~Zen~Hei }
4422     \setCJKmonofont { AR~PL~UMing~CN }
4423     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4424     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4425     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4426   }
4427 }
4428 </ubuntu>
```

14.15.8 中文字体命令

```
4429 <!*windows>
```

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

```

4430 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4431 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
<ubuntu> 4432 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4433 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4434 <*windowsnew|windowsold|founder>
4435 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4436 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4437 </windowsnew|windowsold|founder>
<windowsnew> 4438 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4439 </!windows>
4440 </fontset>
4441 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4442 \begingroup\catcode61\catcode48\catcode32=10\relax%
4443 \catcode 35=6 % #
4444 \catcode123=1 % {
4445 \catcode125=2 % }
4446 \toks0{\endlinechar=\the\endlinechar\relax}%
4447 \toks2{\endlinechar=13 }%
4448 \def\x#1 #2 {%
4449   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4450   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4451 \x 13 5 % carriage return
4452 \x 32 10 % space
4453 \x 35 6 % #
4454 \x 40 12 % (
4455 \x 41 12 % )
4456 \x 45 12 % -
4457 \x 46 12 % .
4458 \x 47 12 % /
4459 \x 58 12 % :
4460 \x 60 12 % <
4461 \x 61 12 % =
4462 \x 64 11 % @
4463 \x 91 12 % [
4464 \x 93 12 % ]
4465 \x 123 1 % {
4466 \x 125 2 % }
4467 \edef\x#1{\endgroup%
4468   \edef\noexpand#1{%
4469     \the\toks0 %
4470     \let\noexpand\noexpand\noexpand#1%
4471     \noexpand\noexpand\noexpand\undefined%
4472     \noexpand\noexpand\noexpand\endinput}%
4473     \the\toks2}%
4474 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4475 \begingroup\expandafter\endgroup
4476 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4477 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4478   \expandafter\ifx\csname pdfoutput\endcsname\relax
4479     iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4480   \else ifctexpdf\fi
4481 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4482 \begingroup
4483 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4484   \long\def\x#1\ProvidesFile#2[#3]{%
4485     #1%
4486     \immediate\write-1{File: #2 #3}%
4487     \expandafter\xdef\csname ver@#2\endcsname{#3}}
4488   \expandafter\x%

```

```

4489 \fi
4490 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4491 <*windows>
4492 \ProvidesFile{zhwindowsfonts.tex}%
4493 [2016/10/25 v2.4.5 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4494
4495 \ifzhmappdf
4496 \pdfmapline{=gbk@UGBK@ <simsum.ttc}
4497 \pdfmapline{=gbksong@UGBK@ <simsum.ttc}
4498 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4499 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4500 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4501 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4502 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4503
4504 \pdfmapline{=cyberb@Unicode@ <simsum.ttc}
4505 \pdfmapline{=unisong@Unicode@ <simsum.ttc}
4506 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4507 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4508 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4509 \pdfmapline{=unili@Unicode@ <simli.ttf}
4510 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4511
4512 \pdfmapline{=gbksongsl@UGBK@ <simsum.ttc}
4513 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4514 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4515 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4516 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4517 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4518
4519 \pdfmapline{=unisongsl@Unicode@ <simsum.ttc}
4520 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4521 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4522 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4523 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4524 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4525
4526 \else
4527 \special{pdf:mapline gbk@UGBK@ unicode :0:simsum.ttc -v 50}
4528 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsum.ttc -v 50}
4529 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4530 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4531 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4532 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4533 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4534
4535 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsum.ttc -v 50}
4536 \special{pdf:mapline unisong@Unicode@ unicode :0:simsum.ttc -v 50}
4537 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4538 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4539 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4540 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4541 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4542
4543 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsum.ttc -s .167 -v 50}
4544 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4545 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4546 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4547 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4548 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4549
4550 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsum.ttc -s .167 -v 50}
4551 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4552 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}

```



```

4553 \special{pdf:mapline unifssl@Unicode@    unicode simfang.ttf -s .167 -v 50}
4554 \special{pdf:mapline unilisl@Unicode@    unicode simli.ttf -s .167 -v 150}
4555 \special{pdf:mapline uniyousl@Unicode@    unicode simyou.ttf -s .167 -v 60}
4556
4557 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4558 <*adobe>
4559 \ProvidesFile{zhadobefonts.tex}%
4560 [2016/10/25 v2.4.5 Adobe font map loader for DVIPDFMx (CTEX)]
4561
4562 \ifzhmappdf
4563 %% pdfTeX does not support OTF fonts
4564
4565 \else
4566 \special{pdf:mapline gbk@UGBK@    UniGB-UTF16-H AdobeSongStd-Light.otf}
4567 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4568 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4569 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4570 \special{pdf:mapline gbkfs@UGBK@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4571
4572 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4573 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4574 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4575 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4576 \special{pdf:mapline unifs@Unicode@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4577
4578 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4579 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4580 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4581 \special{pdf:mapline gbkfssl@UGBK@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4582
4583 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4584 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4585 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4586 \special{pdf:mapline unifssl@Unicode@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4587
4588 </adobe>

```

14.15.9.3 zhfandolfonts.tex

```

4589 <*fandol>
4590 \ProvidesFile{zhfandolfonts.tex}%
4591 [2016/10/25 v2.4.5 Fandol font map loader for DVIPDFMx (CTEX)]
4592
4593 \ifzhmappdf
4594 %% pdfTeX does not support OTF fonts
4595
4596 \else
4597 \special{pdf:mapline gbk@UGBK@    UniGB-UTF16-H FandolSong-Regular.otf}
4598 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4599 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4600 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4601 \special{pdf:mapline gbkfs@UGBK@  UniGB-UTF16-H FandolFang-Regular.otf}
4602
4603 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4604 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4605 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4606 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4607 \special{pdf:mapline unifs@Unicode@  UniGB-UTF16-H FandolFang-Regular.otf}
4608
4609 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4610 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4611 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4612 \special{pdf:mapline gbkfssl@UGBK@  UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4613
4614 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}

```

```

4615 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4616 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4617 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4618
4619 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4620 <*founder>
4621 \ProvidesFile{zhfounderfonts.tex}%
4622 [2016/10/25 v2.4.5 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4623
4624 \ifzhmappdf
4625 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4626 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4627 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4628 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4629 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4630 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4631 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4632
4633 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4634 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4635 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4636 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4637 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4638 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4639 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4640
4641 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4642 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4643 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4644 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4645 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4646 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4647
4648 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4649 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4650 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4651 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4652 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4653 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4654
4655 \else
4656 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4657 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4658 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4659 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4660 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4661 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4662 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4663
4664 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4665 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4666 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4667 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4668 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4669 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4670 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4671
4672 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4673 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4674 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4675 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4676 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4677 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4678
4679 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}

```



```

4680 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4681 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4682 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4683 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4684 \special{pdf:mapline uniyosl@Unicode@ unicode FZY1K.TTF -s .167}
4685
4686 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4687 <*ubuntu>
4688 \ProvidesFile{zhubuntufonts.tex}%
4689 [2016/10/25 v2.4.5 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4690
4691 \ifzhmappdf
4692 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4693 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4694 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4695 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4696 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4697 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4698
4699 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4700 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4701 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4702 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4703 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4704 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
4705
4706 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4707 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4708 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4709 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4710 \pdfmapline{=gbkyosl@UGBK@ <wqy-zenhei.ttc}
4711
4712 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4713 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4714 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4715 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4716 \pdfmapline{=uniyosl@Unicode@ <wqy-zenhei.ttc}
4717
4718 \else
4719 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4720 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4721 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4722 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4723 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4724
4725 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4726 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4727 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4728 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4729 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4730
4731 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4732 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4733 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4734 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4735
4736 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4737 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4738 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4739 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4740
4741 </ubuntu>
4742 \fi
4743

```

```
4744 \ctex@zhmap@endinput
4745 </zhmap>
```

14.15.10 制作 spa 文件

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

```

4746 <'spa>
4747 <'macro>
4748 \input expl3-generic %
4749 \ExplSyntaxOn
4750 \sys_if_engine_xetex:F
4751 {
4752   \msg_new:nn { ctex } { xetex }
4753   { XeTeX~is~required~to~compile~this~document! }
4754   \msg_fatal:nn { ctex } { xetex }
4755 }

```

CJKpunct 定义的标点符号是:

‘ “ [] { < 《 ☒ 【
—…、。 , . : ; ! ? %))] } > » ∑ 】 ’ ” 」 ∟

注意顺序不能改变。

```

4756 \seq_new:N \c__ctex_punct_seq
4757 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4758 {
4759     "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4760     "3008 , "300A , "3016 , "3010 ,
4761     "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4762     "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4763     "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4764 }

```

`\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。

```

4765 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4766 {
4767   \iow_open:Nn \g__ctex_spa_iow {#1}
4768   \clist_map_inline:nn {#2}
4769     { \__ctex_write_family:nn ##1 }
4770   \iow_close:N \g__ctex_spa_iow
4771 }
4772 \iow_new:N \g__ctex_spa_iow
4773 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn
4774
4775 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4776 {
4777   \group_begin:
4778     \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4779     \l__ctex_punct_font
4780     \clist_clear:N \l__ctex_punct_bounds_clist
4781     \seq_map_inline:Nn \c__ctex_punct_seq
4782       { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4783     \iow_now:Nx \g__ctex_spa_iow
4784     {
4785       \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4785         { \l__ctex_punct_bounds_clist , , , }
4786     }
4787     \group_end:
4788 }
4789 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4790 {

```

```

4791 \clist_put_right:Nx \l__ctex_punct_bounds_clist
4792 {
4793     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4794     \__ctex_calc_bounds:nn { \c_three } {#1}
4795 }
4796 }
4797 \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。

```

4798 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4799 {
4800     \fp_eval:n
4801     {
4802         round
4803         (
4804             \dim_to_decimal_in_unit:nn
4805             { 100 \xetex_glyphbounds:D #1 ~ #2 }
4806             { 1 em }
4807         )
4808     }
4809 }
4810 \ExplSyntaxOff
4811 </macro>

```

下面是 CTeX 定义的一些字体。

```

4812 <*make>
4813 \input ctexpamacro %
4814
4815 \MAKESPA {ctexpunct.spa}
4816 {
4817     {adobezhsong}      {AdobeSongStd-Light} ,
4818     {adobezhhei}      {AdobeHeitiStd-Regular} ,
4819     {adobezhkai}      {AdobeKaitiStd-Regular} ,
4820     {adobezhfs}       {AdobeFangsongStd-Regular} ,
4821     {fandolzhsong}    {FandolSong} ,
4822     {fandolzhsongb}   {FandolSong-Bold} ,
4823     {fandolzhhei}     {FandolHei} ,
4824     {fandolzhheib}    {FandolHei-Bold} ,
4825     {fandolzhkai}     {FandolKai} ,
4826     {fandolzhfs}      {FandolFang} ,
4827     {founderzhsong}   {FZShuSong-Z01} ,
4828     {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
4829     {founderzhhei}    {FZHei-B01} ,
4830     {founderzhheil}   {FZXiHeiI-Z08} ,
4831     {founderzhkai}    {FZKai-Z03} ,
4832     {founderzhfs}     {FZFangSong-Z02} ,
4833     {founderzhli}     {FZLiShu-S01} ,
4834     {founderzhyou}    {FZXiYuan-M01} ,
4835     {founderzhhyoub}  {FZZhunYuan-M02} ,
4836     {ubuntuzhsong}    {AR PL UMinG CN} ,
4837     {ubuntuzhhei}     {WenQuanYi Zen Hei} ,
4838     {ubuntuzhkai}     {AR PL UKai CN} ,
4839     {windowszhsong}   {SimSun} ,
4840     {windowszhhei}    {SimHei} ,
4841     {windowszhkai}    {KaiTi} ,
4842     {windowszhfs}     {FangSong} ,
4843     {windowszhli}     {LiSu} ,
4844     {windowszhyou}    {YouYuan} ,
4845     {windowszhyawei}   {Microsoft YaHei} ,
4846     {windowszhyawei}   {Microsoft YaHei Bold}
4847 }
4848
4849 \primitive\end
4850 </make>
4851 </spa>

```

14.16 translator 宏包的中文字典

4852 `<*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。

```

4853 <*theorem>
4854 \providetranslation{Comments}{评论}
4855 \providetranslation{comments}{评论}
4856 \providetranslation{Comment}{评论}
4857 \providetranslation{comment}{评论}
4858 \providetranslation{Corollaries}{推论}
4859 \providetranslation{corollaries}{推论}
4860 \providetranslation{Corollary}{推论}
4861 \providetranslation{corollary}{推论}
4862 \providetranslation{Definitions}{定义}
4863 \providetranslation{definitions}{定义}
4864 \providetranslation{Definition}{定义}
4865 \providetranslation{definition}{定义}
4866 \providetranslation{Examples}{例}
4867 \providetranslation{examples}{例}
4868 \providetranslation{Example}{例}
4869 \providetranslation{example}{例}
4870 \providetranslation{Exercises}{练习}
4871 \providetranslation{exercises}{练习}
4872 \providetranslation{Exercise}{练习}
4873 \providetranslation{exercise}{练习}
4874 \providetranslation{Facts}{事实}
4875 \providetranslation{facts}{事实}
4876 \providetranslation{Fact}{事实}
4877 \providetranslation{fact}{事实}
4878 \providetranslation{Key Lemmas}{关键引理}
4879 \providetranslation{key lemmas}{关键引理}
4880 \providetranslation{Key Lemma}{关键引理}
4881 \providetranslation{key lemma}{关键引理}
4882 \providetranslation{Key Observations}{关键观察}
4883 \providetranslation{key observations}{关键观察}
4884 \providetranslation{Key Observation}{关键观察}
4885 \providetranslation{key observation}{关键观察}
4886 \providetranslation{Lemmas}{引理}
4887 \providetranslation{lemmas}{引理}
4888 \providetranslation{Lemma}{引理}
4889 \providetranslation{lemma}{引理}
4890 \providetranslation{Main Theorems}{主要定理}
4891 \providetranslation{main theorems}{主要定理}
4892 \providetranslation{Main Theorem}{主要定理}
4893 \providetranslation{main theorem}{主要定理}
4894 \providetranslation{Observations}{观察}
4895 \providetranslation{observations}{观察}
4896 \providetranslation{Observation}{观察}
4897 \providetranslation{observation}{观察}
4898 \providetranslation{Problems}{问题}
4899 \providetranslation{problems}{问题}
4900 \providetranslation{Problem}{问题}
4901 \providetranslation{problem}{问题}
4902 \providetranslation{Proofs}{证明}
4903 \providetranslation{proofs}{证明}
4904 \providetranslation{Proof}{证明}
4905 \providetranslation{proof}{证明}
4906 \providetranslation{Proof Sketch}{证明提要}
4907 \providetranslation{Proof sketch}{证明提要}
4908 \providetranslation{proof sketch}{证明提要}
4909 \providetranslation{Proof Sketches}{证明提要}
4910 \providetranslation{Proof sketches}{证明提要}
4911 \providetranslation{proof sketches}{证明提要}
4912 \providetranslation{Sketch of Proof}{证明提要}
4913 \providetranslation{Sketch of Proofs}{证明提要}

