

CT_EX 宏集手册

ctex.org

2016/05/14 v2.4.1*

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、XeL^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	23
第 2 节 简明教程	3	8.1 字号与间距	23
2.1 CT _E X 宏集的组成	3	8.2 中文数字转换	24
2.2 CT _E X 宏集的安装和更新	3	8.3 杂项	24
2.3 使用 CT _E X 文档类	4	第 9 节 LuaL ^A T _E X 下的中文支持方式	24
2.4 使用 ctex 宏包	4	9.1 LuaL ^A T _E X 下替代字体的设置	25
第 3 节 宏包选项与 \ctexset 命令	5	第 10 节 CT _E X 宏集的配置文件	26
第 4 节 编译方式、编码与中文字库	5	10.1 修改宏包默认选项	26
4.1 编译方式	5	10.2 宏包载入后的配置	27
4.2 中文编码	6	10.3 配置标题中文翻译	27
4.3 中文字库	6	10.4 自定义字体集	27
第 5 节 排版格式设定	8	第 11 节 对旧版本的兼容性	27
5.1 文档默认字号	8	11.1 CT _E X 0.8a 及以前的版本	27
5.2 章节标题风格	9	11.2 CT _E X 0.9–CT _E X 1.0d	28
5.3 排版方案选项	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	29
第 6 节 文档汉化	11	11.4 CT _E X 2.2 之前的版本	30
6.1 日期汉化	11	第 12 节 宏集依赖情况与手工安装方法	30
6.2 文档标题汉化	11	第 13 节 开发人员	31
6.3 页面格式设置与汉化	12	第 14 节 代码实现	32
第 7 节 章节标题格式设置	14	版本历史	137
		代码索引	138

*ctex-kit rev. f745b21.

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 LuaTeX-j_a 宏包

第 2 节 简明教程

2.1 C_TE_X 宏集组成

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

表 1 C_TE_X 宏集的组成

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

2.2 C_TE_X 宏集的安装和更新

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

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

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

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

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

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

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

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

⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

2.3 使用 CT_EX 文档类

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

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

例 1

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

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

以下是使用 `ctexbeamer` 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

`\ctexset {<键值列表>}`

New: 2014-03-18

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

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

例 4

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

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

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

表 2 CT_EX 宏集的中文支持方式

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

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

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

不同的编译方式和中文支持方式会在一定程度上影响 $\text{CT}_{\text{E}}\text{X}$ 宏集的行为, 比如宏包对编码的处理。在用户使用 $\text{X}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 、 $\text{Lua}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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-j 华文字库	LuaTeX-j 中易字库 + 微软雅黑	LuaTeX-j 中易字库	LuaTeX-j 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-j 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 = <code><true false zhmCJK></code>
-------	---	--

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

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

New: 2014-03-08

指定 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 <code>pdfL^AT_EX</code> 。
fandol	使用 Fandol 中文字体, 不支持 <code>pdfL^AT_EX</code> 。
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 <code>(pdf)L^AT_EX</code> 。
ubuntu	使用 Ubuntu 系统下的文泉驿和文鼎字体。
windows	使用简体中文 Windows 系统下的中文字体, 自动判断 Windows 系统版本, 采用 <code>windowsnew</code> 或 <code>windowsold</code> 的设置。
windowsnew	使用简体中文 Windows Vista 或之后系统下的中易字体和微软雅黑字体。
windowsold	使用简体中文 Windows XP 或之前系统下的中易字体。

注意: 使用 `(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 _E X 宏集将载入名为 <code>ctex-fontset-<name>.def</code> 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录树下合适位置建立一个名为 <code>ctex-fontset-<name>.def</code> 的文件, 在这个文件里面自定义中文字体。然后通过使用 <code>fontset=<name></code> 选项来调用它。字体配置文件的具体写法可以参考 C _T _E X 宏集 <code>fontset</code> 目录下的字体配置文件。

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

例 5

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

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 <code>\CJKfamily{zhsong}</code> 。
---------	--

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

¹⁰`zhmCJK` 宏包基于 `zhmetrics` 和 `CJK` 宏包, 提供与 `xeCJK` 宏包类似的用户接口。

¹¹可以使用 `zhmetrics` 宏包提供的脚本 `CTeXFonts.lua`。

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

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

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

在 windowsnew 字库中,还有 `\yahei`。

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

第 5 节 排版格式设定

5.1 文档默认字号

`zihao` ☆

New: 2015-05-06

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

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

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

表 4 标准字体命令与字号的对应

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

10pt ☆

11pt ☆

12pt ☆

C_T_E_X 文档类是在 L_AT_E_X 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_E_X 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, 只有在使用了除了 `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: 2015-04-15

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 = `<true|false|auto>`

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。

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

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

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

linespread ☆ linespread = `<数值>`

New: 2014-04-23

接受一个浮点数值,设置行距倍数。本选项的初始值与 scheme 有关。

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

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

New: 2014-03-13

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

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

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

例 8

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

linestretch ☆ linestretch = `<数值或长度>`

New: 2014-03-26

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

¹⁵LaTeX 将单个换行视作一个空格。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_εX 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 5 月 14 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 5 月 14 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年五月十四日”。使用全汉字的日期格式。
`old` 效果为“May 14, 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_εX 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{\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{\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`, `beforeskip`, `afterskip`, `indent`, `pagestyle` 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` 和 `afterindent` 这两个格式无效。

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 $\text{\translate{Section}}$
section	同右	{}	
subsection (beamer)	{}	{\subsectionname\space}	原 \subsectionname 为 $\text{\translate{Subsection}}$
subsection	同右	{}	
subsubsection	同右	{}	
paragraph	同右	{}	
subparagraph	同右	{}	

.../number ★

number = {<数字输出命令>}

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

```

.../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 15

```

\ctexset{
  chapter/format = \sffamily\raggedright,
  section/format += \sffamily
}

```

```

.../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 19)。

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

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

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 16

```
\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 17

```
\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+ *
New: 2015-06-19
```

aftertitle = {<代码>}
aftertitle+= {<代码>}
aftertitle 选项的参数 <代码> 将被插入到章节标题内容之后。
aftertitle+ 选项用于在原有代码后面附加代码。
aftertitle 选项的默认值, 在 scheme 选项的不同取值下相同。
sub3section 或 sub4section 宏包选项(见 5.2 节)会影响 aftertitle 选项的默认值。

表 11 aftertitle 选项的默认设置

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

表 12 runin 选项的默认设置

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

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

```
subsection/runin  ★
```

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

```
paragraph/runin  ★
```

```
subparagraph/runin ★
```

New: 2015-06-27

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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

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

New: 2015-06-27

afterindent 选项用于设置章节标题后首段的缩进。

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

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

表 13 afterindent 选项的默认设置

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

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

Updated: 2016-05-10

beforeskip 选项用于设置章节标题前的垂直间距。

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

表 14 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

```
part/fixbeforeskip ★ fixbeforeskip = true|false
```

```
chapter/fixbeforeskip ★
```

New: 2016-05-10

book 和 report 类的 \part 和 \chapter 标题与正文顶部的距离除了由 beforeskip 选项设置的垂直间距外, 还会有一个空行。fixbeforeskip 选项用于抑制这个多余空行。

`.../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。

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	<code>\parindent</code>
(sub3section)	0pt
(sub4section)	同上

`.../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 18

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

`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`)。

`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。

表 17 pagestyle 选项的默认设置

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

 appendix/name

 Updated: 2014-03-08

name = {⟨前名字⟩,⟨后名字⟩}

name = {⟨前名字⟩}

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 18 appendix/name 选项的默认设置

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

 appendix/number ★

number = {⟨数字输出命令⟩}

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

用法与普通章节的 **number** 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

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

表 19 appendix/number 选项的默认设置

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

我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 19

```
\ctexset {
  chapter = {
    beforekip      = 0pt,
    fixbeforekip   = 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：

第 1 章

熟悉 L^AT_EX

第8节 实用命令

8.1 字号与间距

\zihao

\zihao {〈字号〉}

Updated: 2014-03-08

用于调整字号大小。其中〈字号〉的有效值共有 16 个,如表 20 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 20 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

$\backslash\text{ziju}$	$\backslash\text{ziju} \{(\text{中文字符宽度的倍数})\}$
Updated: 2014-03-28	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 $\backslash\text{ccwd}$ 的值,但不会影响英文字距。

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

8.2 中文数字转换

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

$\backslash\text{chinese}$	$\backslash\text{chinese} \{(\text{counter})\}$
Updated: 2016-05-01	$\backslash\text{pagenumbering} \{\text{chinese}\}$
	$\backslash\text{chinese}$ 命令与 $\backslash\text{roman}$ 等命令的用法类似,作用在一个 \LaTeX 计数器上,将计数器的值以中文数字的形式输出。

$\backslash\text{zhnumber}$	$\backslash\text{zhnumber} \{(\text{number})\}$
-----------------------------	---

New: 2014-03-08

以中文格式输出数字。这里的数字可以是整数、小数和分数。

$\backslash\text{zhdigits}$	$\backslash\text{zhdigits} \{(\text{number})\}$
-----------------------------	---

New: 2014-03-08

将阿拉伯数字转换为中文数字串。

$\backslash\text{CTEXnumber}$	$\backslash\text{CTEXnumber} \backslash(\text{macro}) \{(\text{number})\}$
-------------------------------	--

$\backslash(\text{macro})$ 必须是一个 $\text{T}\text{\TeX}$ 宏,不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 $\langle\text{number}\rangle$ 转为中文数字,最后将结果存储在 $\backslash(\text{macro})$ 里。对 $\backslash(\text{macro})$ 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。

$\backslash\text{CTEXdigits}$	$\backslash\text{CTEXdigits} \backslash(\text{macro}) \{(\text{number})\}$
-------------------------------	--

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

8.3 杂项

$\backslash\text{CTeX}$	用于显示 $\text{CT}\text{\TeX}$ 标志。
-------------------------	---------------------------------

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

在 $\text{Lua}\text{\LaTeX}$ 下, $\text{CT}\text{\TeX}$ 宏集依赖 $\text{Lua}\text{\TeX}-\text{ja}$ 宏包来完成中文支持。该宏包是日本 $\text{T}\text{\TeX}$ 社区的北川弘典、前田一贵、八登崇之等人开发的,设计目的主要是在 $\text{Lua}\text{\TeX}$ 引擎下实现日本

*p*TeX 引擎的(大部分)功能。它为了兼容 *p*TeX 的使用习惯,对 *La*TeX_{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,因而 CTeX 禁用了它在 *La*TeX 格式下的大部分设置,只保留了必要的部分。同时修改了它的字体设置方式,使得相关命令与 *xeCJK* 宏包大致相同。}

20150420 版以后的 *Lua*TeX-ja 宏包开始支持竖排,但 CTeX 暂不支持竖排。

9.1 *Lua*TeX 下替代字体的设置

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

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

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

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

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

例 20

```

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

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

```

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

```

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

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

与 C_TE_X 宏集的源代码一样,配置文件采用 L_AT_EX3 的语法编写。

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

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

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

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

10.1 修改宏包默认选项

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

例 21

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


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

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

10.2 宏包载入后的配置

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

例 22

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

例 23

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

11.1 C_TE_X 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

<hr/> cs4size c5size <hr/>	分别相当于 <code>zihao=-4</code> 和 <code>zihao=5</code> , 过时选项。
<hr/> CCT CCTfont <hr/>	相关选项已删除。
<hr/> indent noindent <hr/>	<code>indent</code> 和 <code>noindent</code> 什么也不做, 过时选项。 在中文版式下, <code>ctex</code> 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 <code>afterindent</code> 选项的值来确定。
<hr/> zhmap nozhmap <hr/>	<code>zhmap</code> 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 <code>zhmCJK</code> 作为底层中文处理宏包。(4.3 节) <code>nozhmap</code> 选项相当于 <code>zhmap=false</code> 。过时选项。
<hr/> winfonts adobefonts nofonts <hr/>	宏包选项 <code>winfonts</code> 相当于 <code>fontset=windows</code> , <code>adobefonts</code> 相当于 <code>fontset=adobe</code> , <code>nofonts</code> 相当于 <code>fontset=none</code> 。这几个选项是过时选项, 对于新文档, 应使用 <code>fontset</code> 选项设置不同字体集。 另外, 第 2 版 C _T E _X 宏包的默认字体不再是 Windows 系统字体, 而是根据检测到的操作系统选择使用 Windows、Mac 的系统字体还是 Fandol 字体(4.3 节)。
<hr/> punct nopunct <hr/>	旧版本中宏包 <code>punct</code> 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 <code>punct</code> 选项相当于 <code>punct=quanjiao</code> 。 旧版宏包中 <code>nopunct</code> 选项的效果大致相当于 <code>punct=plain</code> 。过时选项, 不推荐使用。
<hr/> cap nocap <hr/>	原有的 <code>cap</code> 和 <code>nocap</code> 选项由新的 <code>scheme</code> 选项代替。(5.3 节) <code>cap</code> 选项相当于 <code>scheme = chinese</code> , <code>nocap</code> 选项相当于 <code>scheme = plain</code> 。它们均已过时, 仅因兼容性而保留。
<hr/> space nospace <hr/>	新版本宏包 <code>space</code> 选项增加真假值参数。(5.3 节) <code>nospace</code> 选项相当于 <code>space=false</code> , 成为过时选项。
<hr/> fancyhdr <hr/>	新版本宏包中总是自动处理对 <code>fancyhdr</code> 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 <code>fancyhdr</code> 宏包。 <code>fancyhdr</code> 选项过时, 因兼容性保留, 功能是载入 <code>fancyhdr</code> 宏包。

<u>hyperref</u>	<p>新版本宏包中总是自动处理对 <code>hyperref</code> 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 <code>hyperref</code> 宏包。</p> <p><code>hyperref</code> 选项过时, 因兼容性保留, 功能是在导言区末尾载入 <code>hyperref</code> 宏包。</p>
<u>fntef</u>	<p>旧版本的 <code>fntef</code> 选项用于统一 <code>CCTfntef</code> 与 <code>CJKfntef</code> 的界面, 新版本 <code>CT_EX</code> 宏集不再支持 <code>CCT</code>, 而是直接载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包并禁用其彩色设置。该选项是过时选项。</p>
<u>\CTEXunderdot \CTEXunderline \CTEXunderdblline \CTEXunderwave \CTEXsout \CTEXxout \CTEXfilltwosides</u>	<p>在调用 <code>fntef</code> 宏包选项的同时, 旧版本 <code>CT_EX</code> 宏包由于需要支持 <code>CCT</code> 系统, 会将以 <code>\CJK</code> 开头的 <code>\CJKunderline</code> 等宏换名为以 <code>\CTEX</code> 开头的 <code>\CTEXunderline</code> 等宏。此功能在新版本的 <code>CT_EX</code> 宏集中已失去意义。</p> <p>此外, 在 <code>pdf_TE_X</code> 引擎下, 用于设置格式的 <code>\CJKunderdotbasesep</code> 等宏也被换名为 <code>\CTEXunderdotbasesep</code> 等宏。</p> <p>在新版本中, 上述由 <code>fntef</code> 衍生的相关宏都成为过时命令。</p>
<u>\CTEXsetfont</u>	<p>更新当前的中文字体信息, 包括当前字距 (<code>\ccwd</code>) 和段首缩进 (<code>\parindent</code>)。一般来说, 用户无需使用这个命令。</p>
<u>\CTEXindent</u>	<p>更新 <code>\ccwd</code> 宽度后设置 <code>\parindent=2\ccwd</code>。过时命令。</p>
<u>\CTEXnoindent</u>	<p>设置 <code>\parindent=0pt</code>。过时命令。</p>
<u>\CTEXsetup</u>	<p><code>\CTEXsetup[⟨选项⟩]{⟨标题⟩}</code></p> <p>相当于设置了 <code>\ctexset{⟨标题⟩ = {⟨选项⟩}}</code>。过时命令。</p>
<u>\CTEXoptions</u>	<p><code>\CTEXoptions[⟨选项⟩]</code></p> <p>相当于设置了 <code>\ctexset{⟨选项⟩}</code>。过时命令。</p>
<u>\Chinese</u>	<p><code>\Chinese{⟨counter⟩}</code></p> <p>新版宏集中 <code>\chinese</code> 统一了旧版本中 <code>\chinese</code> 和 <code>\Chinese</code> 的功能。因此, 该命令已过时。</p>
<u>captiondelimenter</u>	<p>原为 <code>\CTEXoptions</code> 命令的选项, 用于控制 <code>\caption</code> 编号后面的标点。此选项已过时, 并在新版本的 <code>CT_EX</code> 宏包中失效。</p> <p>可以使用 <code>caption</code> 宏包的 <code>labelsep</code> 选项来完成同样的功能。</p>

例 24

```
% 代替 \CTEXoptions[captiondelimenter={:}]
\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`。

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

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, 它们还依赖
 - `etex` 宏包。
- `indentfirst` 宏包, 属于 `tools` 宏集。
- `everysel` 宏包, 属于 `ms` 宏集。
- `zhnumber` 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- `CJK` 宏集, 它的下划线功能依赖 `ulem` 宏包。
- `CJKpunct` 宏包。
- `xCJK2uni` 宏包。
- `zhmetrics` 宏包。
- `zhmCJK` 宏包。
- ➡ 以上是使用 `pdfLATEX` 或 `LATEX + DVIPDFMx` 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 `tipa` 宏包。
- ➡ 以上是使用 `XLLATEX` 编译时的依赖项。
- `LuaTEX-ja` 宏集, 它还依赖
 - `oberdiek` 宏集。
 - `xkeyval` 宏包。
 - `lualibs` 宏包。
 - `luaotfload` 宏包, 它还依赖 `luatexbase` 宏包。

- ➡ 以上是使用 Lua \LaTeX 编译时的依赖项。
 - zhmetrics-uptex 宏包。
- ➡ 以上是使用 up \LaTeX 编译时的依赖项。

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

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

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

若要手工安装, 请遵循如下步骤:

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

第 13 节 开发人员

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

目前比较活跃的开发维护人员是刘海洋、李清和黄晨成。

参考文献

- [1] DONALD ERVIN KNUTH. *The \TeX book, Computers & Typesetting*, volume A. Addison-Wesley, 1986
- [2] FRANK MITTELBAACH and MICHEL GOOSSENS. *The \LaTeX 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,~uplatex,~or~aplatex. }
48 \file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }
49 {
50   \str_const:Nx \c__ctex_engine_file_str
51   { ctex-engine- \c__ctex_engine_str .def }
52 }
53 { \msg_critical:nnx { ctex } { engine-not-supported } { \c__ctex_engine_str } }

```

(End definition for \c__ctex_engine_str and \c__ctex_engine_file_str.)

54 </class|ctex>


```

55 <*class|ctex|ctexheading>
56 \RequirePackage { ctexhook , ctexpatch }
57 <!ctexheading>\RequirePackage { fix-cm , everyysel }

```

14.1 内部函数与变量

```

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_dim
\l__ctex_tmp_box
58 \tl_new:N \l__ctex_tmp_tl
59 \int_new:N \l__ctex_tmp_int
60 <!ctexheading>\dim_new:N \l__ctex_tmp_dim
61 \box_new:N \l__ctex_tmp_box

```

(End definition for \l__ctex_tmp_tl and others.)

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

```

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

```

(End definition for \ctex_file_wrapper:nnn.)

`\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} } { } }

```

(End definition for \ctex_file_input:n.)