```

```

4914 \providetranslation{Sketch of proof}{证明提要}
4915 \providetranslation{Sketch of proofs}{证明提要}
4916 \providetranslation{sketch of proof}{证明提要}
4917 \providetranslation{sketch of proofs}{证明提要}
4918 \providetranslation{Propositions}{命题}
4919 \providetranslation{propositions}{命题}
4920 \providetranslation{Proposition}{命题}
4921 \providetranslation{proposition}{命题}
4922 \providetranslation{Remarks}{注}
4923 \providetranslation{remarks}{注}
4924 \providetranslation{Remark}{注}
4925 \providetranslation{remark}{注}
4926 \providetranslation{Solutions}{解}
4927 \providetranslation{solutions}{解}
4928 \providetranslation{Solution}{解}
4929 \providetranslation{solution}{解}
4930 \providetranslation{Theorems}{定理}
4931 \providetranslation{theorems}{定理}
4932 \providetranslation{Theorem}{定理}
4933 \providetranslation{theorem}{定理}
4934 </theorem>
4935 </dict>

```

14.17 ctexcap 宏包

4936 <*ctexcap>

ctexcap 是过时宏包。

```

4937 \clist_new:N \l__ctex_ctexcap_options_clist
4938 \clist_set:Nx \l__ctex_ctexcap_options_clist
4939 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
4940 \msg_new:nnn { ctexcap } { deprecated }
4941 {
4942   Package~`ctexcap'~is~deprecated.\\
4943   Please~use~package~`ctex'~with~option~`#1'~instead: \\\
4944   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4945 }
4946 \msg_warning:nnx { ctexcap } { deprecated }
4947 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

4948 \PassOptionsToPackage { heading = true } { ctexcap }
4949 \RequirePackageWithOptions { ctex }
4950 </ctexcap>

```

14.18 ctexhook 宏包

4951 <*ctexhook>

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4952 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4953 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
4954 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4955 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
4956 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4957 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4958 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4959 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4960 \cs_set_nopar:Npx \document
4961 {
4962   \CTEX@document@left@hook
4963   \exp_not:o { \document }
4964   \CTEX@document@right@hook
4965 }
4966 \tl_new:N \g__ctex_end_preamble_hook_tl

```

```
4967 \tl_new:N \g__ctex_after_end_preamble_hook_tl
```

`\ctex_at_end_package:nn` 与 `filehook` 的 `\AtEndOfPackageFile*` 类似, 如果原来没有在载入宏包则在宏包末尾执行语句, 否则立即执行。

```
4968 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4969 {
4970   \@ifpackageloaded {#1}
4971     {#2}
4972     { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
4973 }
```

`\ctex_gadd_hook:Nn` 给钩子附加内容。

```
\ctex_gadd_hook:cn
4974 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4975 {
4976   \tl_if_exist:NF #1 { \tl_new:N #1 }
4977   \tl_gput_right:Nn #1 {#2}
4978 }
4979 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }
```

`\ctex_package_end_hook:n` 宏包末尾钩子, 只执行一次, 用后清除。
`\ctex_package_end_hook:o`

```
4980 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
4981 {
4982   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
4983     { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
4984 }
4985 \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。

```
4986 \tl_put_left:Nn \@popfilename
4987 {
4988   \cs_if_eq:NNT \@currentx \@pkgextension
4989     { \ctex_package_end_hook:o { \@currname } }
4990 }
4991 </ctexhook>
```

14.19 ctexpatch 宏包

```
4992 <*ctexpatch>
```

```
4993 \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。

```
4994 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
4995 {
4996   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4997     { once } {#2} { \use_i:nn } { \use_ii:nn }
4998 }
```

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

```
4999 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5000 {
5001   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5002     { all } {#2} { \use_i:nn } { \use_ii:nn }
5003 }
```

`\ctex_patch_cmd:Nnn` 快捷方式, 在补丁的时候关闭 `LATEX3` 语法和设置 `@` 为字母类, 补丁失败时给出警告。

```
5004 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5005 {
```

```

5006 \ctex_patch_boot:NNnnTF \_ctex_patch_cmd:Nnnnw #1
5007 { once }
5008 {
5009 \ExplSyntaxOff
5010 \char_set_catcode_letter:n { 64 }
5011 }
5012 { }
5013 { \ctex_patch_failure:N #1 }
5014 }
5015 \cs_new_protected:Npn \ctex_patch_failure:N #1
5016 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5017 \msg_new:nnn { ctex } { patch-failure }
5018 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

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

```

5019 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5020 {
5021 \ctex_patch_boot:NNnnTF \_ctex_hookto_cmd:Nnnnw #1
5022 { left } {#2} { \use_i:nn } { \use_ii:nn }
5023 }

```

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

```

5024 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5025 {
5026 \ctex_patch_boot:NNnnTF \_ctex_hookto_cmd:Nnnnw #1
5027 { right } {#2} { \use_i:nn } { \use_ii:nn }
5028 }

```

`\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。

```

5029 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
5030 {
5031 \tl_set:Nn \_ctex_patch_true:w {#5}
5032 \tl_set:Nn \_ctex_patch_false:w {#6}
5033 \group_begin:
5034 \char_set_catcode_other:n { 35 }
5035 \ctex_parse_name:NN #1 #2 {#3} {#4}
5036 }
5037 \tl_new:N \_ctex_patch_true:w
5038 \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。

```

5039 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5040 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5041 \group_begin:
5042 \cs_set_protected:Npn \_ctex_tmp:w #1#2#3

```

```

5043 {
5044   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5045   {
5046     \bool_if:nTF
5047     {
5048       \cs_if_exist_p:c { ##3 ~ } ||
5049       \cs_if_exist_p:c { #1##3 }
5050     }
5051     {
5052       \group_begin:
5053       \use:x
5054       {
5055         \group_end:
5056         \__ctex_parse_name:nNNNnN
5057         { \token_get_replacement_spec:N ##2 }
5058         \exp_not:N ##2
5059         \exp_not:c { ##3 ~ }
5060         \exp_not:c { #1##3 }
5061       } {##3} ##1
5062     }
5063     { ##1##2 }
5064   }
5065   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5066   {
5067     \exp_args:Nc ##6
5068     {
5069       \str_case:nnTF {##1}
5070       {
5071         { \protect ##3 } { }
5072         { \x@protect ##2 \protect ##3 } { }
5073       }
5074       {
5075         \str_if_eq_x:nnTF
5076         { \exp_not:n { #1@protected@ ##3 #1##3 } }
5077         {
5078           \exp_last_unbraced:Nf \__ctex_parse_name:w
5079           \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
5080         }
5081         { #1##5 ~ } { ##5 ~ }
5082       }
5083     }
5084     \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5085     {
5086       { #1@protected@ ##2 ##4 } { }
5087       { #1@ ##4 } { }
5088     }
5089     { #1##5 } {##5}
5090   }
5091 }
5092 }
5093 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5094 }
5095 \use:x
5096 {
5097   \__ctex_tmp:w
5098   { \c_backslash_str }
5099   { \c_left_brace_str }
5100   { \tl_to_str:n { testopt } }
5101 }
5102 \group_end:
5103 \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。

```

5104 \str_new:N \l__ctex_prefix_str
5105 \str_new:N \l__ctex_parameter_str
5106 \str_new:N \l__ctex_replacement_str

```


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

```

5107 \group_begin:
5108   \cs_set_protected:Npn \__ctex_tmp:w #1
5109     {
5110       \prg_new_protected_conditional:Npnn
5111         \ctex_get_macro_meaning:N ##1 { TF }
5112         {
5113           \exp_after:wN \__ctex_get_macro_meaning:w
5114           \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5115         }
5116       \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5117         ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5118         {
5119           \tl_if_empty:nTF { ##4 }
5120             { \prg_return_false: }
5121             {
5122               \str_set:Nn \l__ctex_prefix_str { ##1 }
5123               \str_set:Nn \l__ctex_parameter_str { ##2 }
5124               \str_set:Nn \l__ctex_replacement_str { ##3 }
5125               \prg_return_true:
5126             }
5127         }
5128     }
5129   \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5130 \group_end:

```

检查宏是否可以重建。

```

5131 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5132   {
5133     \ctex_get_macro_meaning:NNTF #1
5134     {
5135       \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5136       \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5137     }
5138     {#4}
5139   }
5140 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

```

使用 `\tl_rescan:n` 来重新记号化 `\meaning` 字符串。

```

5141 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5142   {
5143     \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
5144     \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
5145     \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5146     \use:x
5147     {
5148       \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5149       \exp_not:o { \l__ctex_parameter_tl }
5150       { \exp_not:o { \l__ctex_replacement_tl } }
5151     }
5152   }
5153 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5154   {
5155     \str_if_empty:NNTF #2
5156       { \tl_clear:N #1 }
5157       { \tl_set_rescan:Nno #1 {#3} {#2} }
5158   }
5159 \tl_new:N \l__ctex_prefix_tl
5160 \tl_new:N \l__ctex_parameter_tl
5161 \tl_new:N \l__ctex_replacement_tl

```

对宏的替换文本进行字符串替换,然后重建。

```

5162 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5

```

`\ctex_get_macro_meaning:NNTF`
`__ctex_get_macro_meaning:w`

`\ctex_if_rescanable:NnTF`

`__ctex_patch_rebuild:Nn`

`__ctex_patch_cmd:Nnnnw`

```

5163 {
5164   \group_end:
5165   \ctex_if_rescanable:NnTF #1 {#3}
5166   {
5167     \use:x
5168     {
5169       \__ctex_patch_replace:nnnTF {#2}
5170       { \tl_to_str:n {#4} }
5171       { \tl_to_str:n {#5} }
5172     }
5173     {
5174       \__ctex_patch_rebuild:Nn #1 {#3}
5175       \__ctex_patch_true:w
5176     }
5177     { \__ctex_patch_false:w }
5178   }
5179   { \__ctex_patch_false:w }
5180 }

```

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

```

5181 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5182 {
5183   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5184   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5185 }

```

`__ctex_hookto_cmd:Nnnnw` 在宏的前/后附加钩子。

```

5186 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5187 {
5188   \group_end:
5189   \ctex_get_macro_meaning:NnTF #1
5190   {
5191     \str_if_empty:NnTF \l__ctex_parameter_str
5192     { \__ctex_hookto_cmd_parameterless:Nnnnw }
5193     { \__ctex_hookto_cmd_parameter:Nnnnw }
5194     #1 {#2} {#3} {#4}
5195   }
5196   { \__ctex_patch_false:w }
5197 }

```

`__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。

```

5198 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5199 {
5200   \str_if_empty:NnTF \l__ctex_prefix_str
5201   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5202   \tex_edef:D #1
5203   {
5204     \use:c { __ctex_ #2 _hook_aux:nn }
5205     { \exp_not:o {#1} }
5206     { \exp_not:n {#4} }
5207   }
5208   \__ctex_patch_true:w
5209 }
5210 \cs_generate_variant:Nn \tl_rescan:nn { no }
5211 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5212 \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。

```

5213 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5214 {
5215   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5216   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5217   {

```

```
5218         \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}  
5219         \__ctex_patch_rebuild:Nn #1 {#3}  
5220         \__ctex_patch_true:w  
5221     }  
5222     { \__ctex_patch_false:w }  
5223 }  
  
5224 </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。 99
<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。 84
<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。 84
<code>nozhmap</code> 是过时选项。 39		给 <code>enumitem</code> 宏包注册 <code>\chinese</code> 和 <code>\zhnum</code> 。 79
<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。 129
<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。 99
<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。 109
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。 14, 42		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。 130
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。 109		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。 81
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。 111
新增 <code>zihao</code> 选项。 37		正确更新 <code>CJK</code> 包的 <code>\CJKfamilydefault</code> 。 69
新增统一设置接口 <code>\ctexset</code> 。 42		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。 131
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。 105		正确更新 <code>up\LaTeX</code> 的 <code>\CJKfamilydefault</code> 。 69
v2.1 (2015/05/18 – 2015/06/19)		<code>\ccwd</code> : 正确设置 <code>up\TeX</code> 下的 <code>\ccwd</code> 。 73
General: <code>format+</code> , <code>nameformat+</code> 等带加号的选项, 加号与前面的文字之间可以有可选的空格。 82		<code>\chinese</code> : 支持 <code>\pagenumbering</code> 。 79
<code>nameformat</code> 可以接受章节名字为参数。 81		<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 CJK 字体族加上 <code>CMap</code> 。 43		

v2.4.2	(2016/05/15)	改进 hyperref 宏包的标题锚点设置。	92
General: 恢复 luatexja 对 \emshape 和 \eminnershape 的		解决 zhmap 文件的 \catcode 问题。	44
重定义。	49	使用 titlesec 时, 章节目录也使用 CT_EX 的编号。	94
\em : 兼容 upL_AT_EX 2016/05/07u00 的定义。	67	提供 \CTEXifname 。	84
v2.4.3	(2016/06/03 – 2016/08/26)	提供 \partmark 。	84
General: 更新 unicode-math 补丁。	49	新的标题格式选项 break 。	82
简化 fontspec 补丁。	49	v2.4.5	(2016/10/01 – 2016/10/25)
确保 \proofname 非空。	80	General: 新的标题格式选项 chapter/lofskip 和	
删除选项 part/fixbeforeskip 和		chapter/lotskip 。	83
chapter/fixbeforeskip 。	83	新的标题格式选项 hang 。	82
新的标题格式选项 fixskip 。	82	新的标题格式选项 tocline 。	82
v2.4.4	(2016/09/09 – 2016/09/19)	\ps@headings : 修复补丁失败。	96
General: 不再默认设置 xeCJK 的伪粗体。	48		

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	A
_	\abovedisplayshortskip
\#	\abovedisplayskip
.../afterindent	3761, 3768, 3772, 3779, 3792, 3795, 3800, 3807, 3811, 3818
.../aftername	2239
.../aftername+	abstractname
.../afterskip	1197, 1220
.../aftertitle	\addCJKfontfeatures
.../aftertitle+	1200
.../beforeskip	\addcontentsline
.../break	2463, 2509, 2513,
.../break+	2518, 2546, 2550, 2555, 2644, 2649, 2655, 2661, 2784, 2798
.../fixskip	\AddEnumerateCounter
.../format	2194, 2196, 2197, 2198
.../format+	\addpenalty
.../indent	2743, 3389, 3422, 3456, 3470, 3484
.../name	\addtocontents
.../nameformat	2451, 2457, 2665, 2666
.../nameformat+	\addvspace
.../number	2452, 2458, 2470, 2473,
.../numberformat	2665, 2666, 2743, 2747, 3210, 3220, 3228, 3238, 3246, 3256
.../numberformat+	adobefonts
.../titleformat	\algorithmname
.../titleformat+	2246, 2262
.../tocline	algorithmname
\	16
\	AlternateFont
\	26, 1320
\	\appendix
\	2923
\	appendix/name
\	23, 2912
\	appendix/number
\	23, 2912
\	appendix/numbering
\	23, 2912
\	\appendixname
\	2241, 3534
\	appendixname
\	12
\	\arabic
\	3436
\	\AtBeginDocument
\	592, 1613
\	\AtBeginDvi
\	449
\	\AtBeginShipoutFirst
\	451, 1824, 1839
\	\AtBeginUTFCommand
\	733
\	\AtEndOfClass
\	135
\	\AtEndOfPackage
\	136
\	\AtEndUTFCommand
\	739
\	autoindent
\	10, 184, 2049

B

`\baselinestretch` 2104
`\begin` 3216, 3234, 3252
`\begingroup` 2775, 2811, 2843, 3206, 3224, 3242, 4442, 4475, 4482
`\belowdisplayshortskip` . 3755, 3763, 3774, 3794, 3802, 3813
`\belowdisplayskip` 3756, 3768, 3779, 3795, 3807, 3818
`\bfdefault` 978, 984, 1310, 1317
`\bfseries` 2525, 2528, 2564, 2567, 2580,
2597, 2693, 2698, 2716, 3317, 3319, 3323, 3336, 3337,
3340, 3365, 3367, 3373, 3391, 3395, 3416, 3450, 3467, 3481
`\bibname` 2244, 2247, 2255, 2263, 2269, 2270
`bibname` 12
bool commands:
`\bool_gset_false:N` 265, 270
`\bool_gset_true:N` 260
`\bool_if:NTF` 70, 1155, 2857, 3017, 3027, 3038, 3553
`\bool_if:nTF` 5046
`\bool_new:N` 131, 1192
`\bool_set_false:N` 1167
`\bool_set_true:N` 1185
`\c_false_bool` 2873, 2875
`\c_true_bool` 2872, 2874
box commands:
`\box_ht:N` 3834
`\box_new:N` 60
`\box_wd:N` 1910, 3130

C

`c5size` 29
`cap` 29
`captiondelimiter` 30, 3841
`\catcode` 4442, 4443, 4444, 4445, 4449, 4450
`CCTF` 29
`CCTfont` 29
`\ccwd` 25, 115, 188, 1906, 1956,
1957, 1958, 2055, 2093, 2112, 2113, 2120, 2147, 2148,
2149, 2152, 2154, 2155, 2166, 2177, 2178, 3851, 3852, 3853
`\centering` 2561, 2595, 3207,
3225, 3243, 3323, 3335, 3340, 3344, 3373, 3395, 3403, 3438
`\chapter` 2621, 3559
`chapter/beforeskip` 31
`chapter/fixbeforeskip` 31
`chapter/lofskip` 21, 2392
`chapter/lotskip` 21, 2392
`chapter/numbering` 15
`chapter/pagestyle` 21, 2392
`\chaptermark` 2664
`\chaptername` 3362
char commands:
`\char_set_catcode:nn` 73, 529, 530, 565, 566
`\char_set_catcode_active:N` 1345
`\char_set_catcode_comment:n` 524
`\char_set_catcode_escape:n` 525
`\char_set_catcode_letter:n` 67, 1616, 3148, 5010
`\char_set_catcode_math_superscript:n` 560
`\char_set_catcode_other:N` . 491, 911, 1328, 3076, 3134

`\char_set_catcode_other:n` 559, 5034
`\char_value_catcode:n` 73, 529, 530, 565, 566
`CharRange` 26, 1320
`\Chinese` 30, 2190
`\chinese` 25, 2188, 2198, 3310, 3372
`\CJK` 1875
`\CJKbold`
3964, 3965, 3975, 3976, 3990, 3991, 3993, 3994, 3996,
3997, 4001, 4002, 4004, 4005, 4007, 4008, 4020, 4021,
4023, 4024, 4026, 4027, 4031, 4032, 4034, 4035, 4037, 4038
`\CJKfamily` 539, 543, 547, 551, 1052, 1074, 1254, 1258, 1262,
1266, 1414, 1635, 4430, 4431, 4432, 4433, 4435, 4436, 4438
`\CJKfamilydefault`
... 533, 534, 551, 607, 1248, 1249, 1266, 1274, 1276,
1280, 1287, 1300, 1631, 1640, 1714, 1718, 1726, 1739, 1741
`\CJKfilltwosides` 1883
`\CJKfontspec` 1200
`\CJKglue` 1909, 1923, 1940, 1953
`\CJKhook` 622
`\CJKnormal` 3957, 3958, 3959, 3960, 3961, 3962, 3963,
3968, 3969, 3970, 3971, 3972, 3973, 3974, 3989, 3992,
3995, 4000, 4003, 4006, 4019, 4022, 4025, 4030, 4033, 4036
`\CJKpunctmapfamily`
. 655, 656, 657, 658, 659, 660, 664, 665, 666, 667, 668, 669
`\CJKrmdefault` 514,
534, 535, 539, 1122, 1133, 1223, 1249, 1250, 1254, 1276,
1279, 1280, 1628, 1631, 1632, 1722, 4086, 4088, 4089,
4094, 4095, 4096, 4099, 4176, 4179, 4180, 4185, 4186,
4187, 4254, 4257, 4258, 4264, 4265, 4266, 4331, 4334,
4335, 4342, 4343, 4344, 4392, 4395, 4396, 4401, 4402, 4403
`\CJKsfdefault` 515, 536,
543, 1123, 1134, 1229, 1251, 1258, 1629, 1633, 1723,
4070, 4076, 4077, 4100, 4177, 4255, 4259, 4332, 4336, 4393
`\CJKtilde` 608
`\CJKttdefault` 516, 537, 547, 1124, 1135, 1234, 1252,
1262, 1630, 1634, 1724, 4087, 4101, 4178, 4256, 4333, 4394
`clearalternatefont` 27, 1506
`\cleardoublepage` 2484, 2622, 3333, 3360
`\clearpage` 2486, 2622, 3333, 3360
clist commands:
`\clist_clear:N` 999, 4779
`\clist_concat:NNN` 1021
`\clist_const:Nn` 635
`\clist_gput_right:Nn` 382, 394, 404, 406
`\clist_gset:Nn` 1240
`\clist_if_empty:NTF` 1333, 1352, 1592
`\clist_map_break:n` 2996, 3544
`\clist_map_function:nN` 1509, 1511, 1543, 1557
`\clist_map_inline:Nn` 625, 1445
`\clist_map_inline:nn`
. 367, 647, 1550, 1867, 1893, 2991, 3541, 3681, 3716, 4768
`\clist_new:N` 138, 1018, 1241, 1387, 4797, 4937
`\clist_put_left:Nn` 1022, 3610, 3636
`\clist_put_right:Nn` 1184, 1433, 4791
`\clist_set:Nn` 1003, 1370, 1371, 4938
`\clist_use:Nn` 4947

- \clubpenalty 2842, 2849
- \contentsname 2234
- contentsname 11
- continuation 12
- CS commands:
 - \cs:w 3281
 - \cs_end: 3281
 - \cs_generate_variant:Nn
 - 93, 467, 479, 963, 1086, 1100, 1173, 1193,
 - 1194, 1366, 1403, 1505, 1543, 1551, 4979, 4985, 5103, 5210
 - \cs_gset_eq:NN
 - 161, 162, 163, 261, 266, 271, 309, 311, 377,
 - 381, 499, 806, 810, 1160, 1163, 2960, 2961, 2966, 3616, 3643
 - \cs_gset_nopar:Npx 807
 - \cs_gset_protected_nopar:Npn 2924, 2972
 - \cs_gset_protected_nopar:Npx
 - 845, 1048, 1407, 1501, 2900
 - \cs_if_eq:NNTF 947, 2958, 4988, 5136, 5216
 - \cs_if_exist:NTF 44, 358, 476, 756, 801, 813,
 - 843, 881, 1406, 1498, 1974, 3062, 3268, 3559, 3561, 4993
 - \cs_if_exist_p:N 5048, 5049
 - \cs_if_exist_use:N 1070, 1071
 - \cs_if_exist_use:NTF 4, 784, 1092, 1520, 4982
 - \cs_if_free:NTF 461, 471, 627, 642,
 - 1157, 1309, 2068, 2082, 2194, 3010, 3055, 3101, 3106, 3110
 - \cs_new:Npn 110, 117, 124,
 - 902, 908, 912, 995, 1073, 2876, 2883, 2909, 3035, 5093, 5211
 - \cs_new_eq:NN
 - 6, 7, 8, 9, 20, 23, 95, 106, 107, 108, 133, 134,
 - 152, 359, 468, 612, 673, 822, 847, 923, 954, 1220, 1226,
 - 1735, 1888, 1889, 1890, 1902, 1953, 1964, 1965, 2189,
 - 2190, 2191, 2437, 2440, 2870, 2871, 2872, 2873, 2874,
 - 2875, 2923, 2948, 2949, 3286, 3557, 3562, 4773, 5140, 5212
 - \cs_new_nopar:Npn
 - 862, 1065, 1118, 1129, 2188, 2890, 3099, 3104, 4798
 - \cs_new_protected:Npn 62,
 - 746, 996, 1015, 1331, 1422, 1436, 1587, 1959, 2385,
 - 2462, 3124, 3736, 3742, 4774, 4952, 4954, 4968, 4974,
 - 4994, 4999, 5004, 5015, 5019, 5024, 5029, 5039, 5044,
 - 5065, 5116, 5131, 5141, 5153, 5162, 5181, 5186, 5198, 5213
 - \cs_new_protected_nopar:Npn 77, 79, 135, 136,
 - 447, 459, 469, 474, 480, 512, 520, 555, 572, 614, 620,
 - 623, 640, 645, 653, 662, 671, 782, 791, 804, 809, 811,
 - 823, 837, 864, 879, 892, 916, 925, 942, 956, 973, 1019,
 - 1027, 1046, 1066, 1076, 1101, 1139, 1151, 1153, 1175,
 - 1177, 1270, 1295, 1304, 1350, 1361, 1388, 1397, 1404,
 - 1410, 1430, 1443, 1462, 1476, 1483, 1495, 1515, 1531,
 - 1549, 1552, 1554, 1567, 1579, 1621, 1653, 1666, 1680,
 - 1712, 1744, 1762, 1777, 1793, 1804, 1808, 1845, 1852,
 - 1856, 1879, 1885, 1906, 1920, 1955, 1989, 2067, 2069,
 - 2094, 2129, 2139, 2144, 2298, 2328, 2421, 2428, 2435,
 - 2441, 2443, 2446, 2855, 2861, 2940, 2951, 2983, 3003,
 - 3008, 3013, 3023, 3033, 3077, 3264, 3273, 3278, 3654,
 - 3669, 3701, 3706, 3856, 3860, 4765, 4789, 4956, 4958, 4980
 - \cs_new_protected_nopar:Npx ... 484, 1206, 1872, 3064
 - \cs_set:Npn 1329, 2357
 - \cs_set_eq:NN 109, 189, 194, 199, 552, 570,
 - 590, 617, 621, 629, 643, 763, 816, 817, 924, 955, 993,
 - 1052, 1267, 1406, 1414, 1424, 1425, 1426, 1498, 1499,
 - 1500, 1624, 1717, 1970, 2205, 2210, 2213, 2218, 2442,
 - 2444, 2863, 2864, 2865, 2866, 2867, 2868, 2955, 2987, 2997
 - \cs_set_nopar:Npn 1652, 1688, 3116
 - \cs_set_nopar:Npx 1903, 3025, 4960
 - \cs_set_protected:Npn 740, 749, 5042, 5108
 - \cs_set_protected_nopar:Npn
 - 616, 764, 1923, 3057, 3114, 3743
 - \cs_set_protected_nopar:Npx 1961
 - \cs_to_str:N 54, 1205, 5040
 - \cs_undefine:N 1031, 1032, 1036, 1037, 1427, 4983
 - cs4size 29
 - \csname 2744, 2753, 2754, 2755, 2762, 2780, 2781, 2783, 2786,
 - 2795, 2796, 2797, 2800, 4474, 4476, 4477, 4478, 4483, 4487
 - \CTeX 25, 3837
 - ctex commands:
 - \ctex_add_cmap:n 464, 469
 - \ctex_add_cmap:Nn 472, 474, 479
 - \ctex_add_to_selectfont:n 1959, 1980, 2008, 2081
 - \ctex_after_end_preamble:n 36, 109, 4952
 - \ctex_appto_cmd:NnnTF 495, 1615, 5024
 - \ctex_assign_heading_name:nn 2334, 2382, 2916
 - \ctex_at_end:n 135, 293, 356, 1953, 3923
 - \ctex_at_end_package:nn
 - 130, 450, 498, 754, 771, 1818, 1820, 1838, 2192, 2970,
 - 3043, 3050, 3109, 3111, 3112, 3145, 3183, 3292, 3839, 4968
 - \ctex_at_end_preamble:n 361, 597, 604, 1269, 1641, 4952
 - \ctex_auto_ignorespaces: 611, 621, 2018
 - \ctex_CJK_input:n 555
 - \ctex_declare_math_sizes:nnnn 3701, 3735
 - \ctex_default_pt:n 113, 117
 - \ctex_detected_platform: 1743, 3864, 3873
 - \ctex_family_cmap:nn 459, 468
 - \ctex_file_input:n 77, 397,
 - 674, 2048, 3568, 3603, 3608, 3613, 3614, 3629, 3634,
 - 3640, 3641, 3745, 3747, 3865, 3869, 3877, 3923, 4048, 4049
 - \ctex_file_wrapper:nnn 62, 78, 81, 522, 557
 - \ctex_fix_varioref_label:n 3276, 3278
 - \ctex_fontset_error:n
 - ... 3856, 4155, 4189, 4223, 4268, 4300, 4303, 4346, 4405
 - \ctex_gadd_hook:Nn 622, 4972, 4974
 - \ctex_get_macro_meaning:N 5111
 - \ctex_get_macro_meaning:NTF 5107, 5133, 5189
 - \ctex_hypersetup:n
 - 70, 359, 1801, 1813, 1816, 1831, 1834, 1837
 - \ctex_if_autoindent_touched:TF 133, 189, 194, 199, 3595
 - \ctex_if_ccglue_touched:TF 1937, 2100
 - \ctex_if_ccglue_touched_p: 1937
 - \ctex_if_macosx:TF 1784, 1785, 1787, 1793
 - \ctex_if_preamble:TF 108, 3892
 - \ctex_if_rescanable:NnTF 5131, 5165
 - \ctex_ignorespaces_case:N 614, 2015, 2018, 2021
 - \ctex_load_fontset: 3860, 3900, 3922
 - \ctex_load_zhmap:nnnn 512, 4092, 4183, 4262, 4340, 4399