`\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 }

```

(End definition for \ctex_scheme_input:n.)

`\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>

```

(End definition for `\g__ctex_section_depth_flag`.)

```

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: }

```

(End definition for `\ifctexpdf`.)

`\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 }

```

(End definition for `\ctex_if_preamble:TF`.)

`\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 }

```

(End definition for `\ctex_set_default_ccwd:Nn`.)

`\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 }

```

(End definition for \ctex_default_pt:n.)

\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 } }
```

(End definition for \l__ctex_encoding_tl.)

\g__ctex_zhmCJK_bool 是否使用 zhmCJK 宏包。

```
131 \bool_new:N \g__ctex_zhmCJK_bool
```

(End definition for \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
```

(End definition for \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
```

(End definition for \ctex_if_autoindent_touched:F.)

\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
```

(End definition for \ctex_zhmap_case:nnn.)

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

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

(End definition for \ctex_at_end:n.)

\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>
```

(End definition for \g__ctex_std_options_clist.)

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>

```

(End definition for `\g__ctex_font_size_flag`.)

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 } ,
164 <ctexsize> }
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 ,

```

(End definition for `zihao`. This function is documented on page 23.)

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 ,

```

(End definition for `linespread`. This function is documented on page 10.)

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     } ,

```

(End definition for autoindent. This function is documented on page 10.)

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 ,

```

(End definition for indent. This function is documented on page 28.)

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 ,

```

(End definition for GBK and UTF8. These functions are documented on page 6.)

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

```

225     fontset .tl_gset:N = \g__ctex_fontset_tl ,
226     nofonts .code:n =
227     {
228         \msg_warning:nnn { ctex } { deprecated-option }
229         {
230             Option~ `fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
231             fontset~ option.~
232         }
233         \keys_set:nn { ctex / option } { fontset = none }
234     } ,
235     adobefonts .code:n =
236     {
237         \msg_warning:nnn { ctex } { deprecated-option }
238         {
239             Option~ `fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
240             fontset~ option.~
241         }
242         \keys_set:nn { ctex / option } { fontset = adobe }
243     } ,
244     winfonts .code:n =

```

```

245     {
246       \msg_warning:nnn { ctex } { deprecated-option }
247       {
248         Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
249         fontset~ option.
250       }
251       \keys_set:nn { ctex / option } { fontset = windows }
252     } ,
253     nofonts      .value_forbidden:n = true ,
254     winfonts     .value_forbidden:n = true ,
255     adobefonts   .value_forbidden:n = true ,

```

(End definition for `fontset`. This function is documented on page 7.)

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 ,

```

(End definition for `zhmap`. This function is documented on page 28.)

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 ,

```

(End definition for `punct`. This function is documented on page 28.)

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 ,

```

(End definition for `space`. This function is documented on page 28.)

heading

```

304     heading .bool_set:N = \l__ctex_heading_bool ,

```

(End definition for `heading`. This function is documented on page 9.)

```

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

```

sub3section

sub4section

```

307 <!*beamer>
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>

```

(End definition for `sub3section` and `sub4section`. These functions are documented on page 9.)

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 ,

```

(End definition for `scheme`. This function is documented on page 9.)

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 }
```

(End definition for *fntef*, *fancyhdr*, and *hyperref*. These functions are documented on page 29.)

```

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
```

(End definition for 10pt, 11pt, and 12pt. These functions are documented on page 8.)

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>

397 <|ctexsize>\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>

```

(End definition for \ctexset. This function is documented on page 5.)

\CTEXsetup 过时命令。
\CTEXoptions

```

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>

```

(End definition for \CTEXsetup and \CTEXoptions. These functions are documented on page 29.)

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>

```

(End definition for \ctex_set_zhmap:n.)

```
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 }

```

(End definition for \c__ctex_cmap_encoding_seq.)

\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

```

(End definition for \ctex_family_cmap:nn.)

\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 }

```

(End definition for \ctex_add_cmap:n.)

\DeclareFontFamily 只在 pdf^{La}T_EX 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: }

```

(End definition for \DeclareFontFamily.)

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 { \file_input:n {#4} }
518 }
519 \@onlypreamble \ctex_load_zhmap:nnnn

```

(End definition for \ctex_load_zhmap:nnnn.)

```

520 \tl_if_exist:NF \CJKfamilydefault
521 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
522 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
523 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
524 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
525 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
526 { \CJKfamily { \CJKrmdefault } }
527 { }

```

```

528 { \ctex_patch_failure:N \rmfamily }
529 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
530 { \CJKfamily { \CJKsfdefault } }
531 { }
532 { \ctex_patch_failure:N \sffamily }
533 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
534 { \CJKfamily { \CJKttdefault } }
535 { }
536 { \ctex_patch_failure:N \ttfamily }
537 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
538 { \CJKfamily { \CJKfamilydefault } }
539 { \cs_set_eq:NN \reset@font \normalfont }
540 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```
541 \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。

```

542 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
543 {
544   \ctex_file_wrapper:nnn
545   {
546     \char_set_catcode_other:n { 60 } % <
547     \char_set_catcode_math_superscript:n { 94 } % ^
548     \int_set_eq:NN \tex_endlinechar:D \c_minus_one
549   }
550   { \file_input:n {#1} }
551   {
552     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
553     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
554     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
555   }
556 }
557 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

(End definition for `\ctex_CJK_input:n` and `\CJK@input`.)

`\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。

```

558 \if_cs_exist:N \CJK@surr
559 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
560 {
561   \tl_set:Nx \l__ctex_tmp_tl {#2}
562   \int_set:Nn \l__ctex_tmp_int
563   { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
564   \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
565   { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
566   {
567     \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
568     \tl_gset:Nx #1
569     {
570       \int_to_Hex:n
571       { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
572       \int_to_Hex:n
573       { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
574     }
575   }
576 }

```

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

```

577 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
578 \fi:

```

(End definition for \ctex_plane_to_utfxvibe:Nn and \CJK@surr.)

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

```

579 \AtBeginDocument
580 {
581   \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
582   { \punctstyle { \l__ctex_punct_tl } }
583 }

```

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

```

584 \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。

```

585 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
586 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
587 {
588   \CJK@makeActive
589   \CJK@loadBinding { standard }
590 }

```

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。zhm-CJK 已经提供类似功能。注意先使用 \ctex_update_default_family: 更新 \CJKfamilydefault。

```

591 \exp_args:Nx \ctex_at_end_preamble:n
592 {
593   \exp_not:N \CJK@envStart
594   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
595   \exp_not:N \CJKtilde
596 }

```

zhmCJK 判断结束。

```

597 \fi:

```

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

```
598 <@@=>
599 \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。

```
600 <@@=ctex>
```

(End definition for \ctex_auto_ignorespaces:.)

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

```
601 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
602 {
603   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
604     { \cs_set_eq:NN \CJK@ignorespaces #1 }
605   \ctex_set_ignorespaces:
606 }
607 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
608 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }
```

(End definition for \ctex_ignorespaces_case:N and \ctex_set_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。

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

(End definition for \CJKhook.)

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

```
610 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
611 {
612   \clist_map_inline:Nn \c__ctex_punct_family_clist
613   {
614     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
615     {
616       \cs_set_eq:cc
617       { CJKpunct@ ##1 @spaces }
618       { c__ctex_ #1 ##1 _punct_spaces_tl }
619     }
620   }
621 }
622 \clist_const:Nn \c__ctex_punct_family_clist
623 {
624   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
625   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
626 }
```

(End definition for \ctex_punct_set:n.)

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

```
627 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
628 {
629   \cs_if_free:cF { CJKpunct@ #2 @spaces }
630   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
631 }
```

(End definition for \ctex_punct_map_family:nn.)

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

```

632 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
633 {
634   \clist_map_inline:nn {#1}
635   {
636     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
637     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
638   }
639 }
640 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
641 {
642   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
643   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
644   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
645   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
646   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
647   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
648 }

```

(End definition for `\ctex_punct_map_bfseries:nn`.)

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

```

649 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
650 {
651   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
652   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
653   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
654   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
655   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
656   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
657 }

```

(End definition for `\ctex_punct_map_itshape:nn`.)

`\ctex_punct_space:nn` 定义标点的边界信息。

`\ctexspadef`

```

658 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
659 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
660 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

(End definition for `\ctex_punct_space:nn` and `\ctexspadef`.)

载入边界信息文件。

```

661 \ctex_file_input:n { ctexspa.def }

```

```

662 </pdfTeX>

```

14.4.2 ctex-engine-xetex.def

```

663 <*xetex>
664 \RequirePackage { xeCJK }
665 \exp_args:Nx \xeCJKsetup
666 {
667   LoadFandol = false ,
668   AutoFakeBold = true ,
669   PunctStyle = \l__ctex_punct_tl
670 }

```

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`。

```

671 \@ifpackagelater { fontspec } { 2014/05/25 } { }
672 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

673 `</xetex>`

14.4.3 ctex-engine-luatex.def

674 `<*luatex>`

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

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

14.4.3.1 LuaTeX-jā 的默认设置

689 `\ExplSyntaxOff`

以下设置抄录自 ltjdefs.sty。

```
690 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
691 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
692 \ltjdefcharrange{3}{%
693   "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
694 \ltjdefcharrange{4}{%
695   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
696   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
697   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
698 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
699 \ltjdefcharrange{6}{%
700   "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
701   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
702 \ltjdefcharrange{7}{
703   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
704   "AC00-"D7FF}
705 \ltjdefcharrange{8}{ "A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
706 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, -8}}
707 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}
```

以下设置抄录自 ltj-latex.sty。

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

14.4.3.2 LuaTeX-jā 的补丁

719 `<@@=ctex_ltj>`

在 L^AT_EX 下, LuaTeX-jā 对 fontspec、xunicode、unicode-math 和 listings 打了补丁。其中

前三个是把 `\char` 换成 `\ltjalchar`, 确保字符是 `ALchar` 类。我们这里用 `xunicode-addon` 来处理 `xunicode`。

```

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

```

对 `fontspec` 沿用 `LuaTeX-ja` 的补丁。

```

728 \RequirePackage { lltjp-fontspec }

```

`lltjp-unicode-math` 让数学符号命令成为普通的文字宏。为了避免它被展开, 应该用 `\protected` 来定义。

```

729 \cs_new_protected_nopar:Npn \__ctex_ltj_um_define_math_chars:
730 {
731   \group_begin:
732   \cs_set_protected:Npn \__um_sym:nnn ##1##2##3
733   {
734     \tl_if_in:nnT
735     {
736       \mathord \mathalpha \mathbin \mathrel
737       \mathpunct \mathop \mathfence
738     }
739     { ##3 }
740     { \__ctex_ltj_um_char:Nn ##2 { ##1 } }
741   }
742   \__um_input_math_symbol_table:
743   \group_end:
744 }
745 \cs_new_protected:Npn \__ctex_ltj_um_char:Nn #1#2
746 {
747   \__ctex_ltj_um_char_aux:Nx #1 { \char_generate:nn {#2} { 12 } }
748   \ltjsetmathletter {#2}
749 }
750 \cs_new_protected:Npn \__ctex_ltj_um_char_aux:Nn #1#2
751 {
752   \cs_gset_protected_nopar:Npn #1
753   {
754     \mode_if_math:TF
755     {#2}
756     {
757       {
758         \lua_now_x:n { tex.globaldefs = 0 }
759         \ltj@allalchar #2
760       }
761     }
762   }
763 }
764 \cs_generate_variant:Nn \__ctex_ltj_um_char_aux:Nn { Nx }
765 \ctex_at_end_package:nn { unicode-math }
766 {
767   \cs_set_eq:NN \__um_define_math_chars: \__ctex_ltj_um_define_math_chars:
768   \@ifpackagelater { unicode-math } { 2015/06/28 }
769   {
770     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
771     \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
772     { \__um_switchto_literal: }
773   }
774   { }
775 }

```

对 `listings` 的补丁是让代码环境支持 `JAchar` 类。`LuaTeX-ja` 的补丁会将代码目录标题改为日

文,我们不需要。

```

776 \ctex_at_end_package:n { listings }
777 {
778   \use:x
779   {
780     \exp_not:N \RequirePackage { lltpj-listings }
781     \tl_set:Nn \exp_not:N \lstlistingname
782       { \exp_not:o { \lstlistingname } }
783     \tl_set:Nn \exp_not:N \lstlistlistingname
784       { \exp_not:o { \lstlistlistingname } }
785   }
786 }

```

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。

```

787 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
788 {
789   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
790     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
791 }
792 \tl_new:N \CJK@family
793 \tl_new:N \l__ctex_ltj_current_font_tl
794 \tl_set:Nn \l__ctex_ltj_current_font_tl
795 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

(End definition for `\ctex_ltj_select_font:` and `\CJK@family`.)

`__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。

```

796 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
797 {
798   \group_begin:
799     \tl_set_eq:NN \f@encoding \CJK@encoding
800     \tl_set_eq:NN \f@family \CJK@family
801     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
802     \ctex_ltj_pickup_font:
803   \group_end:
804   \font@name
805   \__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。

```

806   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
807     { \__ctex_ltj_select_font_aux: }
808 }
809 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
810 {
811   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
812   \cs_gset_nopar:Npx \font@name {#1}
813 }
814 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
815 { \cs_gset_eq:NN \font@name \__ctex_ltj_save_fontname:w }

```

(End definition for `__ctex_ltj_select_font_aux:`.)

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

```

816 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
817 {
818   \exp_after:wN \cs_if_exist:NF \font@name
819   {

```

```

820     \group_begin:
821     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
822     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
823     \define@newfont
824     \group_end:
825   }
826 }
827 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

(End definition for \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。

```

828 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
829 {
830   \get@external@font
831   \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
832   {
833     \tl_set:Nx \external@font
834     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
835   }
836   \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。

```

837   \font@name
838   \lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
839   \use:c { \f@encoding + \f@family }
840   \use:c { \curr@fontshape }
841 }

```

(End definition for \ctex_ltj_extract_font:.)

\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。

```

842 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
843 {
844   \ctex_ltj_if_alter_nate_shape_exist:nF { \curr@fontshape }
845   {
846     \group_begin:
847     \tl_set_eq:NN \CJK@family \f@family
848     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
849     {
850       \cs_gset_protected_nopar:Npx \subst@correction
851       {
852         \cs_new_eq:NN
853         \exp_not:c { \l__ctex_ltj_current_font_tl }
854         \font@name
855       }
856       \group_insert_after:N \group_insert_after:N
857       \group_insert_after:N \subst@correction
858     }
859     \group_end:
860   }
861 }

```

(End definition for \ctex_ltj_subst_font:.)

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

```

862 \prg_new_conditional:Npnn \ctex_ltj_if_alter_nate_shape_exist:n #1 { T , F , TF }
863 {
864   \lua_now_x:n { luatexja.jfont.does_alt_set (' \lua_escape_x:n { #1 } ') }
865   \prg_return_true: \else: \prg_return_false: \fi:
866 }

```

(End definition for \ctex_ltj_if_alter_nate_shape_exist:nTF.)

__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。

```

867 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
868 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for __ctex_ltj_patch_external_font:w.)