- \ctex_ltj_add_font_features:n [1175](#), [1217](#)
- \ctex_ltj_add_font_features:nn [1175](#), [1176](#)
- \ctex_ltj_char_range_key:nn [1584](#), [1587](#)
- \ctex_ltj_clear_alterate_font:n [1509](#), [1515](#)
- \ctex_ltj_declare_alterate_shape:nnnnnn [1416](#), [1436](#)
- \ctex_ltj_declare_char_range:n [1546](#), [1549](#), [1551](#)
- \ctex_ltj_declare_char_range:nn [1553](#), [1554](#)
- \ctex_ltj_def_char_range_key:n [1559](#), [1579](#)
- \ctex_ltj_ensure_default_family: [1270](#), [1731](#)
- \ctex_ltj_extract_font: [816](#), [823](#)
- \ctex_ltj_family_if_exist:nN [1087](#)
- \ctex_ltj_family_if_exist:nNTF
..... [1078](#), [1274](#), [1279](#), [1297](#), [1300](#), [1518](#)
- \ctex_ltj_family_if_exist:nTF [1087](#)
- \l_ctex_ltj_family_tl
[1080](#), [1085](#), [1142](#), [1143](#), [1158](#), [1162](#), [1165](#), [1176](#), [1512](#), [1513](#)
- \ctex_ltj_fontspec:nn [1139](#), [1151](#), [1186](#), [1212](#)
- \ctex_ltj_get_and_define_fonts:nN [916](#)
- \ctex_ltj_get_and_define_fonts_al:nN [920](#), [923](#)
- \ctex_ltj_get_and_define_fonts_ja:nN [919](#), [925](#)
- \ctex_ltj_if_alterate_shape_exist:nTF
..... [826](#), [839](#), [857](#), [866](#)
- \ctex_ltj_if_jfont:nTF [55](#), [902](#), [913](#), [918](#)
- \ctex_ltj_if_jfont_math:NTF [908](#), [949](#)
- \ctex_ltj_math_group_hook: [54](#), [764](#), [948](#), [954](#)
- \ctex_ltj_pickup_font: ... [797](#), [811](#), [886](#), [928](#), [930](#), [932](#)
- \ctex_ltj_reset_alterate_font:n [1511](#), [1531](#)
- \ctex_ltj_save_alterate_family:Nnnn [1381](#), [1401](#), [1404](#)
- \ctex_ltj_save_alterate_seq:Nn [1356](#), [1361](#)
- \ctex_ltj_save_alterate_seq:Nnnwn [1361](#)
- \ctex_ltj_save_alterate_seq:Nnnwnw [1364](#), [1367](#)
- \ctex_ltj_save_char_range:n [1557](#), [1564](#)
- \ctex_ltj_save_char_range:nn [1566](#), [1567](#)
- \ctex_ltj_select_alterate_font: [864](#), [1983](#)
- \ctex_ltj_select_font: [782](#), [1982](#)
- \ctex_ltj_set_alterate_family:nn [1359](#), [1388](#)
- \ctex_ltj_set_alterate_family:nnn . [1400](#), [1408](#), [1410](#)
- \ctex_ltj_set_alterate_family:Nnnnn [1392](#), [1397](#), [1403](#)
- \ctex_ltj_set_alterate_family:nnTF [1004](#), [1350](#)
- \ctex_ltj_set_alterate_seq:n [1322](#), [1327](#)
- \ctex_ltj_set_alterate_shape:n ... [1464](#), [1476](#), [1487](#)
- \ctex_ltj_set_alterate_shape:nnN [1449](#), [1483](#)
- \ctex_ltj_set_alterate_shape:nnn [1452](#), [1459](#)
- \ctex_ltj_set_alterate_shape:nnnn [1461](#), [1462](#)
- \ctex_ltj_set_alterate_shape:Nnnnnnn .. [1439](#), [1443](#)
- \ctex_ltj_set_family:nnn
.... [996](#), [1015](#), [1170](#), [1201](#), [1208](#), [1223](#), [1229](#), [1234](#), [1238](#)
- \ctex_ltj_set_math_letter:NN [746](#), [758](#), [766](#)
- \ctex_ltj_subst_font: [817](#), [837](#)
- \ctex_ltj_swap_cs:NN [1415](#), [1422](#)
- \ctex_ltj_switch_family:n
..... [1075](#), [1076](#), [1086](#), [1143](#), [1171](#), [1207](#)
- \ctex_ltj_update_mathfont: [1292](#), [1295](#)
- \ctex_ltj_update_mathfont:n [1298](#), [1301](#), [1304](#)
- \ctex_ltj_use_math_group:Nn [763](#), [942](#)
- \ctex_make_spa:nn [4765](#)
- \ctex_mono_jfm:n [956](#), [964](#), [2038](#)
- \ctex_package_end_hook:n [4980](#), [4989](#)
- \ctex_parse_name:NN [5035](#), [5039](#)
- \ctex_parse_name:NNn [5040](#), [5044](#), [5103](#)
- \ctex_patch_boot:NnnTF
..... [4996](#), [5001](#), [5006](#), [5021](#), [5026](#), [5029](#)
- \ctex_patch_cmd:Nnn
.... [1648](#), [1690](#), [3079](#), [3085](#), [3091](#), [3157](#), [3160](#), [3164](#),
[3170](#), [3178](#), [3186](#), [3189](#), [3194](#), [3197](#), [3851](#), [3852](#), [3853](#), [5004](#)
- \ctex_patch_cmd_all:NnnTF [4999](#)
- \ctex_patch_cmd_once:NnnTF [1642](#), [4994](#)
- \ctex_patch_failure:N [501](#), [541](#), [545](#), [549](#), [553](#),
[1256](#), [1260](#), [1264](#), [1268](#), [1619](#), [1972](#), [3141](#), [3151](#), [5013](#), [5015](#)
- \ctex_plane_to_utfxvibe:Nn [571](#)
- \ctex_preto_cmd:NnnTF [538](#), [542](#),
[546](#), [550](#), [1253](#), [1257](#), [1261](#), [1265](#), [1966](#), [3138](#), [3147](#), [5019](#)
- \ctex_punct_map_bfseries:nn
..... [645](#), [4077](#), [4089](#), [4095](#), [4180](#),
[4186](#), [4258](#), [4259](#), [4265](#), [4335](#), [4336](#), [4337](#), [4343](#), [4396](#), [4402](#)
- \ctex_punct_map_family:nn [640](#), [4070](#), [4076](#),
[4086](#), [4087](#), [4094](#), [4176](#), [4177](#), [4178](#), [4185](#), [4254](#), [4255](#),
[4256](#), [4264](#), [4331](#), [4332](#), [4333](#), [4342](#), [4392](#), [4393](#), [4394](#), [4401](#)
- \ctex_punct_map_itshape:nn [662](#),
[4088](#), [4096](#), [4179](#), [4187](#), [4257](#), [4266](#), [4334](#), [4344](#), [4395](#), [4403](#)
- \ctex_punct_map_series:nnn [649](#), [650](#), [653](#)
- \ctex_punct_set:n [623](#),
[4065](#), [4093](#), [4175](#), [4184](#), [4253](#), [4263](#), [4330](#), [4341](#), [4391](#), [4400](#)
- \ctex_punct_space:nn [671](#)
- \ctex_scheme_input:n [79](#), [3294](#), [3570](#)
- \ctex_select_size: .. [2056](#), [2063](#), [2066](#), [2067](#), [2088](#), [2142](#)
- \ctex_set_default_ccwd:Nn [110](#), [198](#), [2062](#), [2087](#)
- \ctex_set_font_size:Nnn [3736](#), [3751](#),
[3759](#), [3770](#), [3781](#), [3782](#), [3783](#), [3784](#), [3785](#), [3786](#), [3787](#),
[3790](#), [3798](#), [3809](#), [3820](#), [3821](#), [3822](#), [3823](#), [3824](#), [3825](#), [3826](#)
- \ctex_set_ignorespaces: [614](#), [622](#)
- \ctex_set_upfamily:nnn
..... [1653](#), [4112](#), [4113](#), [4114](#), [4115](#), [4116](#),
[4117](#), [4119](#), [4199](#), [4200](#), [4201](#), [4202](#), [4278](#), [4279](#), [4280](#),
[4281](#), [4355](#), [4356](#), [4357](#), [4358](#), [4359](#), [4361](#), [4414](#), [4415](#), [4416](#)
- \ctex_set_upfonts:nnnnnn
..... [1680](#), [4107](#), [4195](#), [4274](#), [4351](#), [4410](#)
- \ctex_set_upmap:nnn
... [1666](#), [1682](#), [1683](#), [1684](#), [1685](#), [4118](#), [4120](#), [4360](#), [4362](#)
- \ctex_set_zhmap:n [446](#), [517](#), [1668](#)
- \ctex_titleps_hook: [3075](#), [3110](#), [3111](#)
- \ctex_titlesec_hook: [2983](#), [3047](#)
- \ctex_update_ccglue: [1920](#), [2107](#), [2126](#), [2170](#)
- \ctex_update_ccwd: [1906](#), [2101](#), [2178](#)
- \ctex_update_default_family: [46](#), [597](#), [1269](#), [1641](#), [1711](#)
- \ctex_update_em_unit: [1955](#), [2096](#), [2146](#)
- \ctex_update_parindent: [2074](#), [2129](#), [2171](#)
- \ctex_update_size: [2068](#), [2069](#)
- \ctex_update_stretch: [2073](#), [2094](#)
- \ctex_update_xkanjiskip: [1988](#), [2008](#)
- \ctex_update_ziju: [2076](#), [2144](#)
- \ctex_varioref_hook: [3273](#), [3284](#), [3292](#)

`\ctex_zhmap_case:nnn` 134, 261, 266, 271, 4063, 4157, 4225, 4318, 4382
`\ctex_zhmap_input:n` 517, 520
`\ctex_zihao:n` 3653, 3654
`\ctex_ziju:n` 2138, 2139
ctex internal commands:
`\g_ctex_after_end_preamble_hook_tl` 4955, 4959, 4967
`__ctex_assign_heading_name:nnn` 2382
`\l_ctex_autoindent_tl` 132, 188, 193, 198, 2055, 2059, 2062, 2131, 2134
`__ctex_calc_bounds:nn` 4793, 4794, 4798
`__ctex_ccglue:` 1940, 1953
`\l_ctex_ccglue_skip` 1924, 1928, 1931, 1934, 1936, 1944, 1948, 2105, 2116, 2125, 2161, 2169
`\c_ctex_class_tl` 84, 86, 409, 414, 419, 424, 3557, 3563, 3565, 3566, 3568
`__ctex_clear_fntef_color:n` ... 1845, 1852, 1856, 1871
`\c_ctex_cmap_encoding_seq` 456, 463
`\l_ctex_ctexcap_options_clist` 4937, 4938, 4947
`__ctex_def_heading_keys:n` 2328, 2412
`__ctex_default_pt:w` 119, 124
`\l_ctex_encoding_tl` 128, 221, 222, 505, 508, 599, 607, 1814, 2044, 2186, 3598, 3625, 3639
`\g_ctex_end_preamble_hook_tl` 4953, 4957, 4966
`\c_ctex_engine_file_str` 43, 2048
`\c_ctex_engine_str` 43
`\l_ctex_family_default_init_tl` 1714, 1734
`__ctex_family_default_wrap:n` 1717, 1735, 1738
`__ctex_fix_varioref_label:n` 3278
`\g_ctex_font_size_flag` 151, 161, 162, 163, 377, 381, 403, 3615, 3616, 3642, 3643, 3744
`\c_ctex_font_size_prop` 3656, 3667, 3711, 3738
`\c_ctex_font_size_seq` 3665, 3668, 3679
`\l_ctex_font_size_tl` 3656, 3657, 3703, 3704, 3738, 3739
`\g_ctex_fontset_tl` 225, 1746, 1764, 1789, 1796, 1797, 3862, 3865, 3868, 3869, 3875, 3877, 3897, 3899, 3904, 3921
`__ctex_get_font_sizes:Nn` 3703, 3706
`__ctex_get_macro_meaning:w` 5107
`\l_ctex_heading_bool` 304, 3553
`\l_ctex_heading_skip` 2425, 2432, 2434, 2436, 2437, 2448, 2449, 2452, 2454, 2455, 2458
`\c_ctex_headings_cs_seq` 2942, 2956, 2965
`\c_ctex_headings_seq` 2288, 2409, 3060, 3275
`__ctex_hookto_cmd:Nnnnw` 5021, 5026, 5186
`__ctex_hookto_cmd_parameter:Nnnnw` 5193, 5213
`__ctex_hookto_cmd_parameterless:Nnnnw` . 5192, 5198
`__ctex_initial_heading:n` 2298, 2411
`__ctex_left_hook_aux:nn` 5211
`\l_ctex_line_spread_fp` 109, 181, 3617, 3618, 3644, 3645, 3830, 3832
`\l_ctex_line_stretch_tl` 2087, 2092, 2093, 2097
`\l_ctex_ltj_add_alternate_bool` 1155, 1167, 1185, 1192
`__ctex_ltj_alternate_cs:n` 1032, 1035, 1036, 1037, 1070, 1071, 1073, 1158, 1161, 1162, 1164, 1165, 1356, 1393, 1455, 1480, 1520, 1537
`\l_ctex_ltj_alternate_family_tl` 1418, 1421
`\l_ctex_ltj_alternate_options_clist` 1371, 1372, 1383, 1387
`\l_ctex_ltj_alternate_seq` 1000, 1340, 1341, 1349, 1355, 1363, 1611
`\l_ctex_ltj_base_CJKfamily_tl` 1001, 1017, 1068, 1455, 1480
`\l_ctex_ltj_base_family_tl` 64, 1033, 1063, 1069, 1399, 1440, 1499, 1518, 1523, 1524, 1536
`__ctex_ltj_change_encoding:` 965, 1051, 1413
`\l_ctex_ltj_char_range_clist` 999, 1324, 1333, 1352, 1370, 1382, 1394, 1417, 1439, 1590, 1592
`__ctex_ltj_char_range_parse_feature:w` . 1595, 1599
`\g_ctex_ltj_char_range_prop` 1447, 1554
`\l_ctex_ltj_char_range_tl` 1447, 1450, 1556, 1558, 1560, 1562, 1569
`__ctex_ltj_check_family:n` 1027, 1354
`\l_ctex_ltj_current_font_tl` 50, 784, 788, 789, 801, 843, 848
`\l_ctex_ltj_current_shape_tl` 866, 871, 876, 877
`__ctex_ltj_declare_char_range:nn` 1550, 1552
`\g_ctex_ltj_default_features_clist` 1021, 1240, 1241
`__ctex_ltj_family_csname:n` ... 1031, 1048, 1065, 1092
`\g_ctex_ltj_family_font_name_prop` 1006, 1024, 1029, 1103, 1179, 1272, 1284
`\g_ctex_ltj_family_font_options_prop` 1007, 1024, 1182
`\g_ctex_ltj_family_int` . 1145, 1147, 1432, 1433, 1435
`\g_ctex_ltj_family_name_prop` 1024, 1033, 1055, 1089, 1399
`__ctex_ltj_family_unknown_warning:n` 1083, 1101, 1528
`\l_ctex_ltj_font_options_clist` 1003, 1008, 1009, 1010, 1018, 1054, 1183, 1184, 1187, 1390, 1391, 1395
`__ctex_ltj_fontspec:nnn` 1146, 1153, 1173
`\g_ctex_ltj_fontspec_family_tl` 1053, 1056, 1057, 1058, 1064, 1069, 1094
`\g_ctex_ltj_fontspec_prop` 1141, 1169, 1174
`__ctex_ltj_gset_family_cs:nn` 1011, 1046
`__ctex_ltj_if_jfont_math:w` 909, 912
`\l_ctex_ltj_jfm_tl` 956, 977, 979, 1022
`\c_ctex_ltj_math_fam_int` 985, 986
`\c_ctex_ltj_math_family_tl` 1306, 1308, 1310, 1313, 1317
`\c_ctex_ltj_math_tl` 980, 981, 983, 985, 1238, 1297, 1307, 1312, 1316
`__ctex_ltj_msg_def_family_map:n` 1116, 1118
`__ctex_ltj_msg_family_map:n` 1045, 1115, 1129
`__ctex_ltj_noembed_wrap:n` 993, 995
`__ctex_ltj_patch_external_font:w` 829, 862
`__ctex_ltj_pop_fontname:` 800, 809, 888
`__ctex_ltj_push_fontname:n` 796, 804, 885
`\g_ctex_ltj_reset_alternate_prop` 1038, 1522, 1534, 1542
`__ctex_ltj_save_alternate_shape:Nnn` 1454, 1479, 1495
`__ctex_ltj_save_fontname:w` 806, 810
`__ctex_ltj_select_font_aux:` 785, 791
`__ctex_ltj_set_alternate_family:n` 1059, 1066