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

```

869 \cs_new_protected_nopar:Npn \ctex_ltj_select_alter_nate_font:
870 {
871   \ctex_ltj_if_alter_nate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
872   {
873     \lua_now_x:n
874     {
875       luatexja.jfont.output_alt_font_cmd
876       ('y', ' \lua_escape_x:n { \l__ctex_ltj_current_shape_tl } ')
877     }
878     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size' ) }
879   }
880 }
881 \tl_new:N \l__ctex_ltj_current_shape_tl
882 \tl_set:Nn \l__ctex_ltj_current_shape_tl
883 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for `\ctex_ltj_select_alterate_font:`.)

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

```

884 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
885 {
886   \cs_if_exist:cF { #1/\f@size }
887   {
888     \group_begin:
889     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
890     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
891     \ctex_ltj_pickup_font:
892     \group_end:
893     \__ctex_ltj_pop_fontname:
894   }
895 }

```

(End definition for `\ltj@pickup@altfont@auxy`.)

896 `<@@=`

`\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。

```

897 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
898 {
899   \ltj@@getjfontnumber #1
900   \lua_now_x:n
901   {
902     luatexja.jfont.pickup_alt_font_b
903     ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
904   }
905 }

```

(End definition for `\ltj@pickup@altfont@copy`.)

14.4.3.4 数学字体族

906 `<@@=ctex_ltj>`

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

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

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

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

```

907 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
908 {
909   \lua_now_x:n
910   { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[~/]+' ) ) }
911   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
912 }

```

(End definition for `\ctex_ltj_if_jfont:nTF`.)

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

```

913 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
914 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
915 \group_begin:
916 \char_set_catcode_other:N M
917 \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
918 { \ctex_ltj_if_jfont:nTF {#3} }
919 \group_end:

```


(End definition for \ctex_ljt_if_jfont_math:NTF.)

920 <@@=>

\getanddefine@fonts
\ctex_ljt_get_and_define_fonts:nN

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。

```

921 \cs_new_protected_nopar:Npn \ctex_ljt_get_and_define_fonts:nN #1#2
922 {
923   \ctex_ljt_if_jfont:NTF { \token_to_str:N #2 }
924   { \ctex_ljt_get_and_define_fonts_ja:nN }
925   { \ctex_ljt_get_and_define_fonts_al:nN }
926   {#1} #2
927 }
928 \cs_new_eq:NN \ctex_ljt_get_and_define_fonts_al:nN \getanddefine@fonts
929 \cs_set_eq:NN \getanddefine@fonts \ctex_ljt_get_and_define_fonts:nN
930 \cs_new_protected_nopar:Npn \ctex_ljt_get_and_define_fonts_ja:nN #1#2
931 {
932   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
933   \ctex_ljt_pickup_font: \tl_set_eq:NN \textfont@name \font@name
934   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
935   \ctex_ljt_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
936   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
937   \ctex_ljt_pickup_font:
938   \tl_put_right:Nx \math@fonts
939   {
940     \ltj@setpar@global
941     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
942     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
943     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
944   }
945 }

```

(End definition for \getanddefine@fonts and \ctex_ljt_get_and_define_fonts:nN.)

946 <@@=ctex_ljt>

\use@mathgroup
\ctex_ljt_use_math_group:Nn

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

```

947 \cs_new_protected_nopar:Npn \ctex_ljt_use_math_group:Nn #1#2
948 {
949   \mode_if_math:T
950   {
951     \math@bgroup
952     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
953     \ctex_ljt_math_group_hook:
954     \ctex_ljt_if_jfont_math:NTF #1
955     { \jfam } { \mathgroup } #2 \scan_stop:
956     \math@egroup
957   }
958 }
959 \cs_new_eq:NN \ctex_ljt_math_group_hook: \prg_do_nothing:
960 \cs_set_eq:NN \use@mathgroup \ctex_ljt_use_math_group:Nn

```

(End definition for \use@mathgroup and \ctex_ljt_use_math_group:Nn.)

14.4.3.5 字体族的定义与使用

\ctex_mono_jfm:n
\l__ctex_ljt_jfm_tl

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

```

961 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
962 {
963   \str_if_eq:nnTF {#1} { plain }
964   { \tl_set:Nn \l__ctex_ljt_jfm_tl { mono } }
965   { \tl_set:Nn \l__ctex_ljt_jfm_tl {#1} }
966 }

```

```

967 \tl_new:N \l__ctex_ltj_jfm_tl
968 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
969 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

(End definition for \ctex_mono_jfm:n and \l__ctex_ltj_jfm_tl.)

\CJK@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 \ctex_ltj_if_
 $\text{_ctex_ltj_change_encoding:}$ \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。

```

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

```

(End definition for \CJK@encoding and __ctex_ltj_change_encoding:.)

这是 luatexja-fontspec 中新增的一些字体选项。

```

992 \newfontfeature { CID } { cid = #1 }
993 \newfontfeature { JFM } { jfm = #1 }
994 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

995 \keys_define:nn { fontspec-prepare-external }
996 {
997   NoEmbed .code:n =
998     { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
999 }
1000 \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。

```

1001 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1002 {
1003   \group_begin:
1004   \clist_clear:N \l__ctex_ltj_char_range_clist
1005   \seq_clear:N \l__ctex_ltj_alternate_seq
1006   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1007   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1008   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1009   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1010   {
1011     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1012     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1013       {#1} { \l__ctex_ltj_font_options_clist }
1014     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist

```

```

1015         \l__ctex_ltj_font_options_clist
1016         \l__ctex_ltj_gset_family_cs:nn {#1} {#3}
1017     }
1018     \group_end:
1019 }
1020 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1021 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1022 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1023 \clist_new:N \l__ctex_ltj_font_options_clist
1024 \cs_new_protected_nopar:Npn \l__ctex_ltj_use_global_options:N #1
1025 {
1026     \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1027     \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1028 }

```

(End definition for \ctex_ltj_set_family:nnn.)

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

```

1029 \prop_new:N \g__ctex_ltj_family_name_prop
1030 \prop_new:N \g__ctex_ltj_family_font_name_prop
1031 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

(End definition for \g__ctex_ltj_family_name_prop, \g__ctex_ltj_family_font_name_prop, and \g__ctex_ltj_family_font_options_prop.)

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

```

1032 \cs_new_protected_nopar:Npn \l__ctex_ltj_check_family:n #1
1033 {
1034     \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1035     {
1036         \cs_undefine:c { \l__ctex_ltj_family_csname:n {#1} }
1037         \cs_undefine:c { \l__ctex_ltj_alternate_cs:n {#1} }
1038         \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1039         {
1040             \use:c { \l__ctex_ltj_alternate_cs:n { clear / #1 } }
1041             \cs_undefine:c { \l__ctex_ltj_alternate_cs:n { clear / #1 } }
1042             \cs_undefine:c { \l__ctex_ltj_alternate_cs:n { reset / #1 } }
1043             \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1044         }
1045         \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1046     }
1047 }
1048 \tl_new:N \l__ctex_ltj_tmp_tl
1049 \msg_new:nnn { ctex } { redefine-family }
1050 { Redefining~CJKfamily~\l__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

(End definition for \l__ctex_ltj_check_family:n.)

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

```

1051 \cs_new_protected_nopar:Npn \l__ctex_ltj_gset_family_cs:nn #1#2
1052 {
1053     \cs_gset_protected_nopar:cpx { \l__ctex_ltj_family_csname:n {#1} }
1054     {
1055         \group_begin:
1056         \l__ctex_ltj_change_encoding:
1057         \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1058         \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1059         { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1060         \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1061         { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1062         \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1063         \exp_not:N \g__ctex_ltj_fontspec_family_tl

```

```

1064     \__ctex_ltj_set_altername_family:n {#1}
1065     \group_end:
1066   }
1067 }
1068 \tl_new:N \l__ctex_ltj_base_family_tl
1069 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1070 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1071 \cs_new_protected_nopar:Npn \__ctex_ltj_set_altername_family:n #1
1072 {
1073   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1074   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1075   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1076   \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n {#1} }
1077 }
1078 \cs_new:Npn \__ctex_ltj_altername_cs:n #1 { ctex_ltj/altername_family/#1 }

```

(End definition for __ctex_ltj_gset_family_cs:nn.)

\CJKfamily 切换字体。

```

1079 \NewDocumentCommand \CJKfamily { m }
1080 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1081 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1082 {
1083   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1084   {
1085     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1086     \selectfont
1087   }
1088   { \__ctex_ltj_family_unknown_warning:n {#1} }
1089 }
1090 \tl_new:N \l_ctex_ltj_family_tl
1091 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

(End definition for \CJKfamily.)

\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。

```

1092 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1093 {
1094   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1095   { \prg_return_true: }
1096   {
1097     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1098     {
1099       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1100       \prg_return_true:
1101     }
1102     { \prg_return_false: }
1103   }
1104 }
1105 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

__ctex_ltj_family_unknown_warning:n

```

1106 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1107 {
1108   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1109   {
1110     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1111     {
1112       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1113       \msg_warning:nnn { ctex } { family-unknown } {#1}
1114     }

```

```

1115     }
1116 }
1117 \seq_new:N \g__ctex_ltj_unknown_family_seq
1118 \msg_new:nnn { ctex } { family-unknown }
1119 {
1120     Unknown~CJK~family~`__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1121     Try~to~use~`__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1122 }
1123 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1124 {
1125     \str_case_x:nnF {#1}
1126     {
1127         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1128         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1129         \CJKttdefault { \token_to_str:N \setCJKmonofont }
1130     }
1131     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1132     [...] \{...\}
1133 }
1134 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1135 {
1136     \str_case_x:nnF {#1}
1137     {
1138         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1139         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1140         \CJKttdefault { \token_to_str:N \CJKttdefault }
1141     }
1142     {#1}
1143 }

```

(End definition for `__ctex_ltj_family_unknown_warning:n`.)

`\ctex_ltj_fontspec:nn`

```

1144 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1145 {
1146     \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1147     { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1148     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1149     {
1150         \int_gincr:N \g__ctex_ltj_family_int
1151         \__ctex_ltj_fontspec:xnn
1152         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1153         {#1} {#2}
1154     }
1155 }
1156 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1157 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1158 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1159 {
1160     \bool_if:NT \l__ctex_ltj_add_alternate_bool
1161     {
1162         \cs_if_free:cF
1163         { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1164         {
1165             \cs_gset_eq:cc
1166             { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1167             { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1168             \cs_gset_eq:cc
1169             { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1170             { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1171         }
1172         \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1173     }
1174     \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1175     \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1176     \ctex_ltj_switch_family:n {#1}

```

```

1177 }
1178 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1179 \prop_new:N \g__ctex_ltj_fontspec_prop

(End definition for \ctex_ltj_fontspec:nn.)

\ctex_ltj_add_font_features:n
\ctex_ltj_add_font_features:nn
1180 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1181 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1182 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1183 {
1184   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1185     {#1} \l__ctex_ltj_tmp_tl
1186   {
1187     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1188       {#1} \l__ctex_ltj_font_options_clist
1189     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1190     \bool_set_true:N \l__ctex_ltj_add_alternate_bool
1191     \ctex_ltj_fontspec:xx
1192       { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1193       { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1194   }
1195   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1196 }
1197 \bool_new:N \l__ctex_ltj_add_alternate_bool
1198 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1199 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1200 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1201 {
1202   \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1203   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1204 }

```

(End definition for \ctex_ltj_add_font_features:n and \ctex_ltj_add_font_features:nn.)

```

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

```

(End definition for \setCJKfamilyfont and others.)

```

\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1226 \NewDocumentCommand \setCJKmainfont { O { } m }
1227 {

```

```

1228 \ctex_ljtj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1229 \normalfont
1230 }
1231 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1232 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1233 {
1234 \ctex_ljtj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1235 \normalfont
1236 }
1237 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1238 {
1239 \ctex_ljtj_set_family:xxx { \CJKttdefault } {#1} {#2}
1240 \normalfont
1241 }
1242 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1243 { \ctex_ljtj_set_family:xxx { \c__ctex_ljtj_math_tl } {#1} {#2} }
1244 \NewDocumentCommand \defaultCJKfontfeatures { m }
1245 { \clist_gset:Nn \g__ctex_ljtj_default_features_clist {#1} }
1246 \clist_new:N \g__ctex_ljtj_default_features_clist
1247 \@onlypreamble \setCJKmainfont
1248 \@onlypreamble \setCJKsansfont
1249 \@onlypreamble \setCJKmonofont
1250 \@onlypreamble \setCJKmathfont
1251 \@onlypreamble \setCJKromanfont
1252 \@onlypreamble \defaultCJKfontfeatures

(End definition for \setCJKmainfont and others.)

1253 \tl_if_exist:NF \CJKfamilydefault
1254 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1255 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1256 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1257 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1258 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1259 { \CJKfamily { \CJKrmdefault } }
1260 { }
1261 { \ctex_patch_failure:N \rmfamily }
1262 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1263 { \CJKfamily { \CJKsfdefault } }
1264 { }
1265 { \ctex_patch_failure:N \sffamily }
1266 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1267 { \CJKfamily { \CJKttdefault } }
1268 { }
1269 { \ctex_patch_failure:N \ttfamily }
1270 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1271 { \CJKfamily { \CJKfamilydefault } }
1272 { \cs_set_eq:NN \reset@font \normalfont }
1273 { \ctex_patch_failure:N \normalfont }

```

在导言区末尾更新 \CJKfamilydefault。

```

1274 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

\ctex_ljtj_ensure_default_family: 在导言区结束确认 \CJKfamilydefault 确实存在。

```

1275 \cs_new_protected_nopar:Npn \ctex_ljtj_ensure_default_family:
1276 {
1277 \prop_if_empty:NF \g__ctex_ljtj_family_font_name_prop
1278 {
1279 \ctex_ljtj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ljtj_tmpa_tl
1280 {
1281 \str_if_eq:x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1282 { \use:n }
1283 {
1284 \ctex_ljtj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ljtj_tmpa_tl
1285 { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }

```

```

1286         { \use:n }
1287     }
1288     {
1289         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1290         {
1291             \prop_map_break:n
1292             { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1293         }
1294     }
1295 }
1296 \normalfont
1297 \ctex_ltj_update_mathfont:
1298 }
1299 }

```

(End definition for \ctex_ltj_ensure_default_family:.)

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

```

1300 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1301 {
1302     \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1303     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1304     {
1305         \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1306         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1307     }
1308 }
1309 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1310 {
1311     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1312     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1313     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1314     \cs_if_free:cTF
1315     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1316     {
1317         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1318         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1319     }
1320     {
1321         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1322         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1323     }
1324 }

```

(End definition for \ctex_ltj_update_mathfont:.)

14.4.3.6 替代字体的设置

AlternateFont 设置替代字体的选项。

CharRange

```

1325 \keys_define:nn { ctex_ltj / fontspec }
1326 {
1327     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1328     AlternateFont .value_required:n = true ,
1329     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1330     CharRange .value_required:n = true
1331 }

```

(End definition for AlternateFont and CharRange. These functions are documented on page 25.)

\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。

```

1332 \group_begin:

```



```

1333 \char_set_catcode_other:N \l
1334 \cs_set:Npn \__ctex_ltj_tmp:w #1
1335 {
1336   \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1337   {
1338     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1339     {
1340       \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1341       \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1342       \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1343       \seq_set_filter:Nn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1344       { ! \tl_if_blank_p:n { ##### } }
1345       \seq_concat:NNN \l__ctex_ltj_alternate_seq
1346       \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1347     }
1348   }
1349 }
1350 \char_set_catcode_active:N \l
1351 \__ctex_ltj_tmp:w { || }
1352 \group_end:
1353 \seq_new:N \l__ctex_ltj_tmp_seq
1354 \seq_new:N \l__ctex_ltj_alternate_seq

```

(End definition for \ctex_ltj_set_alternate_seq:n)

\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。

```

1355 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1356 {
1357   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1358   {
1359     \__ctex_ltj_check_family:n {#1}
1360     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1361     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1362     #3
1363   }
1364   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1365 }

```

(End definition for \ctex_ltj_set_alternate_family:nnF)

\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
1366 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1367 {
1368   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1369   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1370 }
1371 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1372 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1373 { m m m +O{ } m u{ \q_stop } }
1374 {
1375   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1376   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1377   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1378   \tl_if_blank:nTF {#5}
1379   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1380   {
1381     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1382     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1383   }
1384   \use:x
1385   {
1386     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1387     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1388     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1389     { \exp_not:o { \l__ctex_ltj_tmp_tl } }

```

```

1390     }
1391 }
1392 \clist_new:N \l__ctex_ltj_alternate_options_clist

(End definition for \ctex_ltj_save_alternate_seq:Nn and \ctex_ltj_save_alternate_seq:Nnnwn.)

```

`\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。

```

1393 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1394 {
1395   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1396   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1397   \ctex_ltj_set_alternate_family:coonn
1398   { \__ctex_ltj_alternate_cs:n {#1} }
1399   { \l__ctex_ltj_char_range_clist }
1400   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1401 }
1402 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1403 {
1404   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1405   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1406   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1407 }
1408 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

(End definition for \ctex_ltj_set_alternate_family:nn.)

```

`\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。

```

1409 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnnn #1#2#3#4
1410 {
1411   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1412   \cs_gset_protected_nopar:Npx #1
1413   { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1414 }

(End definition for \ctex_ltj_save_alternate_family:Nnnnn.)

```

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

```

1415 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1416 {
1417   \group_begin:
1418   \__ctex_ltj_change_encoding:
1419   \cs_set_eq:NN \CJKfamily \use_none:n
1420   \ctex_ltj_swap_cs:NN
1421   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1422   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1423   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1424   \group_end:
1425 }
1426 \tl_new:N \l__ctex_ltj_alternate_family_tl

(End definition for \ctex_ltj_set_alternate_family:nnn.)

```

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

```

1427 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1428 {
1429   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1430   \cs_set_eq:NN #1 #2
1431   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1432   \cs_undefine:N \__ctex_ltj_tmp:w
1433 }

```

(End definition for \ctex_ltj_swap_cs:NN.)

LTJFONTUID
_ctex_ltj_update_family_uid:N

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

```
1434 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1435 \cs_new_protected_nopar:Npn \_ctex_ltj_update_family_uid:N #1
1436 {
1437   \int_gincr:N \g__ctex_ltj_family_int
1438   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1439 }
1440 \int_new:N \g__ctex_ltj_family_int
```

(End definition for LTJFONTUID and _ctex_ltj_update_family_uid:N.)

\ctex_ltj_declare_alter_nate_shape:nnnnnn

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

```
1441 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1442 {
1443   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1444   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1445   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1446   { \l_fontspec_family_tl } {#3} {#4}
1447 }
```

(End definition for \ctex_ltj_declare_alter_nate_shape:nnnnnn.)

\ctex_ltj_set_alter_nate_shape:Nnnnnnn

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

```
1448 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1449 {
1450   \clist_map_inline:Nn #1
1451   {
1452     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1453     {
1454       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1455       \l__ctex_ltj_char_range_tl
1456     }
1457     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1458   }
1459   \_ctex_ltj_save_alter_nate_shape:cnn
1460   { \_ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1461   { luatexja.jfont.clear_alt_font_latex }
1462   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1463 }
```

(End definition for \ctex_ltj_set_alter_nate_shape:Nnnnnnn.)

\ctex_ltj_set_alter_nate_shape:nnn

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

```
1464 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1465 { m m > { \SplitArgument { \c_one } { -> } } m }
1466 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1467 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1468 {
1469   \ctex_ltj_set_alter_nate_shape:n
1470   {
1471     \IfNoValueTF {#4}
1472     { \int_eval:n {#3} , \int_eval:n {#3} , }
```

```

1473         {
1474             \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1475             \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1476         }
1477         '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1478         '\lua_escape_x:n { \CJK@encoding/#1 }'
1479     }
1480 }
1481 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1482 {
1483     \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1484     \__ctex_ltj_save_alternate_shape:cnn
1485     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1486     { luatexja.jfont.set_alt_font_latex } {#1}
1487 }

```

(End definition for \ctex_ltj_set_alternate_shape:nnn.)

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

```

1488 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1489 {
1490     \tl_map_inline:Nn #3
1491     {
1492         \ctex_ltj_set_alternate_shape:n
1493         {
1494             ##1 ,
1495             '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1496             '\lua_escape_x:n { \CJK@encoding/#1 }'
1497         }
1498     }
1499 }

```

(End definition for \ctex_ltj_set_alternate_shape:nnN.)

__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。

```

1500 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1501 {
1502     \group_begin:
1503     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1504     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1505     \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1506     \cs_gset_protected_nopar:Npx #1
1507     { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1508     \group_end:
1509 }
1510 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nnn { c }

```

(End definition for __ctex_ltj_save_alternate_shape:Nnn.)

clearalternatefont 清除和重置操作总是全局的。

resetalternatefont

```

1511 \keys_define:nn { ctex }
1512 {
1513     clearalternatefont .code:n =
1514     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1515     resetalternatefont .code:n =
1516     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1517     clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1518     resetalternatefont .default:n = \l_ctex_ltj_family_tl
1519 }
1520 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1521 {

```

```

1522 \group_begin:
1523 \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1524 {
1525 \cs_if_exist_use:cT { \l__ctex_ltj_alterate_cs:n { clear / #1 } }
1526 {
1527 \prop_gput:Nno \g__ctex_ltj_reset_alterate_prop
1528 {#1} { \l__ctex_ltj_base_family_tl }
1529 \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1530 \selectfont
1531 }
1532 }
1533 { \l__ctex_ltj_family_unknown_warning:n {#1} }
1534 \group_end:
1535 }
1536 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alterate_font:n #1
1537 {
1538 \group_begin:
1539 \prop_gpop:NnNT \g__ctex_ltj_reset_alterate_prop {#1} \CJK@family
1540 {
1541 \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1542 \use:c { \l__ctex_ltj_alterate_cs:n { reset / #1 } }
1543 \selectfont
1544 }
1545 \group_end:
1546 }
1547 \prop_new:N \g__ctex_ltj_reset_alterate_prop
1548 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalteratefont` and `resetalteratefont`. These functions are documented on page 26.)

declarecharrange 预先声明字符范围。

```

1549 \keys_define:nn { ctex }
1550 {
1551 declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1552 declarecharrange .value_required:n = true
1553 }
1554 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1555 { \clist_map_inline:nn {#1} { \l__ctex_ltj_declare_char_range:nn ##1 } }
1556 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1557 \cs_new_protected_nopar:Npn \l__ctex_ltj_declare_char_range:nn #1#2
1558 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

(End definition for `declarecharrange`. This function is documented on page 25.)