- _ctex_ltj_tmp:w 1329, 1346, 1424, 1426, 1427
 - \l_ctex_ltj_tmp_seq 1337, 1338, 1341, 1348
 - \l_ctex_ltj_tmp_tl
 - 1002, 1003, 1029, 1040, 1043, 1180, 1188,
 - 1204, 1207, 1208, 1274, 1279, 1297, 1298, 1300, 1301,
 - 1335, 1336, 1337, 1374, 1376, 1377, 1384, 1594, 1602, 1611
 - \g_ctex_ltj_unknown_family_seq ... 1105, 1107, 1112
 - _ctex_ltj_update_family_uid:N ... 1009, 1390, 1429
 - _ctex_ltj_use_global_options:N
 - 1010, 1019, 1372, 1391
 - \c_ctex_macosx_file_str 1795, 1799
 - \l_ctex_msyh_suffix_tl
 - 4052, 4053, 4057, 4073, 4075, 4109
 - \l_ctex_parameter_str 5104, 5123, 5144, 5191
 - \l_ctex_parameter_tl 5144, 5149, 5160
 - _ctex_parse_name:nnnnN 5056, 5065
 - _ctex_parse_name:w 5078, 5084, 5093
 - _ctex_patch_cmd:Nnnnw 4996, 5001, 5006, 5162
 - _ctex_patch_false:w 5032, 5038, 5177, 5179, 5196, 5222
 - _ctex_patch_rebuild:Nn 5135, 5141, 5174, 5215, 5219
 - _ctex_patch_replace:nnTF 5169, 5181
 - _ctex_patch_rescan:NNn 5143, 5144, 5145, 5153
 - _ctex_patch_true:w 5031, 5037, 5175, 5208, 5220
 - \l_ctex_prefix_str 5104, 5122, 5143, 5200, 5201
 - \l_ctex_prefix_tl 5143, 5148, 5159
 - \l_ctex_punct_bounds_clist ... 4779, 4785, 4791, 4797
 - \c_ctex_punct_family_clist 625, 635
 - \l_ctex_punct_font 4777, 4778
 - \c_ctex_punct_seq 4756, 4757, 4780
 - \l_ctex_punct_tl
 - 280, 594, 595, 681, 964, 2035, 2036, 2037, 2038
 - _ctex_rebuild_cmd:w ... 5135, 5136, 5140, 5215, 5216
 - \l_ctex_replacement_str
 - 5104, 5124, 5145, 5183, 5184, 5218
 - \l_ctex_replacement_tl 5145, 5150, 5161
 - _ctex_right_hook_aux:nn 5212
 - _ctex_save_bounds:n 4781, 4789
 - _ctex_save_cmap:Nn 476, 480
 - _ctex_save_font_size:nn 3667
 - _ctex_save_selectfont: 1965, 1970
 - \l_ctex_scheme_tl 315, 3294, 3570
 - \g_ctex_section_depth_flag
 - 94, 309, 311, 3492, 3508, 3524
 - \c_ctex_section_headings_seq . 2283, 2290, 2897, 3001
 - _ctex_set_font_size:nnNn 3739, 3742
 - \g_ctex_spa_iow 4767, 4770, 4772, 4782
 - \c_ctex_std_class_tl ... 3541, 3556, 3557, 3623, 3646
 - \g_ctex_std_options_clist
 - 137, 382, 394, 404, 406, 410, 415, 420, 425
 - _ctex_titlesec_format:n 2997, 3005, 3008
 - _ctex_titlesec_format:Nn 2987, 3011, 3013
 - _ctex_titlesec_format_explicit:Nn ... 2988, 3023
 - _ctex_titlesec_hook:n 2997, 3001, 3003
 - _ctex_titlesec_spacing:Nn 3006, 3033
 - _ctex_titlesec_spacing:nnnnn 3034, 3035
 - _ctex_tmp:w 5042, 5097, 5108, 5129
 - \l_ctex_tmp_box 58, 1909, 1910, 3126, 3130
 - \l_ctex_tmp_dim 58,
 - 2097, 2098, 2103, 2106, 2112, 2151, 2153, 2155, 2164, 2166
 - \l_ctex_tmp_int 58, 575,
 - 577, 578, 580, 584, 586, 2111, 2113, 2114, 2120, 2121,
 - 2153, 2154, 2155, 2158, 2159, 2164, 2166, 2896, 2899, 2903
 - \l_ctex_tmp_tl ... 58, 366, 373, 388, 389, 482, 483,
 - 574, 576, 2330, 2394, 2407, 2417, 3622, 3623, 3711, 3712
 - \c_ctex_upper_case_file_str 1783, 1791
 - \c_ctex_version_tl 3, 6, 7, 8, 9, 20, 23
 - _ctex_write_family:nn 4769, 4774
 - \l_ctex_xkanjiskip_skip 1988
 - \l_ctex_xkanjiskip_tl 1995, 2000, 2001
 - \c_ctex_zero_tl 2071, 2078, 2080
 - \g_ctex_zhmCJK_bool 131, 260, 265, 270, 504, 598
 - \l_ctex_ziju_dim 2147, 2148, 2152, 2163, 2165, 2169, 2173
 - \l_ctex_ziju_tl 2071, 2079, 2080, 2141, 2147
 - \ctexbibname 2256, 2276
 - \CTEXcounter 2191
 - \CTEXdigits 25, 2201
 - \CTEXfilltwosides 30, 1879, 1889
 - \CTEXifname 16, 2440, 2696, 2878, 2885, 2892,
 - 3118, 3159, 3162, 3166, 3176, 3180, 3188, 3191, 3196, 3199
 - \CTEXindent 30, 2174
 - \CTEXnoindent 30, 2174
 - \CTEXnumber 25, 2201
 - \CTEXnumberline 2890, 2910, 3369
 - \CTEXoptions 30, 432
 - \ctexset 5, 429, 436, 442
 - \CTEXsetfont 30, 2066
 - \CTEXsetup 30, 432
 - \CTEXsout 30
 - \ctexspadef 671, 4784
 - \CTEXthechapter 16, 2643, 2886, 3053, 3176, 3196
 - \CTEXtheparagraph 16
 - \CTEXthepart 16, 2879, 3058
 - \CTEXthesection 16, 3159, 3166, 3180, 3188, 3199
 - \CTEXthesubparagraph 16
 - \CTEXthesubsection 16, 3162, 3191
 - \CTEXthesubsubsection 16
 - \CTEXunderdblline 30
 - \CTEXunderdot 30
 - \CTEXunderline 30
 - \CTEXunderwave 30
 - \CTEXxout 30
 - \CurrentOption 394
- D**
- declarecharrange 26, 1544
 - \DeclareFontEncoding 966
 - \DeclareFontFamily
 - 490, 975, 3941, 3942, 3943, 3944, 3945, 3946
 - \DeclareFontShape
 - 976, 978, 1657, 1658, 1661, 1662, 3957, 3958,
 - 3959, 3960, 3961, 3962, 3963, 3964, 3965, 3968, 3969,
 - 3970, 3971, 3972, 3973, 3974, 3975, 3976, 3980, 3981,

3982, 3983, 3984, 3985, 3989, 3990, 3991, 3992, 3993,
3994, 3995, 3996, 3997, 4000, 4001, 4002, 4003, 4004,
4005, 4006, 4007, 4008, 4012, 4013, 4014, 4015, 4019,
4020, 4021, 4022, 4023, 4024, 4025, 4026, 4027, 4030,
4031, 4032, 4033, 4034, 4035, 4036, 4037, 4038, 4042, 4043
`\DeclareFontSubstitution` 969
`\DeclareKanjiFamily`
..... 1655, 1656, 3948, 3949, 3950, 3951, 3952, 3953
`\DeclareMathSizes` 3704
`\DeclareSymbolFont` 981, 1307
`\def` 2504, 2541, 2573, 2591,
2603, 2634, 2674, 2683, 2707, 2726, 2752, 2756, 2791,
2806, 2820, 2821, 2825, 3764, 3775, 3803, 3814, 4448, 4484
`\defaultCJKfontfeatures` 1221
`\defbeamertemplate` 3205, 3223, 3241
`\defbeamertemplatealias` 3259, 3260, 3261
dim commands:
`\dim_abs:n` 2165
`\dim_add:Nn` 2148
`\dim_compare:nNnTF` ... 112, 2098, 2133, 2149, 2156, 2424
`\dim_eval:n` 2118, 2164
`\dim_max:nn` 3129
`\dim_min:nn` 2165
`\dim_new:N` 61, 1919, 2173
`\dim_set:Nn` 1910, 1956, 1957,
1958, 2097, 2103, 2134, 2147, 2151, 2178, 2431, 3127, 3834
`\dim_sub:Nn` 2155
`\dim_to_decimal:n` 36, 3675, 3676, 3713
`\dim_to_decimal_in_unit:nn` 4804
`\dim_use:N` 120, 3834
`\dim_zero:N` 2184
`\c_max_dim` 2098, 2424
`\c_zero_dim` 2106, 2118, 2133, 2149, 2156, 2376, 3525
`\dimexpr` 2523, 2578, 2689, 2713, 3211, 3229, 3247
`\directlua` 719, 720
`\document` 4960, 4963

E

`\edef` 4467, 4468
`\else` . 1644, 1645, 1649, 1650, 2476, 2485, 2494, 2510, 2515,
2547, 2552, 2622, 2630, 2646, 2652, 2658, 2670, 2679,
2737, 2742, 2761, 2766, 2785, 2790, 2799, 2819, 2835,
2848, 3333, 3360, 4479, 4480, 4526, 4565, 4596, 4655, 4718
else commands:
`\else:` 507, 860, 906, 1941, 2964, 3163, 3838
`\em` 1642
`\eminnershape` 1644, 1645, 1650, 1652
`\end` 3219, 3237, 3255, 4849
`\endCJKfilltwosides` 1885
`\endcsname` 2744,
2753, 2754, 2755, 2762, 2780, 2781, 2783, 2786, 2795,
2796, 2797, 2800, 4474, 4476, 4477, 4478, 4481, 4483, 4487
`\endCTEXfilltwosides` 1885, 1890
`\endgroup` 2782, 2818, 2843, 3221, 3239, 3257, 4467, 4475, 4490
`\endinput` 4472
`\endlinechar` 4446, 4447

etex commands:

`\etex_dimexpr:D` 120, 2112
`\etex_glueexpr:D` 2104
`\everypar` 2741, 2838, 2850
`\EverySelectfont` 1700, 1974, 1977
exp commands:
`\exp_after:wN`
119, 813, 829, 831, 906, 909, 3034, 3657, 3704, 3739, 5113
`\exp_args:cc` 3269
`\exp_args:Nc` 3006, 3011, 5067
`\exp_args:Nco` 3116
`\exp_args:Nnc` 99
`\exp_args:NNf` 1602
`\exp_args:NNNo` 3833
`\exp_args:Nno` 3281
`\exp_args:No` 576, 1792, 3030, 4781, 5129
`\exp_args:Nx` 293, 604, 678, 2138, 3653, 3832
`\exp_last_unbraced:Nn` 5078
`\exp_not:N` 124, 125, 486, 606,
607, 608, 775, 776, 778, 848, 884, 969, 1055, 1056, 1057,
1058, 1381, 1502, 1722, 1723, 1724, 1738, 1874, 1875,
1876, 1903, 2304, 2305, 2306, 2311, 2313, 2314, 2315,
2316, 2319, 2320, 2323, 2335, 2336, 2337, 2338, 2339,
2340, 2341, 2342, 2343, 2344, 2345, 2347, 2348, 2349,
2351, 2352, 2354, 2357, 2362, 2364, 2366, 2368, 2370,
2372, 2902, 2904, 2905, 2906, 2907, 3018, 3019, 3028,
3029, 3066, 3068, 3069, 3673, 3875, 5058, 5059, 5060, 5148
`\exp_not:n` 69,
388, 777, 779, 1052, 1053, 1054, 1187, 1188, 1382,
1383, 1384, 1408, 1502, 1717, 1739, 1962, 2358, 2417,
3037, 3270, 4939, 4963, 5076, 5148, 5149, 5150, 5205, 5206
`\exp_stop_f:` 743
`\expandafter`
4449, 4450, 4474, 4475, 4476, 4477, 4478, 4483, 4487, 4488
`\ExplSyntaxOff` 66, 72, 495, 538, 542, 546, 550, 701,
1253, 1257, 1261, 1265, 1643, 1966, 3138, 3204, 4810, 5009
`\ExplSyntaxOn` 71, 730, 3262, 4749

F

`\familydefault` 1720
`fancyhdr` 29, 341
`\fangsong` 4432
`\fi` .. 1704, 2467, 2472, 2478, 2487, 2496, 2499, 2514, 2519,
2527, 2536, 2551, 2556, 2566, 2586, 2606, 2614, 2615,
2618, 2622, 2632, 2650, 2656, 2662, 2673, 2682, 2686,
2694, 2695, 2704, 2710, 2722, 2727, 2733, 2739, 2746,
2748, 2770, 2771, 2787, 2801, 2804, 2823, 2832, 2851,
2852, 3158, 3161, 3165, 3174, 3179, 3187, 3190, 3195,
3198, 3215, 3233, 3251, 3333, 3360, 4479, 4480, 4489, 4742
fi commands:
`\fi:` 407,
554, 591, 610, 860, 906, 1941, 2967, 3167, 3181, 3748, 3838
`\figurename` 2237
`figurename` 11
file commands:
`\g_file_current_name_tl` 693, 1792

`\file_if_exist:nTF` 48, 1779, 1781, 1783, 1795, 3868, 4047, 4054, 4056
`\file_if_exist_input:nTF` 86
`\file_input:n` 78, 87, 89, 527, 563
`fntef` 30, 341
`\font` 742, 3941, 3942, 3943, 3944, 3945, 3946
font commands:
`\font_glyph_if_exist:NnTF` 742
`fontset` 7, 225, 3888
`\fontsize` 3657
fontspec commands:
`\g_fontspec_encoding_tl` 974
`\l_fontspec_family_tl` 1441
`\fontspec_set_family:Nnn` 1053, 1418
`\fontspec_visible_space:` 740
`\fontspec_visible_space_fallback:` 744
fontspec internal commands:
`__fontspec_namewrap:n` 55, 993
`\footnotesep` 3833
`\footnotesize` 3770, 3809, 3833
fp commands:
`\fp_compare:nNnTF` 3617, 3644, 3830
`\fp_eval:n` 2141, 4800
`\fp_set:Nn` 3618, 3645
`\fp_use:N` 2078, 3832
`\c_nan_fp` 182
`\c_zero_fp` 2078, 3617, 3644, 3830

G

`GBK` 6, 221
`\gdef` 1695, 2928, 2929, 2930, 2931, 2934, 2935, 2936, 2937
`\global` 2626, 2837, 2840
`\globaljfont` 831
`\glueexpr` 2777, 2792, 2813, 2820, 2821, 2847, 3210, 3220, 3228, 3238, 3246, 3256
group commands:
`\group_begin:` 490, 735, 748, 793, 815, 841, 883, 910, 998, 1050, 1327, 1412, 1497, 1517, 1533, 1716, 2310, 2318, 2392, 2953, 3075, 3133, 3833, 4776, 4957, 5033, 5041, 5052, 5107
`\group_end:` 494, 503, 739, 752, 798, 819, 854, 887, 914, 1013, 1060, 1347, 1419, 1503, 1529, 1540, 1728, 2321, 2325, 2416, 2968, 3098, 3137, 3833, 4787, 4957, 5055, 5102, 5130, 5164, 5188
`\group_insert_after:N` 851, 852
`\gtfamily` 1644, 1649

H

hbox commands:
`\hbox_set:Nn` 1909, 3126
`heading` 9, 304
`\heiti` 4431
`\hskip` ... 2777, 2792, 2813, 2820, 2821, 2846, 2847, 3187, 3190
`\hspace` 2509, 2546, 2879, 2886, 3058
`\Huge` 2567, 2597, 2698, 2716, 3337, 3367, 3787, 3826
`\huge` 2528, 2564, 2580, 2693, 3319, 3336, 3340, 3365, 3373, 3786, 3825