```

\ctex_ltj_declare_char_range:nn #1 是名字,#2 是范围。
\g__ctex_ltj_char_range_prop
1559 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1560 {
1561 \tl_clear:N \l__ctex_ltj_char_range_tl
1562 \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1563 \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1564 \ctex_ltj_def_char_range_key:n {#1}
1565 \tl_clear:N \l__ctex_ltj_char_range_tl
1566 }
1567 \tl_new:N \l__ctex_ltj_char_range_tl
1568 \prop_new:N \g__ctex_ltj_char_range_prop

```

(End definition for `\ctex_ltj_declare_char_range:nn` and `\g__ctex_ltj_char_range_prop`.)

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

```

1569 \NewDocumentCommand \ctex_ltj_save_char_range:n
1570 { > { \SplitArgument { \c_one } { -> } } m }
1571 { \ctex_ltj_save_char_range:nn #1 }
1572 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2

```

```

1573 {
1574   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1575   { {
1576     \IfNoValueTF {#2}
1577     { \int_eval:n {#1} , \int_eval:n {#1} }
1578     {
1579       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1580       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1581     }
1582   } }
1583 }

```

(End definition for \ctex_ltj_save_char_range:n.)

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

```

1584 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1585 {
1586   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1587   {
1588     \keys_define:nn { ctex_ltj / fontspec }
1589     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1590   }
1591 }

```

(End definition for \ctex_ltj_def_char_range_key:n.)

\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。

```

1592 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1593 {
1594   \tl_if_blank:nTF {#2}
1595   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1596   {
1597     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1598     {
1599       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1600       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1601     }
1602   }
1603 }

```

(End definition for \ctex_ltj_char_range_key:nn.)

__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]。

```

1604 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1605 { +o o u { \q_stop } }
1606 {
1607   \exp_args:NNf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1608   {
1609     \IfNoValueTF {#1} { {#3} }
1610     {
1611       \IfNoValueTF {#2}
1612       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] { #3 } } }
1613       { [ { #1 } ] { [ #2 ] } }
1614     }
1615   }
1616   \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1617 }

```

(End definition for __ctex_ltj_char_range_parse_feature:w.)

14.4.3.7 其它设置

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

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

1618 \AtBeginDocument
1619 {
1620   \ctex_appto_cmd:NnnTF \verbatim@font
1621   { \char_set_catcode_letter:n { 64 } }
1622   { \CTEX@verbatim@font@hook }
1623   { }
1624   { \ctex_patch_failure:N \verbatim@font }
1625 }
1626 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1627 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1628 <@@@=>

@@@italiccorr LATEX 的倾斜校正也要重新定义。
1629 \cs_set_eq:NN \@@italiccorr \/

(End definition for \@@italiccorr.)

1630 <@@@=ctex>
1631 </luatex>
```

14.4.4 ctex-engine-uptex.def

```
1632 <*uptex|aptex>

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。
1633 \tl_set:Nn \CJKrmdefault { rm }
1634 \tl_set:Nn \CJKsfdefault { sf }
1635 \tl_set:Nn \CJKttdefault { tt }
1636 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1637 \tl_put_left:Nn \rmfamily { \kanjifamily { \CJKrmdefault } }
1638 \tl_put_left:Nn \sffamily { \kanjifamily { \CJKsfdefault } }
1639 \tl_put_left:Nn \ttfamily { \kanjifamily { \CJKttdefault } }
1640 \NewDocumentCommand \CJKfamily { m }
1641 {
1642   \kanjifamily { #1 }
1643   \selectfont
1644 }
1645 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }
```

在导言区末尾更新 `\CJKfamilydefault`。

```
1646 \ctex_at_end_preamble:n { \ctex_update_default_family: }
```

`\em` 取消 up_LAT_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。

```

1647 \ctex_patch_cmd:Nnn \em
1648 { \mcfamily \upshape \else \gtfamily \itshape }
1649 { \eminnershape \else \itshape }

```

(End definition for \em. This function is documented on page ??.)

`\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。

```

1650 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1651 {
1652   \DeclareKanjiFamily{JY2}{#1}{}
1653   \DeclareKanjiFamily{JT2}{#1}{}
1654   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1655   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1656   \tl_if_empty:nF { #3 }
1657   {
1658     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1659     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1660   }
1661 }
1662 \@onlypreamble \ctex_set_upfamily:nnn

```

(End definition for \ctex_set_upfamily:nnn.)

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

```

1663 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1664 {
1665   \ctex_set_zhmap:n
1666   {
1667     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1668     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1669     \tl_if_empty:nF { #3 }
1670     {
1671       \special{ pdf:mapline~ #1b-h~ UniGB-UTF16-H~ #3 }
1672       \special{ pdf:mapline~ #1b-v~ UniGB-UTF16-V~ #3 }
1673     }
1674   }
1675 }
1676 \@onlypreamble \ctex_set_upmap:nnn

```

(End definition for \ctex_set_upmap:nnn.)

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

```

1677 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1678 {
1679   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1680   \ctex_set_upmap:nnn { upserifit } { #3 } {}
1681   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1682   \ctex_set_upmap:nnn { upmono } { #6 } {}
1683 }
1684 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

(End definition for \ctex_set_upfonts:nnnnnn.)

`\@EverySelectfont@Init` every sel 宏包 (2011/10/28) 未考虑 up \LaTeX 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。

```

1685 \cs_set_nopar:Npn \@EverySelectfont@Init

```



```

1686 {
1687   \ctex_patch_cmd:Nnn \selectfont
1688   { \enc@update }
1689   {
1690     \@EverySelectfont@EveryHook
1691     \@EverySelectfont@AtNextHook
1692     \gdef \@EverySelectfont@AtNextHook { }
1693     \enc@update
1694   }
1695   \@ifpackageloaded { tracefnt }
1696   {
1697     \EverySelectfont
1698     {
1699       \ifnum \tracingfonts > \tw@
1700         \@font@info { Switching~ to~ \font@name }
1701         \fi
1702       }
1703     }
1704     { }
1705   \let \@EverySelectfont@Init \undefined
1706 }

```

(End definition for \@EverySelectfont@Init.)

1707 $\langle /uptex|aptex \rangle$

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。

```

1708 <*pdf|luatex|uptex|aptex>
1709 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1710 {
1711     \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1712     {
1713         \group_begin:
1714         \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1715         \tl_gset:Nx \CJKfamilydefault
1716         {
1717             \str_case:onF { \familydefault }
1718             {
1719                 { \rmdefault } { \exp_not:N \CJKrmdefault }
1720                 { \sfdefault } { \exp_not:N \CJKsfdefault }
1721                 { \ttdefault } { \exp_not:N \CJKttdefault }
1722             }
1723             { \CJKfamilydefault }
1724         }
1725     }
1726     \group_end:
1727 }

```

使用 `Lua2TeX` 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 `up3TeX` 时, `C19rm`、`JY2rm` 等总是有定义的, 不需要确认。

```

1727 <*/latex>
1728     \ctex_ljt_ensure_default_family:
1729 </latex>
1730 }

```

(End definition for \ctex_update_default_family:.)

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

```

1731 \tl_new:N \l__ctex_family_default_init_tl
1732 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n

```

```

1733 \tl_set:Nx \l__ctex_family_default_init_tl
1734 {
1735   \exp_not:N \__ctex_family_default_wrap:n
1736   { \exp_not:o { \CJKfamilydefault } }
1737 }
1738 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1739 </pdf|luatex|uptex|aptex>

(End definition for \l__ctex_family_default_init_tl.)

```

14.4.6 操作系统的判断

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

```

1740 <*luatex>
1741 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1742 {
1743   \tl_gset:Nx \g__ctex_fontset_tl
1744   {
1745     \lua_now_x:n
1746     {
1747       if ~ os.name == 'windows' then ~
1748         tex.sprint ( 'windows' )
1749       elseif ~ os.name == 'macosx' then ~
1750         tex.sprint ( 'mac' )
1751       else ~
1752         tex.sprint ( 'fandol' )
1753       end
1754     }
1755   }
1756 }
1757 </luatex>

```

ApTeX 可以使用 \ngostype 来判断。

```

1758 <*aptex>
1759 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1760 {
1761   \tl_gset:Nx \g__ctex_fontset_tl
1762   {
1763     \str_case:onF { \ngostype }
1764     {
1765       { Win32 } { windows }
1766       { Win64 } { windows }
1767       { Darwin } { mac }
1768     }
1769     { fandol }
1770   }
1771 }
1772 </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!

```

1773 <*xetex|pdf|uptex>
1774 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1775 {
1776   \file_if_exist:nTF { /dev/null }
1777   {
1778     \file_if_exist:nTF { nul: }
1779     {
1780       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1781       { \ctex_if_macosx:TF { mac } { windows } }
1782       { \ctex_if_macosx:TF { mac } { fandol } }
1783     }
1784   }
1785 }

```

```

1784         { \ctex_if_macosx:TF { mac } { fandol } }
1785     }
1786     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1787 }
1788 \str_const:Nx \c__ctex_upper_case_file_str
1789 { \exp_args:No \str_upper_case:n { \g_file_current_name_tl } }

```

(End definition for \ctex_detected_platform:.)

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

```

1790 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1791 {
1792     \file_if_exist:nTF { \c__ctex_macosx_file_str }
1793     { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1794     { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1795 }
1796 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1797 </xetex|pdfetex|uptex>

```

(End definition for \ctex_if_macosx:TF.)

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。

```

1798 <*class|ctex>
1799 \@ifpackageloaded { hyperref }
1800 {
1801     \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1802     { \hypersetup {#1} }
1803 }
1804 {
1805     \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1806     { \PassOptionsToPackage {#1} { hyperref } }
1807 }
1808 </class|ctex>

```

(End definition for \ctex_hypersetup:n.)

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。

```

1809 <*pdfetex>
1810 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1811 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1812 {
1813     \ctex_hypersetup:n { CJKbookmarks = true }
1814     \sys_if_output_pdf:TF
1815     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1816     {
1817         \ctex_at_end_package:nn { hyperref }
1818         {
1819             \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1820             {
1821                 \AtBeginShipoutFirst

```

```

1822             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1823         }
1824         { \RequirePackage { xCJK2uni } }
1825     }
1826 }
1827 }
1828 { \ctex_hypersetup:n { pdfencoding = unicode } }
1829 </pdfTeX>

```

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

```

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

```

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

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

$\text{X}\text{\LaTeX}$ 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 `CTEX` 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 $\text{X}\text{\LaTeX}$ 下更保险的 `pdfencoding` 选项来设置。为了与 $\text{X}\text{\LaTeX}$ 下的行为一致(使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 $\text{Lua}\text{\LaTeX}$ 下也启用这个选项。

```

1830 <*xetex|luatex>
1831 \ctex_hypersetup:n { pdfencoding = unicode }
1832 </xetex|luatex>

```

我们假定 $\text{up}\text{\LaTeX}$ 使用 DVIPDFMX 驱动输出, 于是使用与 $\text{pdf}\text{\LaTeX}$ 类似的设置。注意 $\text{up}\text{\LaTeX}$ 需要使用 UTF8-UTF16 的编码转换。

```

1833 <*uptex|aptex>
1834 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1835 \ctex_at_end_package:nn { hyperref }
1836 {
1837     \AtBeginShipoutFirst{ \special{ pdf:tounicode~ UTF8-UTF16 } }
1838 }
1839 </uptex|aptex>

1840 <*pdfTeX|xetex|luatex|uptex|aptex>

```

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

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

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

14.4.8 CJKfntef、xeCJKfntef 相关设置

CT_EX 宏集对 pdf_TE_X 与 Xe_TE_X 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。

```

1841 <*pdfTEX>
1842 \RequirePackage { CJKfntef }
1843 \normalem
1844 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1845 { \tl_clear:c { CJK#1color } }
1846 </pdfTEX>
1847 <*xeTEX>
1848 \RequirePackage { xeCJKfntef }
1849 \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
1850 {
1851   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1852   { \xeCJKsetup { #1 / format = { } } }
1853 }
1854 {
1855   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1856   { \tl_clear:c { CJK#1color } }
1857 }
1858 </xeTEX>
1859 <*luatex|uptex|aptex>
1860 \msg_new:nnn { ctex } { fntef-not-available }
1861 <luatex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1862 <uptex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ upLaTeX. }
1863 <aptex> { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ ApLaTeX. }
1864 \msg_warning:nn { ctex } { fntef-not-available }
1865 </luatex|uptex|aptex>
1866 \clist_map_inline:nn
1867 { underdot , underline , underdblline , underwave , sout , xout }
1868 <*pdfTEX|xeTEX>
1869 {
1870   \__ctex_clear_fntef_color:n {#1}
1871   \cs_new_protected_nopar:cpx { CTEX#1 }
1872   {
1873     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1874     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1875     \exp_not:c { CJK#1 }
1876   }
1877 }
1878 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1879 {
1880   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1881   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1882   \CJKfilltwosides
1883 }
1884 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1885 </pdfTEX|xeTEX>
1886 <*luatex|uptex|aptex>
1887 { \cs_new_eq:cN { CTEX#1 } \use:n }
1888 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1889 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1890 </luatex|uptex|aptex>
1891 <*pdfTEX>
1892 \clist_map_inline:nn
1893 {
1894   underdotbasesep , underdotsep , underlinebasesep ,
1895   underlinesep , underdbllinesep , underdbllinebasesep ,
1896   underwavebasesep , underwavesep , southeight ,
1897   underdotcolor , underwavecolor , underlinecolor ,
1898   underdbllinecolor , soutcolor , xoutcolor

```

```

1899 }
1900 {
1901   \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1902   \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1903 }
1904 </pdfTeX>

```

14.4.9 \ccwd 的更新

\ctex_update_ccwd:

```

\ccwd
1905 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1906 <*pdfTeX|xetex>
1907 {
1908   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1909   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1910 }
1911 </pdfTeX|xetex>
1912 <*luatex>
1913 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1914 </luatex>
1915 <*uptex|aptex>
1916 { \skip_set:Nn \ccwd { 1zw + \ptex_kanjiskip:D } }
1917 </uptex|aptex>
1918 \dim_new:N \ccwd

```

(End definition for \ctex_update_ccwd: and \ccwd.)

\ctex_update_ccglue: 更新字间距。

```

1919 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1920 <*pdfTeX>
1921 {
1922   \cs_set_protected_nopar:Npn \CJKglue
1923   { \skip_horizontal:N \l__ctex_ccglue_skip }
1924 }
1925 </pdfTeX>
1926 <*xetex>
1927 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1928 </xetex>
1929 <*luatex>
1930 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1931 </luatex>
1932 <*uptex|aptex>
1933 { \skip_set_eq:NN \ptex_kanjiskip:D \l__ctex_ccglue_skip }
1934 </uptex|aptex>
1935 \skip_new:N \l__ctex_ccglue_skip

```

(End definition for \ctex_update_ccglue:.)

\ctex_if_ccglue_touched_p: 检查用户是否修改过汉字间距。

\ctex_if_ccglue_touched: TF

```

1936 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1937 {
1938 <*pdfTeX|xetex>
1939   \if_meaning:w \CJKglue \__ctex_ccglue:
1940   \prg_return_false: \else: \prg_return_true: \fi:
1941 </pdfTeX|xetex>
1942 <*luatex>
1943   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1944   { \prg_return_false: } { \prg_return_true: }
1945 </luatex>
1946 <*uptex|aptex>
1947   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ptex_kanjiskip:D }
1948   { \prg_return_false: } { \prg_return_true: }

```

```
1949 </uptex|aptex>
1950 }
```

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

```
1951 <*pdfTeX|xetex>
1952 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1953 </pdfTeX|xetex>
```

(End definition for \ctex_if_ccglue_touched:TF.)

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

```
1954 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1955 <pdfTeX|xetex> { \dim_set:Nn \ccwd { \f@size \p@ } }
1956 <luatex> { \dim_set:Nn \ccwd { \zw } }
1957 <uptex|aptex> { \dim_set:Nn \ccwd { 1zw } }
```

(End definition for \ctex_update_em_unit:.)

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 `LuaTeX-ja` 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。`everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。

```
1958 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1959 {
1960   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1961   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1962 }
1963 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1964 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1965 \ctex_preto_cmd:NnnTF \selectfont { \ExplSyntaxOff }
1966 { \CTEX@selectfont@hook }
1967 {
1968   \tl_put_left:Nn \@EverySelectfont@Init
1969   { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1970 }
1971 { \ctex_patch_failure:N \selectfont }
```

(End definition for \ctex_add_to_selectfont:n and \CTEX@selectfont@hook.)

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

```
1972 <*pdfTeX>
1973 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1974 </pdfTeX>
1975 <*xetex|luatex|uptex|aptex>
1976 \EverySelectfont { \CTEX@selectfont@hook }
1977 </xetex|luatex|uptex|aptex>
```

Attribute 寄存器 `\ltj@curjfnt` 的初始值是 `-1`, 必须把它设置为一个有效的 `font.id`, 否则编译时会直接退出。

```
1978 <*luatex>
```

```

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

```

`\ctex_update_xkanjiskip:` 和 `\l__ctex_xkanjiskip_skip` 在 \TeX 和 $\text{\LuaTeX-j}\text{\a}$ 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。

```

1987 <*luatex|uptex|aptex>
1988 \cs_new_protected_nopar:Npn \ctex_update_xkanjiskip:
1989 {
1990   \skip_if_eq:nnT
1991   <luatex> { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
1992   <uptex|aptex> { \ptex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
1993   {
1994     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
1995     <luatex> \ltjsetxkanjiskip \l__ctex_xkanjiskip_skip
1996     <uptex|aptex> \skip_set_eq:NN \ptex_xkanjiskip:D \l__ctex_xkanjiskip_skip
1997   }
1998 }
1999 \tl_new:N \l__ctex_xkanjiskip_tl
2000 \tl_set:Nn \l__ctex_xkanjiskip_tl
2001 <luatex> { .25\zw plus 1pt minus 1pt }
2002 <uptex|aptex> { .25\zw plus 1pt minus 1pt }
2003 \skip_new:N \l__ctex_xkanjiskip_skip
2004 \skip_set:Nn \l__ctex_xkanjiskip_skip
2005 <luatex> { \ltjgetparameter { xkanjiskip } }
2006 <uptex|aptex> { \ptex_xkanjiskip:D }

(End definition for \ctex_update_xkanjiskip: and \l__ctex_xkanjiskip_skip.)

2007 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2008 </luatex|uptex|aptex>

```

space 在导言区或正文中设置忽略空格方式。 $\text{\pdf}\text{\TeX}$ 和 $\text{\Xe}\text{\TeX}$ 下初始设置为 `auto`， $\text{\Lua}\text{\TeX}$ 、 $\text{\up}\text{\TeX}$ 下是无效选项。

```

2009 \keys_define:nn { ctex }
2010 {
2011   <*pdf|tex|xetex>
2012   space .choice: ,
2013   space / true .code:n =
2014   <pdf|tex> { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
2015   <xetex> { \xeCJKsetup { CJKspace = true } } ,
2016   space / auto .code:n =
2017   <pdf|tex> { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
2018   <xetex> { \xeCJKsetup { CJKspace = false } } ,
2019   space / false .code:n =
2020   <pdf|tex> { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
2021   <xetex> { \xeCJKsetup { CJKspace = false } } ,
2022   space .default:n = { true } ,
2023   space .initial:n = { auto }
2024   </pdf|tex|xetex>
2025   <*luatex|uptex|aptex>
2026   space .code:n =
2027   { \msg_warning:nn { ctex } { invalid-option } }
2028   </luatex|uptex|aptex>
2029 }

```

(End definition for space. This function is documented on page 28.)

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。upT_EX 暂时无效。

```

2030 \keys_define:nn { ctex }
2031 {
2032   punct .code:n =
2033   {
2034     \tl_set:Nx \l__ctex_punct_tl { #1 }
2035     <pdfTeX> \punctstyle { \l__ctex_punct_tl }
2036     <xetex> \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2037     <luatex> \ctex_mono_jfm:o { \l__ctex_punct_tl }
2038     <uptex|aptex> \msg_warning:nn { ctex } { invalid-option }
2039   } ,
2040   punct .default:n = { quanjiao } ,
2041 }

```

(End definition for punct. This function is documented on page 28.)

X_YL_AT_EX、LuaL_AT_EX 和 upL_AT_EX 总是使用 UTF8 编码。

```

2042 <*xetex|luatex|uptex|aptex>
2043 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2044 </xetex|luatex|uptex|aptex>
2045 </pdfTeX|xetex|luatex|uptex|aptex>
2046 <*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。

```

2047 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.5 字距与缩进

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

```

2048 \keys_define:nn { ctex }
2049 {
2050   autoindent .choice: ,
2051   autoindent .default:n = { true } ,
2052   autoindent / true .code:n =
2053   {
2054     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2055     \ctex_select_size:
2056   } ,
2057   autoindent / false .code:n =
2058   { \tl_clear:N \l__ctex_autoindent_tl } ,
2059   autoindent / unknown .code:n =
2060   {
2061     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2062     \ctex_select_size:
2063   }
2064 }

```

(End definition for autoindent. This function is documented on page 10.)

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2065 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2066 \cs_new_protected_nopar:Npn \ctex_select_size:
2067 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

(End definition for \CTEXsetfont. This function is documented on page 29.)

`\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。

```

2068 \cs_new_protected_nopar:Npn \ctex_update_size:
2069 {
2070   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2071   {
2072     \ctex_update_stretch:
2073     \ctex_update_parindent:
2074   }
2075   { \ctex_update_ziju: }
2076 }
2077 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2078 \tl_new:N \l__ctex_ziju_tl
2079 \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。

```

2080 \ctex_add_to_selectfont:n
2081 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

(End definition for `\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`。

```

2082 \keys_define:nn { ctex }
2083 {
2084   linestretch .code:n =
2085   {
2086     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2087     \ctex_select_size:
2088   } ,
2089   linestretch .value_required:n = true
2090 }
2091 \tl_new:N \l__ctex_line_stretch_tl
2092 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

(End definition for `linestretch`. This function is documented on page 10.)

`\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。 $\varepsilon\text{-TeX}$ 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。

```

2093 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2094 {
2095   \ctex_update_em_unit:
2096   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2097   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2098   {
2099     \ctex_if_ccglue_touched:TF
2100     { \ctex_update_ccwd: }
2101     {
2102       \dim_set:Nn \l__ctex_tmp_dim
2103       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2104       \skip_set:Nn \l__ctex_ccglue_skip
2105       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2106       \ctex_update_ccglue:
2107     }
2108   }
2109 }

```

```

2109 {
2110   \int_set:Nn \l__ctex_tmp_int
2111   { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2112   \tex_divide:D \l__ctex_tmp_int \ccwd
2113   \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2114   {
2115     \skip_set:Nn \l__ctex_ccglue_skip
2116     {
2117       \c_zero_dim plus \dim_eval:n
2118       {
2119         ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2120         \l__ctex_tmp_int
2121       }
2122     }
2123   }
2124   { \skip_zero:N \l__ctex_ccglue_skip }
2125   \ctex_update_ccglue:
2126 }
2127 }

```

(End definition for \ctex_update_stretch:.)

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

```

2128 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2129 {
2130   \tl_if_empty:NF \l__ctex_autoindent_tl
2131   {
2132     \dim_compare:nNnF \parindent = \c_zero_dim
2133     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2134   }
2135 }

```

(End definition for \ctex_update_parindent:.)

\ziju 若参数为 0, 则恢复正常间距。

```

2136 \NewDocumentCommand \ziju { m }
2137 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2138 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2139 {
2140   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2141   \ctex_select_size:
2142 }

```

(End definition for \ziju. This function is documented on page 24.)

\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。

```

2143 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2144 {
2145   \ctex_update_em_unit:
2146   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2147   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2148   \dim_compare:nNnTF \ccwd > \c_zero_dim

```

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

```

2149 {
2150   \dim_set:Nn \l__ctex_tmp_dim
2151   { \linewidth - \ccwd + \l__ctex_ziju_dim }
2152   \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2153   \tex_divide:D \l__ctex_tmp_int \ccwd
2154   \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。

```

2155     \dim_compare:nNnF \parindent = \c_zero_dim
2156     {
2157         \int_compare:nNnF \l__ctex_tmp_int < \c_three
2158         { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2159     }
2160     \skip_set:Nn \l__ctex_ccglue_skip
2161     {
2162         \l__ctex_ziju_dim
2163         plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2164         minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2165         { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2166     }
2167 }
2168 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2169 \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`。

```

2170     \ctex_update_parindent:
2171 }
2172 \dim_new:N \l__ctex_ziju_dim

```

(End definition for `\ctex_update_ziju:`.)

`\CTEXindent` 过时命令。
`\CTEXnoindent`

```

2173 \NewDocumentCommand \CTEXindent { }
2174 {
2175     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2176     { \parindent is~ set~ to~ 2\ccwd. }
2177     \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2178 }
2179 \NewDocumentCommand \CTEXnoindent { }
2180 {
2181     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2182     { \parindent is~ set~ to~ Opt. }
2183     \dim_zero:N \parindent
2184 }

```

(End definition for `\CTEXindent` and `\CTEXnoindent`. These functions are documented on page 29.)

14.6 中文数字与日期

```

2185 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2186 \RequirePackage { zhnumber }

```

`\chinese`

```

2187 \cs_new_nopar:Npn \chinese { \zhnum_counter:n }
2188 \cs_new_eq:NN \@chinese \@zhnum
2189 \cs_new_eq:NN \Chinese \chinese
2190 \cs_new_eq:NN \CTEXcounter \use_none:n

```

(End definition for `\chinese`. This function is documented on page 24.)

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

```

2191 \ctex_at_end_package:nn { enumitem }
2192 {
2193     \cs_if_free:NF \AddEnumerateCounter
2194     {

```

```

2195         \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2196         \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2197         \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2198     }
2199 }

```

`\CTEXnumber`

`\CTEXdigits`

```

2200 \NewDocumentCommand \CTEXnumber { m m }
2201 { \protected@edef #1 { \zhnumber {#2} } }
2202 \NewDocumentCommand \CTEXdigits { m m }
2203 { \protected@edef #1 { \zhdigits {#2} } }

```

(End definition for `\CTEXnumber` and `\CTEXdigits`. These functions are documented on page 24.)

`today`

```

2204 \cs_set_eq:NN \CTEX@todayold \today
2205 \keys_define:nn { ctex }
2206 {
2207     today .choice: ,
2208     today / old .code:n =
2209     { \cs_set_eq:NN \today \CTEX@todayold } ,
2210     today / small .code:n =
2211     {
2212         \cs_set_eq:NN \today \zhtoday
2213         \zhnumsetup { time = Arabic }
2214     } ,
2215     today / big .code:n =
2216     {
2217         \cs_set_eq:NN \today \zhtoday
2218         \zhnumsetup { time = Chinese }
2219     } ,
2220     today / unknown .code:n =
2221     { \msg_error:nxx { ctex } { today-undef } {#1} }
2222 }
2223 \msg_new:nnnn { ctex } { today-undef }
2224 { Today~format~`#1'~is~undefined. }
2225 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

(End definition for `today`. This function is documented on page 11.)

14.7 其它中文标题定义

```

2226 \keys_define:nn { ctex }
2227 {
2228     contentsname .tl_set:N = \contentsname ,
2229     listfigurename .tl_set:N = \listfigurename ,
2230     listtablename .tl_set:N = \listtablename ,
2231     figurename .tl_set:N = \figurename ,
2232     tablename .tl_set:N = \tablename ,
2233     abstractname .tl_set:N = \abstractname ,
2234     indexname .tl_set:N = \indexname ,
2235     appendixname .tl_set:N = \appendixname ,
2236     proofname .tl_set:N = \proofname ,
2237     <article> bibname .tl_set:N = \refname
2238     <book|report> bibname .tl_set:N = \bibname
2239     <*beamer>
2240     algorithmname .tl_set:N = \algorithmname ,
2241     bibname .tl_set:N = \bibname ,
2242     refname .tl_set:N = \refname ,
2243     continuation .tl_set:N = \insertcontinuationtext
2244 </beamer>
2245 }
2246 <*ctex>

```

```

2247 \msg_new:nnn { ctex } { ctexbibname }
2248 {
2249   Neither~~\token_to_str:N \bibname'~nor~~\token_to_str:N \refname'~can~be~found.\\
2250   The~key~~bibname'~will~set~~\token_to_str:N \ctexbibname'~to~the~given~value.
2251 }
2252 \tl_if_exist:NTF \insertcontinuationtext
2253 {
2254   \keys_define:nn { ctex }
2255   {
2256     algorithmname .tl_set:N = \algorithmname ,
2257     bibname       .tl_set:N = \bibname ,
2258     refname       .tl_set:N = \refname ,
2259     continuation .tl_set:N = \insertcontinuationtext
2260   }
2261 }
2262 {
2263   \tl_if_exist:NTF \bibname
2264   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2265   {
2266     \tl_if_exist:NTF \refname
2267     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2268     {
2269       \msg_warning:nn { ctex } { ctexbibname }
2270       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2271     }
2272   }
2273 }
2274 </ctex>
2275 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 CT_EX 文档类或打开 heading 选项下生效。

```
2276 <*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。

```

2277 <*article|book|report>
2278 \seq_new:N \c__ctex_section_headings_seq
2279 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2280 { section , subsection , subsubsection , paragraph , subparagraph }
2281 </article|book|report>

```

(End definition for \c__ctex_section_headings_seq.)

\c__ctex_headings_seq

```

2282 \seq_new:N \c__ctex_headings_seq
2283 <*article|book|report>
2284 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
2285 <book|report>\seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2286 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2287 </article|book|report>
2288 <*beamer>
2289 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2290 { part , section , subsection }
2291 </beamer>

```

(End definition for \c__ctex_headings_seq.)

__ctex_initial_heading:n

```

2292 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2293 {
2294   \tl_new:c { CTEX@pre#1 }
2295   \tl_new:c { CTEX@post#1 }
2296   \tl_const:cx { CTEX@the#1 }
2297   {
2298     \exp_not:c { CTEX@pre#1 }
2299     \exp_not:c { CTEX@the#1 }
2300     \exp_not:c { CTEX@post#1 }
2301   }
2302   \tl_const:cx { CTEX@#1name }
2303   {
2304     \group_begin:
2305     \exp_not:c { CTEX@#1@nameformat }
2306     {
2307       \exp_not:c { CTEX@pre#1 }
2308       \exp_not:N \tl_if_empty:NTF
2309       \exp_not:c { CTEX@#1@numberformat }
2310       { \exp_not:c { CTEX@the#1 } }
2311       {
2312         \group_begin:
2313         \exp_not:c { CTEX@#1@numberformat }
2314         \exp_not:c { CTEX@the#1 }
2315         \group_end:
2316       }
2317       \exp_not:c { CTEX@post#1 }
2318     }
2319   \group_end:
2320 }
2321 }

```

(End definition for __ctex_initial_heading:n)

__ctex_def_heading_keys:n

```

2322 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2323 {
2324   \tl_put_right:Nx \l__ctex_tmp_tl
2325   {
2326     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2327     #1 / name .code:n =
2328     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2329     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2330     #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2331     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2332     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2333     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2334     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2335     #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2336     #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
2337     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2338     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2339     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2340     < *article|book|report >
2341     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2342     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2343     < /article|book|report >
2344     #1 / format+ .code:n =
2345     { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2346     #1 / nameformat+ .code:n =
2347     { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2348     #1 / numberformat+ .code:n =
2349     { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2350     #1 / aftername+ .code:n =
2351     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2352     #1 / titleformat+ .code:n =
2353     { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,

```

```

2354      #1 / aftertitle+      .code:n =
2355      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { ####1 } } ,
2356      #1 / numbering      .initial:n = true ,
2357      #1 / beforeskip      .initial:n = \c_zero_skip ,
2358      #1 / afterskip      .initial:n = \c_zero_skip ,
2359      #1 / indent          .initial:n = \c_zero_dim ,
2360      #1 / beforeskip      .value_required:n = true ,
2361      #1 / afterskip      .value_required:n = true ,
2362      #1 / indent          .value_required:n = true ,
2363    }
2364  }

```

(End definition for _ctex_def_heading_keys:n.)

`ctex_assign_heading_name:nn` `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。

```

2365 \NewDocumentCommand \ctex_assign_heading_name:nn
2366 { m > { \SplitArgument { \c_one } { , } } +m }
2367 { \_ctex_assign_heading_name:nnn {#1} #2 }
2368 \cs_new_protected:Npn \_ctex_assign_heading_name:nnn #1#2#3
2369 {
2370   \tl_set:cn { CTEX@pre#1 } {#2}
2371   \IfNoValueTF {#3}
2372   { \tl_clear:c { CTEX@post#1 } }
2373   { \tl_set:cn { CTEX@post#1 } {#3} }
2374 }

```

(End definition for `\ctex_assign_heading_name:nn` and `_ctex_assign_heading_name:nnn`.)

```

part/pagestyle 只在 ctexbook 和 ctexrep 下有定义。
chapter/pagestyle
part/fixbeforeskip
chapter/fixbeforeskip
2375 \group_begin:
2376 <*book|report>
2377 \tl_set:Nn \l__ctex_tmp_tl
2378 {
2379   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2380   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2381   part / fixbeforeskip .bool_set:N = \CTEX@part@fixbeforeskip ,
2382   chapter / fixbeforeskip .bool_set:N = \CTEX@chapter@fixbeforeskip ,
2383 }
2384 </book|report>
2385 <*article|beamer>
2386 \tl_clear:N \l__ctex_tmp_tl
2387 </article|beamer>

```

(End definition for `part/pagestyle` and others. These functions are documented on page 21.)

定义标题键值选项。

```

2388 \seq_map_inline:Nn \c__ctex_headings_seq
2389 {
2390   \_ctex_initial_heading:n {#1}
2391   \_ctex_def_heading_keys:n {#1}
2392 }
2393 \use:x
2394 {
2395   \group_end:
2396   \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } }
2397 }
2398 <*article|book|report>

```


14.8.2 标准标题命令的修改

`\CTEX@makeheadinganchor` 用于 `numbering=false` 时, 设置超链接跳转描点。

```

2399 \cs_new_protected_nopar:Npn \CTEX@makeheadinganchor { }
2400 \ctex_at_end_package:nn { hyperref }
2401 { \cs_gset_protected_nopar:Npn \CTEX@makeheadinganchor { \phantomsection } }

(End definition for \CTEX@makeheadinganchor.)