`hyperref` 30, 341
`\hypersetup` 358, 359, 1805
`\hyphenchar` 3941, 3942, 3943, 3944, 3945, 3946

I

if commands:

`\if_bool:N` 504, 598
`\if_case:w` 403, 3744
`\if_cs_exist:N` 571
`\if_false:` 107
`\if_meaning:w` 1940
`\if_true:` 106
`\ifctexpdf` 105
`\ifdim` 2731, 2773, 2809, 2827
`\ifincsname` 3838
`\IfNoValueTF` 437, 443, 1205, 1466, 1571, 1604, 1606, 2388
`\ifnum` 1702, 2505, 2524, 2542, 2563, 2635, 2691, 2757, 2785, 2799, 3158, 3161, 3165, 3172, 3179, 3187, 3190, 3195, 3198, 4479
`\ifodd` 2472, 2474, 2499, 2506, 2524, 2536, 2543, 2563, 2586, 2606, 2628, 2639, 2686, 2704, 2710, 2722, 2735, 2746, 2762, 2774, 2810, 2828, 2832, 3213, 3231, 3249
`\ifx` 4477, 4478, 4483
`\ifzhmappdf` 4495, 4562, 4593, 4624, 4691
`\ignorespaces` 2853
`\immediate` 4486
`indent` 29, 201
`\indexname` 2240
`indexname` 12
`\input` 4748, 4813
`\insertcontinuationtext` 2249, 2258, 2265
`\insertpart` 3217, 3218
`\insertromanpartnumber` 3208, 3306
`\insertsection` 3235, 3236
`\insertsectionnumber` 3226, 3404
`\insertsubsection` 3253, 3254
`\insertsubsectionnumber` 3244, 3433

int commands:

`\c_five` 3676
`\c_four` 311, 584, 586
`\int_compare:nNnTF` 577, 2114, 2158, 3492, 3508, 3524, 3615, 3642
`\int_const:Nn` 985
`\int_div_truncate:nn` 77, 584
`\int_eval:n` 1467, 1469, 1470, 1572, 1574, 1575
`\int_from_hex:n` 576
`\int_gincr:N` 1145, 1432
`\int_incr:N` 2899
`\int_mod:nn` 586
`\int_new:N` 59, 1435
`\int_set:Nn` 567, 575, 2111, 2153
`\int_set_eq:NN` 561
`\int_sub:Nn` 580, 2159
`\int_to_Hex:n` 578, 583, 585
`\int_use:N` 487, 567, 898, 1147, 1433, 2903, 4781
`\int_zero:N` 2896

\c_one	162, 1460, 1565, 2166, 2383, 4793	\listfigurename	2235
\c_six	3676	listfigurename	11
\c_three	309, 2158, 3508, 4794	\listtablename	2236
\c_two	95, 163, 377, 381, 2159, 3130, 3492, 3524	listtablename	11
\c_zero	161, 2114, 3616, 3643	\LoadClass	411, 416, 421, 426
\interlinepenalty	2521, 2560, 2576, 2594, 2697, 2715, 2778, 2779, 2814, 2815	\long	4484
iow commands:		\lstlistingname	776, 777
\iow_close:N	4770	\lstlistlistingname	778, 779
\iow_indent:n	4944	\ltjalchar	743
\iow_new:N	4772	\ltjdefcharrange	702, 703, 704, 706, 710, 711, 714, 717
\iow_now:Nn	4782	LTJFONTUID	1429
\iow_open:Nn	4767	\ltjgetparameter	1914, 1944, 1992, 2006
\itemsep	3767, 3778, 3806, 3817	\ltjsetkanjiskip	1931
\itshape	1644, 1645, 1649, 1650	\ltjsetmathletter	750
		\ltjsetparameter	718, 724, 1622
		\ltjsetxkanjiskip	1996
		lua commands:	
J		\lua_escape_x:n	859, 871, 898, 905, 1457, 1472, 1473, 1490, 1491, 1500
\jfam	950, 986	\lua_now_x:n	736, 833, 859, 863, 868, 873, 895, 904, 972, 1478, 1502, 1748
		M	
K		\MAKESPA	4773, 4815
\kaishu	4433	\markboth	2439, 2530, 2558
\kanjifamily	1632, 1633, 1634, 1637	\mathgroup	950
\kanjifamilydefault	1640	\mcfamily	1649
kernel internal commands:		\mddefault	970, 976, 982, 1308, 1313
\l_kernel_expl_bool	70	mode commands:	
keys commands:		\mode_if_math:TF	944
\l_keys_choice_tl	294	msg commands:	
\keys_define:nn	155, 388, 391, 990, 1320, 1429, 1506, 1544, 1583, 2010, 2031, 2049, 2083, 2206, 2232, 2260, 2270, 2273, 2276, 2417, 2912, 3841, 3888	\msg_critical:nnn	53, 693, 3857
\keys_if_exist:nnTF	1581	\msg_error:nn	3908
\l_keys_key_tl	141, 143, 145	\msg_error:nnn 19, 22, 38, 41, 684, 697, 700, 2222, 3658, 3740	
\keys_set:nn	28, 171, 177, 233, 242, 251, 277, 287, 294, 301, 331, 337, 430, 437, 443, 3297, 3352, 3380, 3412, 3447, 3464, 3478, 3494, 3502, 3510, 3518, 3525, 3526, 3528, 3594, 3596, 3930	\msg_error:nnnn	3874, 3903
\keys_set_known:nn	3573	\msg_fatal:nn	4754
\keys_set_known:nnN	1002	\msg_new:nn	4752
		\msg_new:nnn	140, 142, 144, 146, 148, 687, 1044, 1113, 1195, 1861, 2253, 3547, 3858, 3917, 4940, 5017
		\msg_new:nnnn	12, 30, 45, 99, 2224, 3660, 3882, 3911
L		\msg_warning:nn	1190, 1865, 2028, 2039, 2275
\labelformat	3281	\msg_warning:nnn	169, 175, 203, 213, 228, 237, 246, 275, 285, 299, 329, 335, 343, 348, 354, 1108, 3566, 3845, 3895, 4946, 5016
\languagealias	3602, 3607, 3628, 3633	\msg_warning:nnnn	435, 441, 1040, 1874, 1881, 2176, 2182
\LARGE	3785, 3824		
\Large	2525, 3317, 3323, 3391, 3395, 3784, 3823		
\large	3416, 3783, 3822		
\lastbox	2841		
\leavevmode	2467, 2727		
left commands:		N	
\c_left_brace_str	5099	\newCJKfontfamily	1200
\leftmargin	3764, 3775, 3803, 3814	\NewDocumentCommand ... 430, 433, 439, 1074, 1200, 1202, 1210, 1215, 1221, 1227, 1232, 1237, 1239, 1367, 1459, 1564, 1599, 1635, 2066, 2137, 2174, 2180, 2201, 2203, 2382, 3652, 3837, 4430, 4431, 4432, 4433, 4435, 4436, 4438	
\leftmargini	3764, 3775, 3803, 3814	\newfontfeature	987, 988, 989
\let	1708, 2760, 2769, 3757, 3796, 4470, 4476	\newpage	2608, 2613
\linespread	3832	\ngostype	44, 1766
linespread	10, 181	\nobreak	2525, 2533, 2583, 2693, 2698, 2701, 2716, 2719, 2829, 3318, 3366
linestretch	10, 2083		
\linewidth	2112, 2120, 2152		
\lishu	4435		

`\nobreakspace` 2525, 2564
`nocap` 29
`\noexpand` 4468, 4470, 4471, 4472
`nofonts` 29
`\noindent` 2859
`noindent` 29
`nopunct` 29
`\normalem` 1844
`\normalfont` 550, 552, 553, 1224,
1230, 1235, 1265, 1267, 1268, 1291, 2522, 2523, 2561,
2562, 2577, 2578, 2595, 2596, 2688, 2689, 2712, 2713, 2907
`\normalsize` 3450, 3467, 3481, 3751, 3790, 3828, 3836
`nospace` 29
`nozhmap` 29
`\null` 2497, 2611
`\numberline` 2645, 2786, 2800, 2886, 2893, 3069, 3124

O

one commands:

`\c_minus_one` 152, 561, 3615, 3642
`\onecolumn` 2492

or commands:

`\or:` 405, 3746

P

`\pagestyle` 3593
`\par` 2430, 2468, 2525, 2564, 2567, 2580,
2597, 2693, 2698, 2716, 2728, 2829, 3209, 3210, 3217,
3220, 3227, 3228, 3235, 3238, 3245, 3246, 3253, 3256,
3299, 3318, 3329, 3347, 3355, 3366, 3407, 3409, 3441, 3443
`paragraph/afterskip` 31
`paragraph/beforeskip` 31
`paragraph/hang` 21
`paragraph/numbering` 15
`paragraph/runin` 19
`\parindent`
2133, 2134, 2156, 2177, 2178, 2183, 2184, 2522, 2523,
2577, 2578, 2688, 2689, 2712, 2713, 3211, 3229, 3247, 3526
`\parsep` 3766, 3767, 3777, 3778, 3805, 3806, 3816, 3817
`\part` 2465
`part/beforeskip` 31
`part/fixbeforeskip` 31
`part/numbering` 15
`part/pagestyle` 21, 2392
`\partmark` 2438, 2531, 2559
`\partname` 2525, 2564, 3208, 3301
`\PassOptionsToClass` 410, 415, 420, 425
`\PassOptionsToPackage` 505, 1809, 2186, 4948
`\pdfmapline` 4496,
4497, 4498, 4499, 4500, 4501, 4502, 4504, 4505, 4506,
4507, 4508, 4509, 4510, 4512, 4513, 4514, 4515, 4516,
4517, 4519, 4520, 4521, 4522, 4523, 4524, 4625, 4626,
4627, 4628, 4629, 4630, 4631, 4633, 4634, 4635, 4636,
4637, 4638, 4639, 4641, 4642, 4643, 4644, 4645, 4646,
4648, 4649, 4650, 4651, 4652, 4653, 4692, 4693, 4694,
4695, 4696, 4697, 4699, 4700, 4701, 4702, 4703, 4704,
4706, 4707, 4708, 4709, 4710, 4712, 4713, 4714, 4715, 4716

`\pdfoutput` 4479
`\pdfstringdefDisableCommands` 3840

pdftex commands:

`\pdftex_pdffontattr:D` 486
`\pdftex_pdflastobj:D` 487
`\pdftex_pdfobj:D` 483

prg commands:

`\prg_do_nothing:`
..... 954, 1406, 1498, 1890, 1964, 2015, 2871, 5140
`\prg_new_conditional:Npnn` 857, 1937
`\prg_new_protected_conditional:Npnn` 1087, 5110
`\prg_return_false:` ... 860, 1097, 1941, 1945, 1949, 5120
`\prg_return_true:` 860, 1090, 1095, 1941, 1945, 1949, 5125
`\primitive` 4849
`\ProcessKeysOptions` 400
`\proofname` 2227, 2242
`proofname` 12

prop commands:

`\prop_get:NnN` 1182
`\prop_get:NnNTF`
... 1089, 1100, 1141, 1179, 1399, 1447, 3656, 3711, 3738
`\prop_gpop:NnNTF` 1029, 1033, 1534
`\prop_gput:Nnn` . 1006, 1007, 1055, 1169, 1522, 1558, 3673
`\prop_gremove:Nn` 1038
`\prop_if_empty:NTF` 1103, 1272
`\prop_map_break:n` 1286
`\prop_map_inline:Nn` 1284
`\prop_new:N` 1024, 1025, 1026, 1174, 1542, 1563, 3667
`\protect` 2452, 2458, 2645,
2665, 2666, 2786, 2800, 2886, 2893, 3068, 3080, 3082,
3086, 3088, 3092, 3094, 3102, 3107, 3119, 3120, 5071, 5072
`\ProvideDocumentCommand` 2438
`\ProvidesFile` 4484, 4492, 4559, 4590, 4621, 4688
`\providetranslation`
..... 4854, 4855, 4856, 4857, 4858, 4859, 4860,
4861, 4862, 4863, 4864, 4865, 4866, 4867, 4868, 4869,
4870, 4871, 4872, 4873, 4874, 4875, 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

ptex commands:

`\ptex_kanjiskip:D` 1917, 1934, 1948
`\ptex_xkanjiskip:D` 1993, 1997, 2007
`punct` 9, 29, 280, 2031
`\punctstyle` 595, 2036

Q

`\quad` 2753, 3158, 3159, 3161, 3162, 3165, 3166, 3176,
3188, 3191, 3196, 3324, 3374, 3384, 3417, 3451, 3468, 3482
`quark commands:`
`\q_mark` 5114, 5117
`\q_stop` 120, 125, 909,
912, 1364, 1368, 1595, 1600, 5079, 5084, 5093, 5114, 5117

\quotation 3850

R

\raggedright 2522, 2577, 2688, 2712, 3316, 3364
 \refname 2243, 2248, 2255, 2264, 2272, 2273
 refname 12
 \refstepcounter ... 2508, 2545, 2641, 2764, 3264, 3286, 3287
 \relax 2523, 2542, 2563, 2578, 2689, 2713, 2729,
 2765, 2772, 2777, 2792, 2808, 2813, 2820, 2821, 2826,
 2845, 2847, 3187, 3190, 3210, 3211, 3220, 3228, 3229,
 3238, 3246, 3247, 3256, 4442, 4446, 4449, 4477, 4478, 4483
 \renewcommand 2466, 2482, 2621
 \RenewDocumentCommand 3287
 \RequirePackage 27, 56, 57, 350, 361, 506, 509, 510, 511, 677,
 695, 698, 732, 775, 1818, 1827, 1843, 1849, 2187, 3647, 4993
 \RequirePackageWithOptions 4949
 resetalternatefont 27, 1506
 reverse commands:
 \reverse_if:N 598
 \rmdefault 1722
 \rmfamily 538, 541, 1253, 1256, 1632

S

scan commands:

\scan_stop:
 ... 64, 120, 831, 950, 1499, 1500, 2104, 2112, 4777, 4959
 scheme 9, 315
 \scriptsize 3781, 3820
 \secdef 2479, 2501, 2633
 section/afterskip 31
 section/beforeskip 31
 section/hang 21
 section/numbering 15
 section/runin 19
 \sectionname 3226, 3401
 \selectfont
 1081, 1525, 1538, 1638, 1690, 1966, 1972, 1985, 2068, 3657

seq commands:

\seq_clear:N 1000
 \seq_concat:NNN 1340
 \seq_gput_left:Nn 2291, 2292
 \seq_gput_right:Nn 1107, 3679
 \seq_gset_eq:NN 2290
 \seq_gset_from_clist:Nn 2285, 2295, 2943, 4757
 \seq_if_empty:NTF 1355
 \seq_if_in:NnTF 463, 1105
 \seq_map_function:NN 3001
 \seq_map_inline:Nn
 ... 1363, 2409, 2897, 2946, 2956, 2965, 3060, 3275, 4780
 \seq_new:N
 ... 456, 1112, 1348, 1349, 2284, 2288, 2942, 3668, 4756
 \seq_put_right:Nn 1611
 \seq_set_filter:NNn 1338
 \seq_set_from_clist:Nn 457
 \seq_set_split:Nnn 1337
 \seq_use:Nnnn 3665
 \setbox 2841