```

`\CTEX@fixbeforeskip` 修正 **book** 和 **report** 类的 `\part` 和 `\chapter` 标题之前的多余空行。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 `beforeskip` 上作为修正。

```

2402 <*book|report>
2403 \cs_new_protected_nopar:Npn \CTEX@fixbeforeskip
2404 {
2405   \par
2406   \ifdim \pagegoal = \maxdimen
2407     \advance \@tempskipa -\topskip
2408     \advance \@tempskipa -\parskip
2409     \nointerlineskip
2410   \fi
2411 }
2412 </book|report>

(End definition for \CTEX@fixbeforeskip.)

```

14.8.2.1 part 的标题

```

2413 <@@=>

```

`\part`

```

2414 <*article>
2415 \renewcommand\part{%
2416   \if@noskipsec \leavevmode \fi
2417   \par
2418   % \addvspace{4ex}%
2419   \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
2420   \ifodd \CTEX@part@afterindent
2421     \@afterindenttrue
2422   \else
2423     \@afterindentfalse
2424   \fi
2425   \secdef\@part\@spart}
2426 </article>
2427 <*book|report>
2428 \renewcommand\part{%
2429   \if@openright
2430     \cleardoublepage
2431   \else
2432     \clearpage
2433   \fi
2434   % \thispagestyle{plain}%
2435   \thispagestyle{\CTEX@part@pagestyle}%
2436   \if@twocolumn
2437     \onecolumn
2438     \@tempwattrue
2439   \else
2440     \@tempwafalse
2441   \fi
2442   % \null\vfil
2443   \@tempskipa = \glueexpr \CTEX@part@beforeskip \relax

```

```

2444 \ifodd \CTEX@part@fixbeforeskip
2445 \CTEX@fixbeforeskip
2446 \fi
2447 \vspace*{\@tempskipa}%
2448 \secdef\@part\@spart}
2449 </book|report>

```

(End definition for \part.)

\@part

```

2450 <*article>
2451 \def\@part[#1]#2{%
2452 \ifnum \c@secnumdepth >\m@ne
2453 \ifodd \CTEX@part@numbering
2454 \refstepcounter{part}%
2455 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2456 \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2457 \else
2458 \CTEX@makeheadinganchor
2459 \addcontentsline{toc}{part}{#1}%
2460 \fi
2461 \else
2462 \addcontentsline{toc}{part}{#1}%
2463 \fi
2464 {\interlinepenalty \@M
2465 % \normalfont \parindent \z@ \raggedright
2466 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2467 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2468 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2469 \CTEX@partname \CTEX@part@aftername
2470 \fi \fi
2471 % \huge\bfseries #2%
2472 \CTEX@part@titleformat{#2}%
2473 \markboth{}{}%
2474 \CTEX@part@aftertitle}%
2475 \nobreak
2476 % \vskip 3ex
2477 \vskip \glueexpr \CTEX@part@afterskip \relax
2478 \@afterheading}
2479 </article>
2480 <*book|report>
2481 \def\@part[#1]#2{%
2482 \ifnum \c@secnumdepth >-2\relax
2483 \ifodd \CTEX@part@numbering
2484 \refstepcounter{part}%
2485 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2486 \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2487 \else
2488 \CTEX@makeheadinganchor
2489 \addcontentsline{toc}{part}{#1}%
2490 \fi
2491 \else
2492 \addcontentsline{toc}{part}{#1}%
2493 \fi
2494 \markboth{}{}%
2495 {\interlinepenalty \@M
2496 % \normalfont \centering
2497 \normalfont \CTEX@part@format
2498 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2499 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2500 \CTEX@partname \CTEX@part@aftername
2501 \fi \fi
2502 % \Huge\bfseries #2\par}%
2503 \CTEX@part@titleformat{#2}%
2504 \CTEX@part@aftertitle}%
2505 \@endpart}

```

```
2506 </book|report>
```

(End definition for \@part.)

\@spart

```
2507 <*article>
2508 \def\@spart#1{%
2509     {\interlinepenalty \@M
2510      \normalfont \parindent \z@ \raggedright
2511      \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2512      \CTEX@part@format
2513      \huge \bfseries #1\par}%
2514      \CTEX@part@titleformat{#1}%
2515      \CTEX@part@aftertitle}%
2516      \nobreak
2517      \vskip 3ex
2518      \vskip \glueexpr \CTEX@part@afterskip \relax
2519      \@afterheading}
2520 </article>
2521 <*book|report>
2522 \def\@spart#1{%
2523     {\interlinepenalty \@M
2524      \normalfont \centering
2525      \normalfont \CTEX@part@format
2526      \Huge \bfseries #1\par}%
2527      \CTEX@part@titleformat{#1}%
2528      \CTEX@part@aftertitle}%
2529      \@endpart}
2530 </book|report>
```

(End definition for \@spart.)

\@endpart

```
2531 <*book|report>
2532 \def\@endpart{%
2533     %
2534     \vfil
2535     \vskip \glueexpr \CTEX@part@afterskip \relax
2536     \newpage
2537     \if@twoside
2538     \if@openright
2539     \null
2540     \thispagestyle{empty}%
2541     \newpage
2542     \fi
2543     \fi
2544     \if@tempwa
2545     \twocolumn
2546     \fi}
2547 </book|report>
```

(End definition for \@endpart.)

14.8.2.2 chapter 的标题

```
2547 <*book|report>
```

\chapter

```
2548 \renewcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
2549 %
2550     \thispagestyle{plain}%
2551     \thispagestyle{\CTEX@chapter@pagestyle}%
2552     \global\@topnum\z@
2553     \afterindentfalse
2554     \ifodd \CTEX@chapter@afterindent
```

```

2554             \@afterindenttrue
2555         \else
2556             \@afterindentfalse
2557         \fi
2558     \secdef\@chapter\@schapter}

```

(End definition for \chapter.)

\@chapter

```

2559 \def\@chapter[#1]#2{%
2560     \ifnum \c@secnumdepth >\m@ne
2561     <book>     \if@mainmatter
2562         \ifodd \CTEX@chapter@numbering
2563             \refstepcounter{chapter}%
2564             % \typeout{\@chapapp\space\thechapter.}%
2565             \typeout{\CTEXthechapter}%
2566             \addcontentsline{toc}{chapter}
2567             % {\protect\numberline{\thechapter}#1}%
2568             {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2569         \else
2570             \CTEX@makeheadinganchor
2571             \addcontentsline{toc}{chapter}{#1}%
2572         \fi
2573     <book>     \else
2574     <book>     \addcontentsline{toc}{chapter}{#1}%
2575     <book>     \fi
2576     \else
2577         \addcontentsline{toc}{chapter}{#1}%
2578     \fi
2579     \chaptermark{#1}%
2580     \addtocontents{lof}{\protect\addvspace{10\p@}}%
2581     \addtocontents{lot}{\protect\addvspace{10\p@}}%
2582     \if@twocolumn
2583         \@topnewpage[\@makechapterhead{#2}]%
2584     \else
2585         \@makechapterhead{#2}%
2586     \@afterheading
2587 \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2588 \def\@makechapterhead#1{%
2589     % \vspace*{50\p@}%
2590     \@tempskipa = \glueexpr \CTEX@chapter@beforeskip \relax
2591     \ifodd \CTEX@chapter@fixbeforeskip
2592         \CTEX@fixbeforeskip
2593     \fi
2594     \vspace*{\@tempskipa}%
2595     % {\normalfont \parindent \z@ \raggedright
2596     {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2597     \CTEX@chapter@format
2598     \ifnum \c@secnumdepth >\m@ne
2599     <book>     \if@mainmatter
2600         \ifodd \CTEX@chapter@numbering
2601         % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2602         \CTEX@chaptername \CTEX@chapter@aftername
2603     \fi
2604     <book>     \fi
2605     \fi
2606     \interlinepenalty\@M
2607     % \Huge \bfseries #1\par\nobreak
2608     \CTEX@chapter@titleformat{#1}%
2609     \CTEX@chapter@aftertitle
2610     \nobreak

```

```

2611 % \vskip 40\p@
2612 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2613 }}

```

(End definition for \makechapterhead.)

\@makeschapterhead

```

2614 \def\@makeschapterhead#1{%
2615 % \vspace*{50\p@}%
2616 \@tempskipa = \glueexpr \CTEX@chapter@beforeskip \relax
2617 \ifodd \CTEX@chapter@fixbeforeskip
2618 \CTEX@fixbeforeskip
2619 \fi
2620 \vspace*{\@tempskipa}%
2621 % {\normalfont \parindent \z@ \raggedright
2622 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2623 \CTEX@chapter@format
2624 \interlinepenalty\@M
2625 % \Huge \bfseries #1\par\nobreak
2626 \CTEX@chapter@titleformat{#1}
2627 \CTEX@chapter@aftertitle
2628 \nobreak
2629 % \vskip 40\p@
2630 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2631 }}

```

(End definition for \@makeschapterhead.)

2632 </book|report>

14.8.2.3 section 类的标题

\@startsection

```

2633 \def\@startsection#1#2#3#4#5#6{%
2634 \if@noskipsec \leavevmode \fi
2635 \par
2636 % \@tempskipa #4\relax
2637 % \@afterindenttrue
2638 % \ifdim \@tempskipa <\z@
2639 % \@tempskipa -\@tempskipa \@afterindentfalse
2640 % \fi
2641 \CTEX@update@sectionformat@n{#1}%
2642 \ifodd \CTEX@afterindent
2643 \@afterindenttrue
2644 \else
2645 \@afterindentfalse
2646 \fi
2647 \if@nobreak
2648 \everypar{}%
2649 \else
2650 % \addpenalty\@secpenalty\addvspace\@tempskipa
2651 \addpenalty\@secpenalty\addvspace{\glueexpr #4\relax}%
2652 \fi
2653 \@ifstar
2654 {\@ssect{#3}{#4}{#5}{#6}}%
2655 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

(End definition for \@startsection.)

\@secntformat

```

2656 \def\@secntformat#1{%
2657 % \csname the#1\endcsname\quad}%
2658 \csname CTEX@#1name\endcsname \csname CTEX@#1aftername\endcsname}

```

(End definition for \@secntformat.)

\@sect

```

2659 \def\@sect#1#2#3#4#5#6[#7]#8{%
2660   \ifnum #2>\c@secnumdepth
2661     \let\@svsec\@empty
2662   \else
2663     \ifodd \csname CTEX@#1@numbering\endcsname
2664       \refstepcounter{#1}%
2665       \protected@edef\@svsec{\@secntformat{#1}\relax}%
2666     \else
2667       \CTEX@makeheadinganchor
2668       \let\@svsec\@empty
2669     \fi
2670   \fi
2671   % \@tempskipa #5\relax
2672   % \ifdim \@tempskipa>\z@
2673   \unless \ifodd \CTEX@runin
2674     \begingroup
2675     #6{%
2676       \@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2677     %   \interlinepenalty \@M #8\@par}%
2678     \interlinepenalty \@M
2679     \csname CTEX@#1@titleformat\endcsname{#8}%
2680     \csname CTEX@#1@aftertitle\endcsname}%
2681   \endgroup
2682   \csname #1mark\endcsname{#7}%
2683   \addcontentsline{toc}{#1}{%
2684     \ifnum #2>\c@secnumdepth \else
2685       \ifodd \csname CTEX@#1@numbering\endcsname
2686     %   \protect\numberline{\csname the#1\endcsname}%
2687     \protect\numberline{\csname CTEXthe#1\endcsname}%
2688     \fi
2689     \fi
2690     #7}%
2691   \else
2692     \def\@svsechd{%
2693       #6{\hskip\glueexpr #3\relax
2694     %   \@svsec #8}%
2695     \@svsec
2696     \csname CTEX@#1@titleformat\endcsname{#8}%
2697     \csname CTEX@#1@aftertitle\endcsname}%
2698     \csname #1mark\endcsname{#7}%
2699     \addcontentsline{toc}{#1}{%
2700       \ifnum #2>\c@secnumdepth \else
2701       \ifodd \csname CTEX@#1@numbering\endcsname
2702     %   \protect\numberline{\csname the#1\endcsname}%
2703     \protect\numberline{\csname CTEXthe#1\endcsname}%
2704       \fi
2705       \fi
2706       #7}}%
2707   \fi
2708   \@xsect{#5}}

```

(End definition for \@sect.)

\@ssect

```

2709 \def\@ssect#1#2#3#4#5{%
2710   % \@tempskipa #3\relax
2711   % \ifdim \@tempskipa>\z@
2712   \unless \ifodd \CTEX@runin
2713     \begingroup
2714     #4{%
2715       \@hangfrom{\hskip\glueexpr #1\relax}%
2716     %   \interlinepenalty \@M #5\@par}%

```

```

2717         \interlinepenalty \@M
2718         \CTEX@titleformat@n{#5}%
2719         \CTEX@aftertitle}%
2720     \endgroup
2721 \else
2722 %     \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2723     \def\@svsechd{#4{\hskip\glueexpr #1\relax
2724         \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2725 \fi
2726 \@xsect{#3}}

(End definition for \@ssect.)

```

\@xsect

```

2727 \def\@xsect#1{%
2728 % \@tempskipa #1\relax
2729 % \ifdim \@tempskipa>\z@
2730 \unless \ifodd \CTEX@runin
2731     \par \nobreak
2732 %     \vskip \@tempskipa
2733     \vskip\glueexpr #1\relax
2734     \@afterheading
2735 \else
2736     \@nobreakfalse
2737     \global\@noskipsectrue
2738     \everypar{%
2739         \if@noskipsec
2740             \global\@noskipsecfalse
2741             {\setbox\z@\lastbox}%
2742             \clubpenalty\@M
2743             \begingroup \@svsechd \endgroup
2744             \unskip
2745 %             \@tempskipa #1\relax
2746 %             \hskip -\@tempskipa
2747             \hskip\glueexpr #1\relax
2748         \else
2749             \clubpenalty \@clubpenalty
2750             \everypar{}%
2751         \fi}%
2752 \fi
2753 \ignorespaces}

(End definition for \@xsect.)

```

2754 <@@=ctex>

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

```

2755 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2756 {
2757     \cs_set_eq:Nc \CTEX@titleformat@n { \CTEX@#1@titleformat }
2758     \cs_set_eq:Nc \CTEX@aftertitle { \CTEX@#1@aftertitle }
2759     \cs_set_eq:Nc \CTEX@afterindent { \CTEX@#1@afterindent }
2760     \cs_set_eq:Nc \CTEX@runin { \CTEX@#1@runin }
2761 }
2762 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2763 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2764 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2765 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

(End definition for \CTEX@update@sectionformat@n.)

```

2766 \int_zero:N \l__ctex_tmp_int
2767 \seq_map_inline:Nn \c__ctex_section_headings_seq
2768 {
2769     \int_incr:N \l__ctex_tmp_int

```

```

2770 \cs_gset_protected_nopar:cpx {#1}
2771 {
2772   \exp_not:N \@startsection {#1}
2773   { \int_use:N \l__ctex_tmp_int }
2774   { \exp_not:c { CTEX@#1@indent } }
2775   { \exp_not:c { CTEX@#1@beforeskip } }
2776   { \exp_not:c { CTEX@#1@afterskip } }
2777   { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2778 }
2779 }

```

14.8.2.4 附录标题

```

appendix/name
appendix/number
appendix/numbering
2780 \keys_define:nn { ctex }
2781 {
2782   appendix .meta:nn = { ctex / appendix } {#1} ,
2783   appendix / name .code:n =
2784   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2785   appendix / number .tl_set:N = \CTEX@appendix@number ,
2786   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2787   appendix / numbering .initial:n = true
2788 }
2789 \tl_new:N \CTEX@preappendix
2790 \tl_new:N \CTEX@postappendix

```

(End definition for appendix/name, appendix/number, and appendix/numbering.)

\appendix

```

2791 \cs_new_eq:NN \CTEX@save@appendix \appendix
2792 \cs_gset_protected_nopar:Npn \appendix
2793 {
2794   \CTEX@save@appendix
2795   <*article>
2796   \gdef \CTEX@presection { \CTEX@preappendix }
2797   \gdef \CTEX@thesection { \CTEX@appendix@number }
2798   \gdef \CTEX@postsection { \CTEX@postappendix }
2799   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2800   </article>
2801   <*book|report>
2802   \gdef \CTEX@prechapter { \CTEX@preappendix }
2803   \gdef \CTEX@thechapter { \CTEX@appendix@number }
2804   \gdef \CTEX@postchapter { \CTEX@postappendix }
2805   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2806   </book|report>
2807 }

```

(End definition for \appendix.)

14.8.2.5 兼容 titlesec 宏包

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

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

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

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


我们这里的 $\langle shape \rangle$ 为 hang 或者 runin。 $\backslash titlespacing$ 的设置保存在 $\backslash ttls@ \langle section \rangle$ 之中,它的内容是

$$\{ \langle left \rangle \} \{ \langle right \rangle \} \{ \langle before \rangle \} \{ \langle after \rangle \} \{ \langle afterindent \rangle \}$$

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

```

2808 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
2809 {
2810   \ifpackagewith { titlesec } { explicit }
2811   {
2812     \cs_set_eq:NN \__ctex_titlesec_format:Nn
2813     \__ctex_titlesec_format_explicit:Nn
2814   }
2815   { }
2816   \clist_map_inline:nn
2817   { indentafter , noindentafter , indentfirst , nonindentfirst }
2818   {
2819     \ifpackagewith { titlesec } { ##1 }
2820     {
2821       \clist_map_break:n
2822       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
2823     }
2824     { }
2825   }
2826   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
2827 }
2828 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
2829 {
2830   \__ctex_titlesec_format:n {#1}
2831   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
2832 }
2833 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
2834 {
2835   \cs_if_free:cF { ttlf@#1 }
2836   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
2837 }
2838 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
2839 {
2840   \tl_set:Nx #1
2841   {
2842     \bool_if:cTF { CTEX@#2@runin }
2843     { \exp_not:N \ttlh@runin }
2844     { \exp_not:N \ttlh@hang }
2845     \tl_tail:N #1
2846   }
2847 }
2848 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
2849 {
2850   \cs_set_nopar:Npx #1 ##1
2851   {
2852     \bool_if:cTF { CTEX@#2@runin }
2853     { \exp_not:N \ttlh@runin }
2854     { \exp_not:N \ttlh@hang }
2855     \exp_args:No \tl_tail:n { #1 { } }
2856   }
2857 }
2858 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
2859 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
2860 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
2861 {

```

```

2862 \exp_not:n { {#1} {#2} {#3} {#4} }
2863 { \bool_if:cTF { CTEX@#6@afterindent } { \@one } { \z@ } }
2864 }

(End definition for \ctex_titlesec_hook:.)

2865 \@ifpackageloaded { titlesec }
2866 { }
2867 {
2868   \ctex_at_end_package:nn { titlesec }
2869   {
2870     \ifpackagewith { titlesec } { loadonly }
2871     { }
2872     { \ctex_titlesec_hook: }
2873   }
2874 }

```

14.8.2.6 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。

```

2875 \group_begin:
2876 \char_set_catcode_other:N \#
2877 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
2878 {
2879   \ctex_patch_cmd:Nnn \ttl@settopmark
2880   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
2881   {
2882     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
2883     \CTEX@titlepslabel@set {#1}
2884   }
2885   \ctex_patch_cmd:Nnn \ttl@setsubmark
2886   { \protect \@namedef { the#1 } { } }
2887   {
2888     \protect \@namedef { the#1 } { }
2889     \CTEX@titlepslabel@clear {#1}
2890   }
2891   \ctex_patch_cmd:Nnn \ttl@setsubmark
2892   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
2893   {
2894     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
2895     \CTEX@titlepslabel@set {#2}
2896   }
2897 }
2898 \group_end:

```

(End definition for \ctex_titleps_hook:.)