\setCJKfamilyfont 1126,
 1200, 4074, 4080, 4081, 4082, 4083, 4084, 4085, 4128,
 4129, 4137, 4138, 4140, 4141, 4142, 4143, 4145, 4167,
 4169, 4171, 4173, 4212, 4213, 4214, 4215, 4239, 4244,
 4249, 4251, 4289, 4291, 4293, 4294, 4308, 4309, 4310,
 4311, 4324, 4325, 4326, 4327, 4328, 4329, 4370, 4371,
 4372, 4373, 4374, 4375, 4388, 4389, 4390, 4423, 4424, 4425
 \setCJKmainfont 1122, 1221, 4066, 4124,
 4132, 4159, 4205, 4227, 4284, 4305, 4320, 4365, 4384, 4419
 \setCJKmathfont 1221
 \setCJKmonofont 1124, 1221, 4079, 4127,
 4136, 4166, 4211, 4238, 4288, 4307, 4323, 4369, 4387, 4422
 \setCJKromanfont 1226, 1246
 \setCJKsansfont 1123, 1221, 4069, 4073, 4126,
 4134, 4165, 4210, 4233, 4287, 4306, 4322, 4367, 4386, 4421
 \SetSymbolFont 983, 1312, 1316
 \sfdefault 1723
 \sffamily 542, 545, 1257, 1260, 1633

skip commands:

\skip_horizontal:N 1924, 1928
 \skip_if_eq:nnTF 1944, 1948, 1991, 2449, 2455
 \skip_new:N 1936, 2004, 2434
 \skip_set:Nn 1914,
 1917, 1995, 2005, 2105, 2116, 2161, 2169, 2436, 2448, 2454
 \skip_set_eq:NN 1934, 1997
 \skip_sub:Nn 2425, 2432
 \skip_use:N 2452, 2458
 \skip_zero:N 2125
 \c_zero_skip 2374, 2375, 2400, 2401, 2449, 2455
 \small 3759, 3798
 \songti 4430
 \space 2642, 2693, 3301, 3362, 3401, 3432, 3534
 space 10, 29, 290, 2010
 \special 1670, 1671, 1674,
 1675, 1825, 1839, 4527, 4528, 4529, 4530, 4531, 4532,
 4533, 4535, 4536, 4537, 4538, 4539, 4540, 4541, 4543,
 4544, 4545, 4546, 4547, 4548, 4550, 4551, 4552, 4553,
 4554, 4555, 4566, 4567, 4568, 4569, 4570, 4572, 4573,
 4574, 4575, 4576, 4578, 4579, 4580, 4581, 4583, 4584,
 4585, 4586, 4597, 4598, 4599, 4600, 4601, 4603, 4604,
 4605, 4606, 4607, 4609, 4610, 4611, 4612, 4614, 4615,
 4616, 4617, 4656, 4657, 4658, 4659, 4660, 4661, 4662,
 4664, 4665, 4666, 4667, 4668, 4669, 4670, 4672, 4673,
 4674, 4675, 4676, 4677, 4679, 4680, 4681, 4682, 4683,
 4684, 4719, 4720, 4721, 4722, 4723, 4725, 4726, 4727,
 4728, 4729, 4731, 4732, 4733, 4734, 4736, 4737, 4738, 4739
 \SplitArgument 1460, 1565, 2383
 str commands:
 \c_backslash_str 5098
 \c_colon_str 936, 937, 938
 \str_case:nnTF 1720, 1766, 5069, 5084
 \str_case_x:nnTF 1120, 1131
 \str_const:Nn 43, 50, 1791, 1799
 \str_if_empty:NTF 5155, 5191, 5200
 \str_if_eq:nnTF 508,
 599, 958, 1814, 1822, 3598, 3625, 3639, 3897, 3914, 3921

`\str_if_eq_x:nnTF` 594, 1276, 3894, 5075
`\str_lower_case:n` 482
`\str_new:N` 4993, 5104, 5105, 5106
`\str_set:Nn` 5122, 5123, 5124
`\str_upper_case:n` 1792
`\strutbox` 3834
`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` 3244, 3432
`subsubsection/afterskip` 31
`subsubsection/beforeskip` 31
`subsubsection/hang` 21
`subsubsection/numbering` 15
`subsubsection/runin` 19
sys commands:
`\c_sys_engine_str` 44
`\sys_if_engine_pdftex:TF`
 130, 4061, 4152, 4220, 4299, 4316, 4380
`\sys_if_engine_uptex:TF`
 4105, 4193, 4272, 4302, 4349, 4408
`\sys_if_engine_xetex:TF` 4368, 4750
`\sys_if_output_pdf:TF` 105, 492, 1817, 4154, 4222

T

`\tablename` 2238
`tablename` 12
`\TeX` 3838

T_EX and L^AT_EX₂_ε commands:

`\@italiccorr` 1624
`\@par` 2778, 2814, 3385, 3418, 3452, 3496, 3512
`\@afterheading` 2538, 2588, 2672, 2681, 2834
`\@afterindentfalse` 2477, 2627, 2631, 2732, 2738
`\@afterindenttrue` 2475, 2629, 2730, 2736
`\@Alph` 3530, 3535
`\@chapapp` 2642, 2693, 3173, 3195
`\@chapter` 2633, 2634
`\@chinese` 2189, 2198
`\@clubpenalty` 2849
`\@currentHref` 2977
`\@currentlabel` 3266, 3270
`\@currentxt` 4, 4939, 4988
`\@currname` 4, 4939, 4989
`\@dblarg` 2751
`\@defaultunits` 36
`\@empty` 99, 2760, 2769
`\@endpart` 2570, 2600, 2602

`\@EverySelectfont@AtNextHook` 1694, 1695
`\@EverySelectfont@EveryHook` 1693
`\@EverySelectfont@Init` 1688, 1969
`\@firstoftwo` 3119
`\@font@info` 1703
`\@hangfrom` 2858
`\@ifclassloaded` 3543
`\@ifpackagelater` 37, 40, 683, 696, 699, 1850
`\@ifpackageloaded`
 18, 21, 692, 1698, 1802, 3040, 3142, 3283, 4970
`\@ifpackagewith` 2985, 2994, 3045
`\@ifstar` 2749
`\@listI` 3757, 3796
`\@listi` 3757, 3764, 3775, 3796, 3803, 3814
`\@M` 2521,
 2560, 2576, 2594, 2697, 2715, 2778, 2779, 2814, 2815, 2842
`\@makechapterhead` 2669, 2671, 2683
`\@makeschapterhead` 2678, 2680, 2707
`\@minus` 3386, 3419, 3453, 3469, 3483, 3753,
 3755, 3761, 3763, 3765, 3766, 3772, 3774, 3776, 3777,
 3792, 3794, 3800, 3802, 3804, 3805, 3811, 3813, 3815, 3816
`\@namedef` 3080, 3082, 3086, 3088, 3092, 3094, 3102, 3107
`\@nameuse` 3080, 3082, 3092, 3094, 3102
`\@ne` 3038, 3161, 3190
`\@nil` 884
`\@nobreakfalse` 2836
`\@noskipsecfalse` 2840
`\@noskipsectrue` 2837
`\@onlypreamble` 36, 453, 519,
 1242, 1243, 1244, 1245, 1246, 1247, 1665, 1679, 1687, 3881
`\@part` 2479, 2501, 2503
`\@pkgextension` 6, 7, 8, 9, 20, 23, 694, 4988
`\@plus` 3330,
 3331, 3386, 3387, 3419, 3420, 3453, 3454, 3469, 3483,
 3497, 3513, 3753, 3754, 3755, 3761, 3762, 3763, 3765,
 3766, 3772, 3773, 3774, 3776, 3777, 3792, 3793, 3794,
 3800, 3801, 3802, 3804, 3805, 3811, 3812, 3813, 3815, 3816
`\@popfilename` 130, 4986
`\@schapter` 2633, 2674
`\@seccntformat` 2752, 2765
`\@secondoftwo` 3120
`\@secpenalty` 2743, 3389, 3422, 3456, 3470, 3484
`\@sect` 2751, 2756
`\@setfontsize` 3743
`\@spart` 2479, 2501, 2572
`\@ssect` 2750, 2806
`\@startsection` 91, 93, 99, 2726, 2902
`\@svsec` 2760, 2765, 2769, 2777, 2793, 2794
`\@svsechd` 2791, 2820, 2821, 2843
`\@tempdima` 3127, 3129
`\@tempskipa` 2729, 2731, 2732,
 2743, 2772, 2773, 2808, 2809, 2826, 2827, 2830, 2845, 2846
`\@tempswafalse` 2495
`\@tempswatrue` 2493
`\@topnewpage` 2669, 2678
`\@topnum` 2626

\@xsect	2805, 2824, 2825	\contentsname	11, 11
\@zhdig	2197	\CS	51, 51
\@zhnum	2189, 2196	\csname	109
\abstractname	12	\CTEX@addloflotskip	2445, 2667
\addCJKfontfeature	64	\CTEX@addtocline	2462, 2520, 2557, 2663, 2789, 2803
\AfterEndPreamble	129	\CTEX@afterindent	2735, 2865, 2872
\algorithmname	12	\CTEX@aftertitle	2817, 2822, 2864, 2871
\appendixname	12, 23, 23, 23	\CTEX@appendix@number	2917, 2929, 2935
\AtBeginDocument	36, 45	\CTEX@appendix@numbering	2918, 2931, 2937
\AtBeginDvi	43	\CTEX@chapter@afterindent	2628
\AtBeginShipoutFirst	43	\CTEX@chapter@aftername	2696
\AtEndOfClass	37	\CTEX@chapter@afterskip	2703, 2721
\AtEndOfPackage	37	\CTEX@chapter@aftertitle	2700, 2718
\AtEndOfPackageFile*	130	\CTEX@chapter@beforeskip	2685, 2709
\AtEndPreamble	129	\CTEX@chapter@break	2624
\baselineskip	10, 11, 75, 77, 77	\CTEX@chapter@fixskip	2686, 2704, 2710, 2722
\bfseries	47	\CTEX@chapter@format	2690, 2714
\bibname	12	\CTEX@chapter@indent	2689, 2713
\c@chapter	3535	\CTEX@chapter@lofskip	2398
\c@secnumdepth	2505, 2524, 2542, 2563, 2635, 2691, 2757, 2785, 2799, 3158, 3161, 3165, 3172, 3179, 3187, 3190, 3195, 3198	\CTEX@chapter@lotskip	2399
\c@section	3530	\CTEX@chapter@numbering	2639, 2937
\caption	30	\CTEX@chapter@pagestyle	2397, 2625
\catcode	35, 35, 44, 45, 45, 121, 130	\CTEX@chapter@titleformat	2699, 2717
\ccwd	10, 10, 11, 24, 25, 30, 30, 36, 73, 74, 74, 77, 77, 77, 78	\CTEX@chapter@tocline	2876
\chapter	9, 15, 16, 21, 23, 23, 83, 103	\CTEX@chaptername	2696
\chaptermark	13	\CTEX@document@left@hook	4956, 4962
\chaptername	15	\CTEX@document@right@hook	4958, 4964
\char	49	\CTEX@Family@CMap	468, 496, 499
\CheckCommand	74	\CTEX@fixheadingskip	2423, 2428, 2472, 2536, 2586, 2606, 2704, 2722, 2746, 2832
\Chinese	30, 30	\CTEX@fixskip	2746, 2832, 2866, 2873
\chinese	25, 25, 30, 30, 79	\CTEX@fixtopskip	2420, 2499, 2686, 2710
\CJK@ignorespaces	46, 612	\CTEX@hang	2857, 2867, 2874
\CJK@encoding	55, 56, 63, 790, 794, 878, 965, 1307, 1310, 1312, 1316, 1457, 1472, 1473, 1490, 1491	\CTEX@hangfrom	2777, 2813, 2855
\CJK@envStart	46, 46, 46, 46, 606	\CTEX@headingskip	2437, 2473, 2500, 2537, 2587, 2607, 2687, 2705, 2711, 2723, 2747, 2833
\CJK@family	50, 782, 795, 842, 878, 1078, 1524, 1534, 1536, 1985, 1986	\CTEX@hyperheadinghook	2951, 2981
\CJK@ignorespaces	47, 47, 617, 621	\CTEX@ifnamefalse	2440, 2511, 2516, 2548, 2553, 2574, 2592, 2647, 2653, 2659, 2675, 2758, 2767, 2807
\CJK@input	46, 555	\CTEX@ifnametrue	2440, 2507, 2544, 2640, 2763
\CJK@loadBinding	45, 46, 602	\CTEX@makeanchor	2512, 2517, 2549, 2554, 2648, 2768, 2940, 2949, 2961, 2966, 2972
\CJK@loadEncoding	45	\CTEX@makeanchor@chapter	2654, 2660
\CJK@makeActive	46, 601	\CTEX@makeanchor@schapter	2676
\CJK@plane	45, 75, 471, 472, 482, 1974	\CTEX@makeanchor@sect	2759
\CJK@surr	45, 45, 571	\CTEX@makeanchor@spart	2575, 2593
\CJK@upperReset	46	\CTEX@makeanchor@ssect	2750
\CJKfamily	56	\CTEX@part@afterindent	2474
\CJKfamilydefault	46, 46, 60, 60, 67, 69, 69, 69, 69, 69	\CTEX@part@aftername	2526, 2565, 3214
\CJKglue	77, 77, 109	\CTEX@part@afterskip	2535, 2585, 2605, 3220
\CJKhook	47, 47	\CTEX@part@aftertitle	2532, 2569, 2582, 2599, 3218
\CJKrmdefault	44	\CTEX@part@beforeskip	2471, 2498, 3210
\CJKsymbol	75	\CTEX@part@break	2469, 2488
\CJKunderdotbasesep	30	\CTEX@part@fixskip	2472, 2499, 2536, 2586, 2606
\CJKunderline	30	\CTEX@part@format	2523, 2562, 2579, 2596, 3212
\clearpage	46	\CTEX@part@indent	2523, 2578, 3211