\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。

```

2899 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
2900 {
2901   \cs_if_free:cF { CTEXthe#1 }
2902   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
2903 }
2904 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
2905 {
2906   \cs_if_free:cF { CTEXthe#1 }

```

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

```

2907     { \protect \@namedef { CTEXthe#1 } { } }
2908 }

```

(End definition for \CTEX@titleslabel@set and \CTEX@titleslabel@clear.)

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

```

2909 \ctex_at_end_package:nn { titlesec }
2910 { \cs_if_free:NF \ttl@settopmark { \ctex_titles_hook: } }
2911 \ctex_at_end_package:nn { titles } { \ctex_titles_hook: }

```

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

```

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

```

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

14.8.3 目录标签的宽度

\numberline

```

2912 \cs_new_protected:Npn \CTEX@toc@width@n #1
2913 {
2914   \hbox_set:Nn \l__ctex_tmp_box {#1}
2915   \dim_set:Nn \@tempdima
2916   {
2917     \dim_max:nn { \@tempdima }
2918     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2919   }
2920 }
2921 \group_begin:
2922 \char_set_catcode_other:N \#
2923 \use:n
2924 {
2925   \group_end:
2926   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
2927   { \CTEX@toc@width@n {#1} }
2928   { }
2929   { \ctex_patch_failure:N \numberline }
2930   \ifpackageloaded { tocloft }
2931   { }
2932   {
2933     \ctex_at_end_package:nn { tocloft }
2934     {
2935       \ctex_preto_cmd:NnnTF \numberline
2936       { \char_set_catcode_letter:n { 64 } }
2937       { \CTEX@toc@width@n {#1} }
2938       { }
2939       { \ctex_patch_failure:N \numberline }
2940     }
2941   }
2942 }

```

(End definition for \numberline.)

14.8.4 页眉信息的修改

\ps@headings

```

2943 <*article>
2944 \ctex_patch_cmd:Nnn \ps@headings
2945 { \thesection \quad }
2946 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2947 \if@twoside
2948   \ctex_patch_cmd:Nnn \ps@headings

```

```

2949 { \thesubsection \quad }
2950 { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2951 \fi:
2952 </article>
2953 <*book|report>
2954 \ctex_patch_cmd:Nnn \ps@headings
2955 { \@chapapp\ \thechapter.\~\ }
2956 { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2957 \if@twoside
2958 \ctex_patch_cmd:Nnn \ps@headings
2959 { \thesection.\~\ }
2960 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2961 \fi:
2962 </book|report>

```

(End definition for \ps@headings.)

\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。

```

2963 \ctex_at_end_package:nn { fancyhdr }
2964 {
2965 <*article>
2966 \ctex_patch_cmd:Nnn \ps@fancy
2967 { \thesection \hspace{1em} \relax }
2968 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2969 \ctex_patch_cmd:Nnn \ps@fancy
2970 { \thesubsection \hspace{1em} \relax }
2971 { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2972 </article>
2973 <*book|report>
2974 \ctex_patch_cmd:Nnn \ps@fancy
2975 { \@chapapp\ \thechapter.\~\ }
2976 <book> { \if@mainmatter \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi \fi }
2977 <report> { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2978 \ctex_patch_cmd:Nnn \ps@fancy
2979 { \thesection.\~\ }
2980 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2981 </book|report>
2982 }

```

(End definition for \ps@fancy.)

2983 </article|book|report>

14.8.5 beamer 标题页模板的修改

```

2984 <*beamer>
2985 \ExplSyntaxOff

    对应 \partpage。

2986 \defbeamertemplate*{part page}{CTEX}[1][]{
2987 \begingroup
2988 % \centering
2989 % {\usebeamerfont{part name}\usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
2990 % \vskip1em\par
2991 \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
2992 \parindent \dimexpr \CTEX@part@indent \relax
2993 \CTEX@part@format
2994 \ifodd \CTEX@part@numbering
2995 \CTEX@partname \CTEX@part@aftername
2996 \fi
2997 \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
2998 % \usebeamerfont{part title}\insertpart\par
2999 \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle

```

```

3000     \end{beamercolorbox}
3001     \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3002 \endgroup
3003 }

```

对应 \sectionpage。

```

3004 \defbeamertemplate*{section page}{CTEX}[1][]{
3005   \begingroup
3006   % \centering
3007   % {\usebeamerfont{section name}\usebeamercolor[fg]{section name}\sectionname~\insertsectionnu
3008   % \vskip1em\par
3009   \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3010   \parindent \dimexpr \CTEX@section@indent \relax
3011   \CTEX@section@format
3012   \ifodd \CTEX@section@numbering
3013     \CTEX@sectionname \CTEX@section@aftername
3014   \fi
3015   \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3016   % \usebeamerfont{section title}\insertsection\par
3017     \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3018   \end{beamercolorbox}
3019   \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3020 \endgroup
3021 }

```

对应 \subsectionpage。

```

3022 \defbeamertemplate*{subsection page}{CTEX}[1][]{
3023   \begingroup
3024   % \centering
3025   % {\usebeamerfont{subsection name}\usebeamercolor[fg]{subsection name}\subsectionname~\insert
3026   % \vskip1em\par
3027   \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3028   \parindent \dimexpr \CTEX@subsection@indent \relax
3029   \CTEX@subsection@format
3030   \ifodd \CTEX@subsection@numbering
3031     \CTEX@subsectionname \CTEX@subsection@aftername
3032   \fi
3033   \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3034   % \usebeamerfont{subsection title}\insertsubsection\par
3035     \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3036   \end{beamercolorbox}
3037   \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3038 \endgroup
3039 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3040 \defbeamertemplatealias{part page}{default}{CTEX}
3041 \defbeamertemplatealias{section page}{default}{CTEX}
3042 \defbeamertemplatealias{subsection page}{default}{CTEX}
3043 \ExplSyntaxOn
3044 </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。

```

3045 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
3046 {
3047   \protected@edef \@currentlabel
3048   {
3049     \cs_if_exist:CTF { CTEX@the#1 }
3050     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3051     { \exp_not:o { \@currentlabel } }
3052   }
3053 }

```

(End definition for \refstepcounter.)

`\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。

```
3054 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
3055 {
3056   \seq_map_inline:Nn \c__ctex_headings_seq
3057   { \ctex_fix_varioref_label:n { ##1 } }
3058 }
```

(End definition for \ctex_varioref_hook:.)

`__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。

```
3059 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
3060 {
3061   \tl_if_empty:cT { p@#1 }
3062   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
3063 }
```

(End definition for __ctex_fix_varioref_label:n.)

如果 `varioref` 已经被载入,则使用它来设置。

```
3064 \@ifpackageloaded { varioref }
3065 { \ctex_varioref_hook: }
3066 {
3067   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3068   \RenewDocumentCommand \refstepcounter { m }
3069   {
3070     \CTEX@save@refstepcounter {#1}
3071     \CTEX@setcurrentlabel@n {#1}
3072   }
3073   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3074 }
```

14.8.7 载入 `<scheme>` 文件

```
3075 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3076 </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。

```
3077 <*scheme&(article|book|report|beamer)>

3078 \keys_set:nn { ctex / part }
3079 {
3080   aftertitle = \par ,
3081   <*plain>
3082   name = \partname \space ,
3083   <*article|book|report>
```

```

3084     number      = \thepart ,
3085 </article|book|report>
3086 <*beamer>
3087     number      = \insertromanpartnumber ,
3088 </beamer>
3089 </plain>
3090 <*chinese>
3091     number      = \chinese { part } ,
3092 </chinese>
3093 <*article>
3094     beforeskip  = 4ex ,
3095     afterskip   = 3ex ,
3096 <*plain>
3097     format      = \raggedright ,
3098     nameformat  = \Large \bfseries ,
3099     aftername   = \par \nobreak ,
3100     titleformat = \huge \bfseries ,
3101     afterindent = false
3102 </plain>
3103 <*chinese>
3104     format      = \Large \bfseries \centering ,
3105     aftername   = \quad ,
3106     afterindent = true
3107 </chinese>
3108 </article>
3109 <*book|report>
3110     aftername   = \par \vskip 20 \p@ ,
3111     beforeskip  = 0pt \@plus 1fil ,
3112     afterskip   = 0pt \@plus 1fil ,
3113     pagestyle   = plain ,
3114 <*plain>
3115     format      = \centering ,
3116     nameformat  = \huge \bfseries ,
3117     titleformat = \Huge \bfseries
3118 </plain>
3119 <*chinese>
3120     format      = \huge \bfseries \centering
3121 </chinese>
3122 </book|report>
3123 <*beamer>
3124     format      = \centering ,
3125     nameformat  = \usebeamerfont { part ~ name }
3126                  \usebeamercolor [fg] { part ~ name } ,
3127     aftername   = \vskip 1em \par ,
3128     titleformat = \usebeamerfont { part ~ title }
3129 </beamer>
3130 }

3131 <*book|report>
3132 \keys_set:nn { ctex / chapter }
3133 {
3134     pagestyle   = plain ,
3135     aftertitle  = \par ,
3136     beforeskip  = 50 \p@ ,
3137     afterskip   = 40 \p@ ,
3138 <*plain>
3139     name        = \chaptername \space ,
3140     number      = \thechapter ,
3141     format      = \raggedright ,
3142     nameformat  = \huge \bfseries ,
3143     aftername   = \par \nobreak \vskip 20 \p@ ,
3144     titleformat = \Huge \bfseries ,
3145     afterindent = false
3146 </plain>
3147 <*chinese>
3148     number      = \chinese { chapter } ,
3149     format      = \huge \bfseries \centering ,

```

```

3150     aftername    = \quad ,
3151     afterindent  = true
3152 </chinese>
3153 }
3154 </book|report>

3155 <@@= >

3156 \keys_set:nn { ctex / section }
3157 {
3158   <*article|book|report>
3159     number       = \thesection ,
3160     aftername    = \quad ,
3161     aftertitle   = \@@par ,
3162     beforeskip   = 3.5ex \@plus 1ex \@minus .2ex ,
3163     afterskip    = 2.3ex \@plus .2ex ,
3164     runin        = false ,
3165   <*plain>
3166     format       = \Large \bfseries ,
3167     afterindent  = false
3168   </plain>
3169   <*chinese>
3170     format       = \Large \bfseries \centering ,
3171     afterindent  = true
3172   </chinese>
3173   </article|book|report>
3174   <*beamer>
3175   <*plain>
3176     name         = \sectionname \space ,
3177   </plain>
3178     format       = \centering ,
3179     number       = \insertsectionnumber ,
3180     nameformat    = \usebeamerfont { section ~ name }
3181                   \usebeamercolor [fg] { section ~ name } ,
3182     aftername    = \vskip 1em \par ,
3183     titleformat  = \usebeamerfont { section ~ title } ,
3184     aftertitle   = \par
3185   </beamer>
3186 }

3187 \keys_set:nn { ctex / subsection }
3188 {
3189   <*article|book|report>
3190     number       = \thesubsection ,
3191     format       = \large \bfseries ,
3192     aftername    = \quad ,
3193     aftertitle   = \@@par ,
3194     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
3195     afterskip    = 1.5ex \@plus .2ex ,
3196     runin        = false ,
3197   <*plain>
3198     afterindent  = false
3199   </plain>
3200   <*chinese>
3201     afterindent  = true
3202   </chinese>
3203   </article|book|report>
3204   <*beamer>
3205   <*plain>
3206     name         = \subsectionname \space ,
3207     number       = \insertsubsectionnumber ,
3208   </plain>
3209   <*chinese>
3210     number       = \arabic { section } . \arabic { subsection } ,
3211   </chinese>
3212     format       = \centering ,
3213     nameformat    = \usebeamerfont { subsection ~ name }
3214                   \usebeamercolor [fg] { subsection ~ name } ,

```



```

3215     aftername    = \vskip 1em \par ,
3216     titleformat = \usebeamerfont { subsection ~ title } ,
3217     aftertitle   = \par
3218 </beamer>
3219 }

3220 < *article|book|report >

3221 \keys_set:nn { ctex / subsubsection }
3222 {
3223     number        = \thesubsubsection ,
3224     format        = \normalsize \bfseries ,
3225     aftername     = \quad ,
3226     aftertitle    = \@@par ,
3227     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3228     afterskip     = 1.5ex \@plus .2ex ,
3229     runin         = false ,
3230 < *plain >
3231     afterindent   = false
3232 </plain >
3233 < *chinese >
3234     afterindent   = true
3235 </chinese >
3236 }

3237 \keys_set:nn { ctex / paragraph }
3238 {
3239     number        = \theparagraph ,
3240     format        = \normalsize \bfseries ,
3241     aftername     = \quad ,
3242     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3243 < *plain >
3244     afterindent   = false
3245 </plain >
3246 < *chinese >
3247     afterindent   = true
3248 </chinese >
3249 }

3250 \keys_set:nn { ctex / subparagraph }
3251 {
3252     number        = \thesubparagraph ,
3253     format        = \normalsize \bfseries ,
3254     aftername     = \quad ,
3255     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3256 < *plain >
3257     afterindent   = false
3258 </plain >
3259 < *chinese >
3260     afterindent   = true
3261 </chinese >
3262 }

```

处理 sub3section 与 sub4section 的格式。

```

3263 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3264 {
3265     \keys_set:nn { ctex / paragraph }
3266     {
3267         aftertitle = \@@par ,
3268         afterskip  = 1ex \@plus .2ex ,
3269         runin      = false
3270     }
3271 }
3272 {
3273     \keys_set:nn { ctex / paragraph }
3274     {
3275         afterskip  = 1em ,

```

```

3276         runin      = true
3277     }
3278 }
3279 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3280 {
3281     \keys_set:nn { ctex / subparagraph }
3282     {
3283         aftertitle = \@@par ,
3284         afterskip  = 1ex \@plus .2ex ,
3285         runin      = false
3286     }
3287 }
3288 {
3289     \keys_set:nn { ctex / subparagraph }
3290     {
3291         afterskip  = 1em ,
3292         runin      = true
3293     }
3294 }
3295 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3296 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3297 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3298 <@@=ctex>

```

处理附录的格式。

```

3299 \keys_set:nn { ctex / appendix }
3300 <*article>
3301 { number      = \@Alph \c@section }
3302 </article>
3303 <*book|report>
3304 {
3305     name       = \appendixname \space ,
3306     number     = \@Alph \c@chapter
3307 }
3308 </book|report>
3309 </article|book|report>
3310 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

```

3311 <*ctex|ctexheading>

```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3312 \clist_map_inline:nn { article , book , report , beamer }
3313 {
3314     \ifclassloaded {#1}
3315     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3316     { }
3317 }

```

(End definition for \c__ctex_std_class_tl.)

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。

```

3318 \msg_new:nnn { ctex } { not-standard-class }
3319 {
3320     None~of~the~standard~document~classes~was~loaded.\\
3321     Heading~`#1'~is~selected.\\
3322     ctex~may~not~work~as~expected.
3323 }
3324 <ctex>\bool_if:NTF \l__ctex_heading_bool
3325 <ctexheading>\use:n

```

```

3326 {
3327   \tl_if_exist:NTF \c__ctex_std_class_tl
3328   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3329   {
3330     \cs_if_exist:NTF \chapter
3331     {
3332       \cs_if_exist:NF \if@mainmatter
3333       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3334       \tl_const:Nn \c__ctex_class_tl { book }
3335     }
3336     { \tl_const:Nn \c__ctex_class_tl { article } }
3337     \msg_warning:nx { ctex } { not-standard-class } { \c__ctex_class_tl }
3338   }
3339   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3340 }
3341 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3342 </ctex>[ctexheading]

```

14.8.10 标题配置文件

```

3343 <*name>
3344 \keys_set_known:nn { ctex }
3345 {
3346   contentsname = 目录 ,
3347   listfigurename = 插图 ,
3348   listtablename = 表格 ,
3349   figurename = 图 ,
3350   tablename = 表 ,
3351   abstractname = 摘要 ,
3352   indexname = 索引 ,
3353   bibname = 参考文献 ,
3354   appendixname = 附录 ,
3355   proofname = 证明 ,
3356   algorithmname = 算法 ,
3357   refname = 参考文献 ,
3358   continuation = (续) ,
3359   part / name = { 第 , 部分 } ,
3360   chapter / name = { 第 , 章 }
3361 }
3362 </name>

```

14.9 chinese 方案的其他设置

```

3363 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

3364 <article|book|report>\pagestyle { headings }

```

日期格式。

```

3365 \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。

```

3366 \ctex_if_autoindent_touched:F
3367 { \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。

```

3368 <!*generic>
3369 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3370 <*beamer>
3371 {
3372   \uselanguage { ChineseGBK }
3373   \languagealias { chinese } { ChineseGBK }
3374   \ctex_file_input:n { ctex-name-gbk.cfg }

```

```

3375 }
3376 {
3377   \uselanguage { ChineseUTF8 }
3378   \languagealias { chinese } { ChineseUTF8 }
3379   \ctex_file_input:n { ctex-name-utf8.cfg }
3380 }

```

让 translator 包优先查找中文翻译。

```

3381 \clist_put_left:Nn \trans@languagepath { chinese }
3382 </beamer>
3383 <*beamer>
3384 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3385 { \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。

```

3386 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3387 { \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。

```

3388 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3389 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3390 </beamer>
3391 </!generic>

```

不使用标题定义时的通用设置。

```

3392 <*generic>
3393 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3394 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3395 {
3396   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3397   {
3398     \uselanguage { ChineseGBK }
3399     \languagealias { chinese } { ChineseGBK }
3400     \ctex_file_input:n { ctex-name-gbk.cfg }
3401   }
3402   {
3403     \uselanguage { ChineseUTF8 }
3404     \languagealias { chinese } { ChineseUTF8 }
3405     \ctex_file_input:n { ctex-name-utf8.cfg }
3406   }
3407   \clist_put_left:Nn \trans@languagepath { chinese }
3408 }
3409 {
3410   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3411   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3412   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3413   \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3414   { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }
3415   \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3416   { \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。

```

3417   \tl_if_exist:NT \c__ctex_std_class_tl
3418   { \RequirePackage { indentfirst } }
3419 }
3420 </generic>
3421 </scheme&chinese>

```

14.10 中文字号

```

3422 <*class|ctex|ctexsize>

```

`\zihao`

```

3423 \NewDocumentCommand \zihao { m }
3424 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

(End definition for `\zihao`. This function is documented on page 23.)