\CTEX@part@numbering	2506, 2524, 2543, 2563, 3213	\CTEXunderline	30
\CTEX@part@pagestyle	2396, 2490	\curr@fontshape	796, 826, 835, 839, 885
\CTEX@part@titleformat	2529, 2568, 2581, 2598, 3218	\DeclareAlternateKanjiFont	63
\CTEX@part@tocline	2876	\DeclareFontEncoding	53
\CTEX@partname	2526, 2565, 3214	\DeclareFontFamily	43, 51
\CTEX@postappendix	2922, 2930, 2936	\DeclareFontShape	51
\CTEX@postchapter	2936	\DeclareFontShape@	63, 1416
\CTEX@postsection	2930	\DeclareRobustCommand	131
\CTEX@preappendix	2921, 2928, 2934	\define@newfont	51, 818
\CTEX@prechapter	2934	\do@subst@correction	51, 52, 817
\CTEX@presection	2928	\em	67, 67, 67
\CTEX@runin	2774, 2810, 2828, 2868, 2875	\enc@update	1691, 1696
\CTEX@save@appendix	2923, 2926	\escapechar	54
\CTEX@save@refstepcounter	3286, 3289	\EverySelectfont	74
\CTEX@section@aftername	3232	\external@font	828, 829, 831
\CTEX@section@afterskip	3238	\extract@font	51, 816
\CTEX@section@aftertitle	3236	\f@baselineskip	2104
\CTEX@section@beforeskip	3228	\f@encoding	794, 834, 947
\CTEX@section@format	3230	\f@family	50, 795, 834, 842
\CTEX@section@indent	3229	\f@series	790, 878
\CTEX@section@numbering	2931, 3231	\f@shape	790, 878
\CTEX@section@titleformat	3236	\f@size	790, 796, 873, 881, 884, 885, 1910, 1956, 3130
\CTEX@sectionname	3232	\fangsong	8, 8
\CTEX@selectfont@hook	1959, 1974, 1977	\figurename	11
\CTEX@setcurrentlabel@n	3264, 3290	\font	51, 51
\CTEX@setheadingskip	2435, 2471, 2498, 2535, 2585, 2605, 2685, 2703, 2709, 2721, 2745, 2831	\font@name	50, 51, 799, 806, 807, 810, 813, 831, 832, 849, 927, 928, 929, 930, 931, 938, 1703
\CTEX@subsection@aftername	3250	\fontcharwd	74, 74
\CTEX@subsection@afterskip	3256	\footnotesep	109, 109, 109
\CTEX@subsection@aftertitle	3254	\footnotesize	109
\CTEX@subsection@beforeskip	3246	\get@external@font	825
\CTEX@subsection@format	3248	\getanddefine@fonts	916
\CTEX@subsection@indent	3247	\globaljfont	51
\CTEX@subsection@numbering	3249	\H@old@chapter	2955
\CTEX@subsection@titleformat	3254	\hbox	84
\CTEX@subsectionname	3250	\heiti	8
\CTEX@thechapter	2935	\hrule	84
\CTEX@thesection	2929	\Hy@chapapp	2648, 2654, 2660, 2676
\CTEX@titleformat@n	91, 2816, 2822, 2863, 2870	\Hy@driver	1822
\CTEX@titleslabel@clear	3089, 3099	\Hy@MakeCurrentHrefAuto	2974
\CTEX@titleslabel@set	3083, 3095, 3099	\Hy@org@chapter	2955
\CTEX@toc@width@n	3124, 3139, 3149	\Hy@raisedlink	2975
\CTEX@todayold	2205, 2210	\Hy@unicodetrue	71
\CTEX@update@sectionformat@n	2734, 2861	\hyper@anchorend	2978
\CTEX@verbatim@font@hook	1617, 1621	\hyper@anchorstart	2977
\ctex@zhmap@endinput	4744	\HyPsd@ConvertToUnicode	71
\CTEXdigits	25, 25	\HyPsd@LoadUnicode	72
\CTEXifname	16, 16	\HyPsd@pdfencoding	71, 72
\CTEXnumber	25, 25, 25, 25, 25	\HyPsd@ToBigChars	71
\CTEXnumberline	22	\if@mainmatter	2637, 2692, 3172, 3561, 3562
\CTEXoptions	30, 30	\if@nobreak	2740
\ctexset	5, 5, 5, 5, 5, 5, 7, 11, 14, 14, 26, 27, 30, 30, 110	\if@noskipsec	2467, 2727, 2839
\CTEXsetup	30	\if@openright	2483, 2610, 2622, 3333, 3360
\CTEXthechapter	12, 13, 13, 16, 94	\if@tempswa	2616
\CTEXthesection	12, 13, 16, 97	\if@twocolumn	2491, 2668, 2677
\CTEXunderdotbasesep	30	\if@twoside	2609, 3156, 3177

\iffalse	53	\partpage	14, 97
\ifHy@implicit	2954	\PassOptionsToClass	42
\ifin@	53, 906	\PassOptionsToPackage	70
\iftrue	53	\pdfmapline	115, 118
\indexname	12	\pdfstringdef	71
\insertcontinuationtext	12	\pickup@font	50
\itdefault	52	\pickup@jfont	822
\itshape	48	\ProcessKeysOptions	28
\kaishu	8	\proofname	12, 80
\kanjiskip	75	\protected	95
\labelformat	99, 99	\protected@edef	2202, 2204, 2765, 3266
\leftmark	13	\ProvidesFile	121
\linewidth	77, 77	\ps@fancy	3183
\lishu	8, 8	\ps@headings	3155
\listfigurename	11	\refname	12, 12
\listtablename	11	\refstepcounter	99
\ltj@does@alt@set	52	\relax	45, 77
\ltj@getjfontnumber	53, 894	\reset@font	552, 1267
\ltj@set@stackfont	936, 937, 938	\rightmark	13
\ltj@allalchar	737	\rmfamily	48
\ltj@curjfont	51, 75	\roman	25
\ltj@pickup@altfont@auxy	879	\scantokens	131
\ltj@pickup@altfont@copy	892	\scriptfont@name	930, 937
\ltj@setpar@global	935	\section 9, 9, 14, 15, 19, 20, 20, 21, 21, 22, 23, 23, 31, 81, 97	
\ltj@tempcntc	53, 898	\section*	15
\ltjalchar	49	\sectionmark	13, 13
\m@ne	2505, 2524, 2635, 2691, 3165, 3172, 3195, 3941, 3942, 3943, 3944, 3945, 3946	\sectionname	15
\markboth	13	\sectionpage	14, 98
\markright	13	\selectfont	52, 68, 74, 74, 74, 75, 77
\math@bgroup	946	\setCJKfamilyfont	26, 26, 26
\math@egroup	951	\setCJKmainfont	26
\math@fonts	933	\sf@size	929
\mathbb	109	\sffamily	48
\maxdimen	11, 77	\size@update	77, 77, 2068, 2082
\meaning	131, 132, 133, 133	\sldefault	52
\newcommand	131	\songti	7
\newrobustcmd	131	\special	7, 71
\newtitlemark	96, 96	\split@name	884
\nfss@catcodes	55	\SplitArgument	83
\ngostype	70	\ssf@size	931
\normalsize	8, 8, 105	\strutbox	109
\nouppercase	45	\subparagraph	9, 9, 9, 9, 19, 35
\p@	1910, 1956, 2564, 2665, 2666, 2684, 2693, 2702, 2708, 2720, 3130, 3329, 3356, 3357, 3358, 3359, 3366, 3753, 3754, 3755, 3761, 3762, 3763, 3765, 3766, 3772, 3773, 3774, 3776, 3777, 3792, 3793, 3794, 3800, 3801, 3802, 3804, 3805, 3811, 3812, 3813, 3815, 3816	\subsectionname	15
\pagenumbering	25	\subsectionpage	14, 98
\pagestyle	9	\subst@correction	845, 852
\paragraph	9, 9, 9, 9, 9, 19, 35, 35	\tablename	12
\parindent	10, 30, 77, 79, 109	\textfont@name	928, 936
\parskip	84, 84, 84	\tf@size	927
\part	14, 20, 21, 83	\thechapter	13, 16, 23
\partmark	84	\thesection	13, 15, 16, 23, 97
\partname	15	\titleformat	93
		\titlespacing	93
		\today	11, 11
		\trans@languagepath	3610, 3636
		\ttfamily	48
		\ttl@a	3058, 3066

- \ttl@chapterout 3053
- \ttl@extract 93
- \ttl@setifthe 3112
- \ttl@setsubmark 95, 3085, 3091
- \ttl@settopmark 95, 3079, 3110
- \ttl@tocpart 3055, 3057
- \ttlh@hang 3019, 3029
- \ttlh@runin 3018, 3028
- \tw@ 1702
- \uppercase 45, 45
- \use@mathgroup 763, 942
- \usepackage 29, 30
- \verbatim@font 1615, 1619
- \wrong@fontshape 50
- \x@protect 5072
- \xdef 95
- \XeTeXglyphbounds 126
- \xkanjiskip 75, 75, 75, 75
- \yahei 8, 8
- \youyuan 8, 8
- \z@ 724, 727, 2522, 2577,
2626, 2688, 2712, 2731, 2773, 2809, 2827, 2841, 3038,
3158, 3179, 3187, 3198, 3754, 3762, 3773, 3793, 3801, 3812
- \zhdig 79
- \zhdigits 25
- \zhnum 79
- \zhnumber 25, 25, 25, 25
- \zihao 24, 24
- \ziju 24
- \zw 74
- tex commands:
 - \tex_def:D 5148
 - \tex_divide:D 77, 2113, 2154
 - \tex_edef:D 5202
 - \tex_endlinechar:D 561, 567
 - \tex_font:D 486, 4777
 - \tex_iftrue:D 3562
 - \tex_ignorespaces:D
1075, 1213, 1218, 2021, 2138, 3653, 4959
 - \tex_immediate:D 483
 - \tex_pagegoal:D 2424
 - \tex_parskip:D 2432
 - \tex_prevdepth:D 2431
 - \tex_topskip:D 2425
- \the 4446, 4449, 4450, 4469, 4473
- \thechapter 2642, 2645, 2693, 3173, 3195, 3363
- \theparagraph 3466
- \thepart 2509, 2525, 2546, 2564, 3303
- \thesection 3158, 3165, 3179, 3187, 3198, 3383
- \thesubparagraph 3480
- \thesubsection 3161, 3190, 3415
- \thesubsubsection 3449
- \thispagestyle 2489, 2490, 2612, 2623, 2625
- \tiny 3782, 3821
- tl commands:
 - \c_space_tl 487
 - \tl_clear:N 193, 366, 389,
1556, 1560, 1846, 1857, 1986, 2059, 2389, 2407, 3708, 5156
 - \tl_const:Nn
.. 3, 409, 414, 419, 424, 534, 672, 694, 965, 980, 1249,
1250, 1251, 1252, 1306, 2078, 2302, 2308, 3544, 3563, 3565
 - \tl_gput_right:Nn 464, 4953, 4955, 4977
 - \tl_gset:Nn 578, 581, 927,
929, 931, 1280, 1718, 1746, 1764, 1789, 1796, 1797, 3899
 - \tl_gset_eq:NN 1057, 1741
 - \tl_gset_rescan:Nnn 1287
 - \tl_head:N 3712
 - \tl_if_blank:nTF 1373, 1469, 1470, 1574, 1575, 1589, 1607
 - \tl_if_blank_p:n 1339
 - \tl_if_empty:NTF 785, 2131, 2314, 3280, 3862
 - \tl_if_empty:nTF 1659, 1672, 5119
 - \tl_if_eq:NNTF 1714, 2071, 3623
 - \tl_if_exist:NTF 84, 533, 535, 536, 537, 1248,
1250, 1251, 1252, 2227, 2258, 2269, 2272, 3556, 3646, 4976
 - \tl_if_in:NnTF 5183
 - \tl_map_inline:Nn 1485
 - \tl_map_inline:nn 3709
 - \tl_new:N 58, 128, 132, 535, 536, 537,
787, 788, 876, 962, 1017, 1043, 1063, 1064, 1085, 1421,
1562, 1734, 2000, 2079, 2092, 2229, 2300, 2301, 2921,
2922, 4052, 4966, 4967, 4976, 5037, 5038, 5159, 5160, 5161
 - \tl_put_left:Nn 1632, 1633, 1634, 1969, 4986
 - \tl_put_right:Nn 373, 933, 1569, 1602,
2330, 2354, 2362, 2364, 2366, 2368, 2370, 2372, 3712, 3713
 - \tl_replace_all:Nnn 1336, 1377
 - \tl_rescan:nn 133, 5201, 5210
 - \tl_set:Nn 114,
115, 129, 188, 221, 222, 482, 514, 515, 516, 574, 776,
778, 789, 828, 877, 959, 960, 1001, 1068, 1080, 1204,
1335, 1374, 1376, 1417, 1590, 1594, 1628, 1629, 1630,
1631, 1640, 1736, 1985, 2001, 2035, 2044, 2055, 2093,
2141, 2230, 2387, 2390, 2394, 3015, 3034, 3053, 3058,
3066, 3622, 3840, 4053, 4057, 4099, 4100, 4101, 5031, 5032
 - \tl_set_eq:NN
794, 795, 842, 928, 930, 974, 1069, 1094, 1524, 1536, 2080
 - \tl_set_rescan:Nnn 5157
 - \tl_tail:N 3020
 - \tl_tail:n 3030
 - \tl_to_str:n 125, 126, 5100, 5129, 5170, 5171
 - \tl_trim_spaces:n 1553
- \today 2205, 2210, 2213, 2218
- today 11, 2205
- token commands:
 - \token_get_replacement_spec:N 5057, 5079
 - \token_to_meaning:N 5114
 - \token_to_str:N
.. 54, 909, 918, 927, 929, 931, 1122, 1123, 1124, 1126,
1133, 1134, 1135, 1197, 2255, 2256, 3661, 4784, 4944, 5016
- \toks 4446, 4447, 4449, 4450, 4469, 4473
- \topsep 3765, 3776, 3804, 3815
- \tracingfonts 1702
- \TrimSpaces 433

\ttdefault 1724
 \ttfamily 546, 549, 1261, 1264, 1634
 two commands:
 \c_two_hundred_fifty_six 577, 580
 \twocolumn 2617
 \typeout 2642, 2643, 3053

U

um commands:
 \um_input_math_symbol_table: 756, 760
 \um_sym:nnn 759
 um internal commands:
 _um_input_math_symbol_table: 768
 _um_switchto_literal: 765
 _um_sym:nnn 767
 \undefined 1708, 4471
 \unless 2774, 2810, 2828
 \unskip 2844
 \updefault ... 970, 976, 978, 982, 984, 1308, 1310, 1313, 1317
 \upshape 1649, 1652
 use commands:
 \use:N 796, 834, 835, 885, 927, 929, 931,
 985, 1035, 1537, 2448, 2454, 2463, 2893, 5184, 5204, 5218
 \use:n 64, 122, 133, 388, 773, 884, 967,
 1016, 1152, 1277, 1281, 1379, 1553, 1735, 1888, 2414,
 2859, 2870, 3135, 3554, 3671, 3871, 5053, 5095, 5146, 5167
 \use:nn 5212
 \use_i:nn ... 108, 906, 2440, 2442, 4997, 5002, 5022, 5027
 \use_i:nnn 261
 \use_ii:nn 109, 906, 2444, 4997, 5002, 5022, 5027
 \use_ii:nnn 134, 266
 \use_iii:nnn 271
 \use_none:n .. 189, 194, 199, 1052, 1280, 1414, 1889, 2191
 \use_none:nn 499
 \usebeamercolor 3208, 3226, 3244, 3346, 3406, 3440
 \usebeamerfont 3208, 3217,
 3226, 3235, 3244, 3253, 3345, 3348, 3405, 3408, 3439, 3442
 \uselanguage 3601, 3606, 3627, 3632
 \usepackage 4944
 UTF8 6, 221

V

\verse 3850
 \vfil 2497, 2604
 \vskip 2534, 2537,
 2564, 2584, 2587, 2607, 2693, 2702, 2705, 2720, 2723,
 2830, 2833, 3209, 3227, 3245, 3329, 3347, 3366, 3407, 3441
 \vspace 2500, 2684, 2687, 2708, 2711

W

winfonts 29
 \write 4486

X

\x 4448, 4451,
 4452, 4453, 4454, 4455, 4456, 4457, 4458, 4459, 4460,
 4461, 4462, 4463, 4464, 4465, 4466, 4467, 4474, 4484, 4488
 \xdef 4487
 \xeCJKsetup 678, 1853, 1928, 2016, 2019, 2022, 2037
 xetex commands:
 \xetex_charglyph:D 4781
 \xetex_glyphbounds:D 4805

Y

\yahei 4438
 \youyuan 4436

Z

\zhdig 2197
 \zhdigits 25, 2204
 zhmap 7, 29, 256
 \zhnum 2196
 zhnum commands:
 \zhnum_counter:n 2188
 \zhnumber 25, 2202
 \zhnumsetup 2214, 2219
 \zhtoday 2213, 2218
 \zihao 24, 3652, 3661
 zihao 8, 158
 \ziju 24, 2137
 \zw 725, 1914, 1957, 2002