`\ctex_zihao:n`

```

3425 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3426 {
3427   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3428   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3429   { \msg_error:nnn { ctex } { fontsize } {#1} }
3430 }
3431 \msg_new:nnnn { ctex } { fontsize }
3432 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3433 {
3434   The~old~font~size~is~used~if~you~continue.\\
3435   The~available~font~sizes~are~listed~as~follow.\\
3436   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3437 }

```

(End definition for `\ctex_zihao:n`.)

14.10.1 定义中文字号

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

```

3438 \prop_new:N \c__ctex_font_size_prop
3439 \seq_new:N \c__ctex_font_size_seq
3440 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3441 {
3442   \use:x
3443   {
3444     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3445     {
3446       { \dim_to_decimal:n {#2} }
3447       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3448     }
3449   }
3450   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3451 }
3452 \clist_map_inline:nn
3453 {
3454   { 8 } { 5 bp } ,
3455   { 7 } { 5.5 bp } ,
3456   { -6 } { 6.5 bp } ,
3457   { 6 } { 7.5 bp } ,
3458   { -5 } { 9 bp } ,
3459   { 5 } { 10.5 bp } ,
3460   { -4 } { 12 bp } ,
3461   { 4 } { 14 bp } ,
3462   { -3 } { 15 bp } ,
3463   { 3 } { 16 bp } ,
3464   { -2 } { 18 bp } ,
3465   { 2 } { 22 bp } ,
3466   { -1 } { 24 bp } ,
3467   { 1 } { 26 bp } ,
3468   { -0 } { 36 bp } ,
3469   { 0 } { 42 bp }
3470 }
3471 { \__ctex_save_font_size:nn #1 }

```

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

(End definition for \c__ctex_font_size_prop.)

\ctex_declare_math_sizes:nnnn

```

3472 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3473 {
3474   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3475   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3476 }

```

(End definition for \ctex_declare_math_sizes:nnnn.)

__ctex_get_font_sizes:Nn

```

3477 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3478 {
3479   \tl_clear:N #1
3480   \tl_map_inline:nn {#2}
3481   {
3482     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3483     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3484     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3485   }
3486 }

```

(End definition for __ctex_get_font_sizes:Nn.)

```

3487 \clist_map_inline:nn
3488 {
3489   { 8 }{ 8 }{ 5pt }{ 5pt } ,
3490   { 7 }{ 7 }{ 5pt }{ 5pt } ,
3491   { -6 }{ -6 }{ 5pt }{ 5pt } ,
3492   { 6 }{ 6 }{ 5pt }{ 5pt } ,
3493   { -5 }{ -5 }{ 6pt }{ 5pt } ,
3494   { 5 }{ 5 }{ 7pt }{ 5pt } ,
3495   { -4 }{ -4 }{ 8pt }{ 6pt } ,
3496   { 4 }{ 4 }{ 5 }{ 6 } ,
3497   { -3 }{ -3 }{ -4 }{ -5 } ,
3498   { 3 }{ 3 }{ 4 }{ 5 } ,
3499   { -2 }{ -2 }{ -3 }{ -4 } ,
3500   { 2 }{ 2 }{ 3 }{ 4 } ,
3501   { -1 }{ -1 }{ -2 }{ -3 } ,
3502   { 1 }{ 1 }{ 2 }{ 3 } ,
3503   { -0 }{ -0 }{ -1 }{ -2 } ,
3504   { 0 }{ 0 }{ 1 }{ 2 }
3505 }
3506 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

\ctex_set_font_size:Nnn

```

3507 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3508 {
3509   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3510   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3511   { \msg_error:nnn { ctex } { fontsize } {#2} }
3512 }
3513 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3514 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

(End definition for \ctex_set_font_size:Nnn.)

```

3515 \if_case:w \g__ctex_font_size_flag
3516 \ctex_file_input:n { ctex-c5size.clo }
3517 \or:

```

```

3518 \ctex_file_input:n { ctex-cs4size.clo }
3519 \fi:

3520 </class|ctex|ctexsize>

3521 <*c5size>
3522 \ctex_set_font_size:Nnn \normalsize { 5 }
3523 {
3524   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3525   \abovedisplayshortskip \z@ \@plus3\p@
3526   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3527   \belowdisplayskip \abovedisplayskip
3528   \let\@listi\@listI
3529 }
3530 \ctex_set_font_size:Nnn \small { -5 }
3531 {
3532   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3533   \abovedisplayshortskip \z@ \@plus2\p@
3534   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3535   \def\@listi{\leftmargin\leftmarginI
3536             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3537             \parsep 2\p@ \@plus\p@ \@minus\p@
3538             \itemsep \parsep}
3539   \belowdisplayskip \abovedisplayskip
3540 }
3541 \ctex_set_font_size:Nnn \footnotesize { 6 }
3542 {
3543   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3544   \abovedisplayshortskip \z@ \@plus\p@
3545   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3546   \def\@listi{\leftmargin\leftmarginI
3547             \topsep 3\p@ \@plus\p@ \@minus\p@
3548             \parsep 2\p@ \@plus\p@ \@minus\p@
3549             \itemsep \parsep}
3550   \belowdisplayskip \abovedisplayskip
3551 }
3552 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3553 \ctex_set_font_size:Nnn \tiny { 7 } { }
3554 \ctex_set_font_size:Nnn \large { -4 } { }
3555 \ctex_set_font_size:Nnn \Large { -3 } { }
3556 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3557 \ctex_set_font_size:Nnn \huge { 2 } { }
3558 \ctex_set_font_size:Nnn \Huge { 1 } { }
3559 </c5size>
3560 <*cs4size>
3561 \ctex_set_font_size:Nnn \normalsize { -4 }
3562 {
3563   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3564   \abovedisplayshortskip \z@ \@plus3\p@
3565   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3566   \belowdisplayskip \abovedisplayskip
3567   \let\@listi\@listI
3568 }
3569 \ctex_set_font_size:Nnn \small { 5 }
3570 {
3571   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3572   \abovedisplayshortskip \z@ \@plus3\p@
3573   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3574   \def\@listi{\leftmargin\leftmarginI
3575             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3576             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3577             \itemsep \parsep}
3578   \belowdisplayskip \abovedisplayskip
3579 }
3580 \ctex_set_font_size:Nnn \footnotesize { -5 }
3581 {
3582   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3583   \abovedisplayshortskip \z@ \@plus3\p@

```

```

3584 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3585 \def\@listi{\leftmargin\leftmargini
3586 \topsep 6\p@ \@plus2\p@ \@minus2\p@
3587 \parsep 3\p@ \@plus2\p@ \@minus\p@
3588 \itemsep \parsep}
3589 \belowdisplayskip \abovedisplayskip
3590 }
3591 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3592 \ctex_set_font_size:Nnn \tiny { -6 } { }
3593 \ctex_set_font_size:Nnn \large { -3 } { }
3594 \ctex_set_font_size:Nnn \Large { -2 } { }
3595 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3596 \ctex_set_font_size:Nnn \huge { -1 } { }
3597 \ctex_set_font_size:Nnn \Huge { 1 } { }
3598 </cs4size>
3599 <ctexsize>\normalsize
3600 <*classctex>

```

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`。

```

3601 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3602 {
3603 \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。

```

3604 \group_begin: \footnotesize \exp_args:NNNo \group_end:
3605 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3606 }

```

(End definition for `\footnotesep`.)

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

```

3607 \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。

```

3608 \NewDocumentCommand \CTeX { }
3609 { \ifinCsname CTeX \else: C \TeX \fi: }
3610 \ctex_at_end_package:nn { hyperref }
3611 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

(End definition for `\CTeX`. This function is documented on page 24.)

captiondelimater 过时选项。

```

3612 \keys_define:nn { ctex }
3613 {
3614 captiondelimater .code:n =
3615 {

```



```

3616         \msg_warning:nnn { ctex } { deprecated-option }
3617         { You~can~load~the~package~`caption'~to~get~its~functionality. }
3618     }
3619 }

```

(End definition for captiondelimiter. This function is documented on page 29.)

```
3620 </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`

```

3621 <*scheme&chinese&(article|book|report)>
3622 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3623 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3624 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3625 </scheme&chinese&(article|book|report)>

```

(End definition for \verse and \quotation.)

```
3626 <*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。

```

3627 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3628 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3629 \msg_new:nnn { ctex } { fontset-unavailable }
3630 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }

```

(End definition for \ctex_fontset_error:n.)

`\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。

```

3631 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3632 {
3633     \tl_if_empty:NTF \g__ctex_fontset_tl
3634     {
3635         \ctex_detected_platform:
3636         \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3637     }
3638     {
3639         \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3640         { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3641         {
3642             \use:x
3643             {
3644                 \ctex_detected_platform:
3645                 \msg_error:nxxx { ctex } { fontset-not-found }
3646                 { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3647             }
3648             \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3649         }
3650     }
3651 }
3652 \@onlypreamble \ctex_load_fontset:
3653 \msg_new:nnnn { ctex } { fontset-not-found }
3654 {
3655     CTeX~fontset~`#1'~could~not~be~found.\\
3656     Fontset~`#2'~will~be~used~instead.
3657 }
3658 { You~may~run~`mktexlsr'~firstly. }

```

(End definition for `\ctex_load_fontset::`)

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

3659 \keys_define:nn { ctex }
3660 {
3661   fontset .code:n =
3662   {
3663     \ctex_if_preamble:TF
3664     {
3665       \str_if_eq_x:nnTF {#1} { none }
3666       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3667       {
3668         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3669         {
3670           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3671           \ctex_load_fontset:
3672         }
3673         {
3674           \msg_error:nxxx { ctex } { fontset-loaded }
3675           { \g__ctex_fontset_tl } {#1}
3676         }
3677       }
3678     }
3679     { \msg_error:nn { ctex } { fontset-only-preamble } }
3680   }
3681 }
3682 \msg_new:nnnn { ctex } { fontset-loaded }
3683 {
3684   CTeX~fontset~`#1'~has~been~loaded.
3685   \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
3686 }
3687 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3688 \msg_new:nnn { ctex } { fontset-only-preamble }
3689 {
3690   The~`fontset'~option~can~be~used~only~in~preamble.
3691 }

```

(End definition for `fontset`. This function is documented on page 7.)

载入中文字库。

```

3692 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3693 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

3694 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3695 </class|ctex>
3696 <*config>
3697 %%
3698 </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。

```

3699 <*ctexopts>
3700 %%
3701 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3702 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

3703 <*c19|c70>
3704 %%
3705 %% Chinese characters
3706 %%
3707 <c19>%% character set: GBK (extension of GB 2312)
3708 <c70>%% character set: Unicode
3709 %% font encoding: Unicode
3710 %%
3711 </c19|c70>

```

CJK 宏包使用的字体族。

```

3712 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3713 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3714 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3715 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3716 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3717 <tt&c70>\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。

```

3718 <*uptex|aptex>
3719 \DeclareKanjiFamily{JY2}{rm}{-}{}
3720 \DeclareKanjiFamily{JY2}{sf}{-}{}
3721 \DeclareKanjiFamily{JY2}{tt}{-}{}
3722 \DeclareKanjiFamily{JT2}{rm}{-}{}
3723 \DeclareKanjiFamily{JT2}{sf}{-}{}
3724 \DeclareKanjiFamily{JT2}{tt}{-}{}
3725 </uptex|aptex>
3726 <*rm>
3727 <*c19>
3728 \DeclareFontShape{C19}{rm}{m}{n}{-}{} CJK * gbksong{\CJKnormal}
3729 \DeclareFontShape{C19}{rm}{b}{n}{-}{} CJK * gbkhei{\CJKnormal}
3730 \DeclareFontShape{C19}{rm}{bx}{n}{-}{} CJK * gbkhei{\CJKnormal}
3731 \DeclareFontShape{C19}{rm}{m}{sl}{-}{} CJK * gbksongsl{\CJKnormal}
3732 \DeclareFontShape{C19}{rm}{b}{sl}{-}{} CJK * gbkheisl{\CJKnormal}
3733 \DeclareFontShape{C19}{rm}{bx}{sl}{-}{} CJK * gbkheisl{\CJKnormal}
3734 \DeclareFontShape{C19}{rm}{m}{it}{-}{} CJK * gbkkai{\CJKnormal}
3735 \DeclareFontShape{C19}{rm}{b}{it}{-}{} CJKb * gbkkai{\CJKbold}
3736 \DeclareFontShape{C19}{rm}{bx}{it}{-}{} CJKb * gbkkai{\CJKbold}
3737 </c19>
3738 <*c70>
3739 \DeclareFontShape{C70}{rm}{m}{n}{-}{} CJK * unisong{\CJKnormal}
3740 \DeclareFontShape{C70}{rm}{b}{n}{-}{} CJK * unihei{\CJKnormal}
3741 \DeclareFontShape{C70}{rm}{bx}{n}{-}{} CJK * unihei{\CJKnormal}
3742 \DeclareFontShape{C70}{rm}{m}{sl}{-}{} CJK * unisongsl{\CJKnormal}
3743 \DeclareFontShape{C70}{rm}{b}{sl}{-}{} CJK * uniheisl{\CJKnormal}
3744 \DeclareFontShape{C70}{rm}{bx}{sl}{-}{} CJK * uniheisl{\CJKnormal}
3745 \DeclareFontShape{C70}{rm}{m}{it}{-}{} CJK * unikai{\CJKnormal}
3746 \DeclareFontShape{C70}{rm}{b}{it}{-}{} CJKb * unikai{\CJKbold}
3747 \DeclareFontShape{C70}{rm}{bx}{it}{-}{} CJKb * unikai{\CJKbold}
3748 </c70>
3749 </rm>
3750 <*uptex|aptex>
3751 \DeclareFontShape{JY2}{rm}{m}{n}{-}{} upzhserif-h{}
3752 \DeclareFontShape{JY2}{rm}{m}{it}{-}{} upzhserifit-h{}
3753 \DeclareFontShape{JY2}{rm}{bx}{n}{-}{} upzhserifb-h{}
3754 \DeclareFontShape{JT2}{rm}{m}{n}{-}{} upzhserif-v{}
3755 \DeclareFontShape{JT2}{rm}{m}{it}{-}{} upzhserifit-v{}
3756 \DeclareFontShape{JT2}{rm}{bx}{n}{-}{} upzhserifb-v{}
3757 </uptex|aptex>
3758 <*sf>

```

```

3759 <*c19>
3760 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3761 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3762 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3763 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3764 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3765 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3766 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3767 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3768 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3769 </c19>
3770 <*c70>
3771 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyousl}{\CJKnormal}
3772 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyousl}{\CJKbold}
3773 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyousl}{\CJKbold}
3774 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
3775 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3776 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3777 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyousl}{\CJKnormal}
3778 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyousl}{\CJKbold}
3779 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyousl}{\CJKbold}
3780 </c70>
3781 </sf>
3782 <*uptex|aptex>
3783 \DeclareFontShape{JY2}{sf}{m}{n}{<-> upzhsans-h}{\CJKnormal}
3784 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{\CJKbold}
3785 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{\CJKnormal}
3786 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{\CJKbold}
3787 </uptex|aptex>
3788 <*tt>
3789 <*c19>
3790 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfss}{\CJKnormal}
3791 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfss}{\CJKbold}
3792 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfss}{\CJKbold}
3793 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
3794 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3795 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3796 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfss}{\CJKnormal}
3797 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfss}{\CJKbold}
3798 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfss}{\CJKbold}
3799 </c19>
3800 <*c70>
3801 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifss}{\CJKnormal}
3802 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifss}{\CJKbold}
3803 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifss}{\CJKbold}
3804 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
3805 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
3806 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
3807 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifss}{\CJKnormal}
3808 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifss}{\CJKbold}
3809 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifss}{\CJKbold}
3810 </c70>
3811 </tt>
3812 <*uptex|aptex>
3813 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{\CJKnormal}
3814 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{\CJKnormal}
3815 </uptex|aptex>
3816 <*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。)

```

3817 <*windows>
3818 \file_if_exist:nTF { C:/bootfont.bin }
3819 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3820 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3821 </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。

```

3822 <*windowsnew>
3823 \tl_new:N \l__ctex_msyh_suffix_tl
3824 \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttc }
3825 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
3826 {
3827   \file_if_exist:nF { msyh.ttc }
3828   { \tl_set:Nn \l__ctex_msyh_suffix_tl { .ttf } }
3829 }
3830 </windowsnew>
3831 <*windowsnew|windowsold>
3832 \sys_if_engine_pdftex:TF
3833 {
3834   \ctex_zhmap_case:nnn
3835   {
3836     \ctex_punct_set:n { windows }
3837     \setCJKmainfont
3838     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3839 <*windowsold>
3840     \setCJKsansfont { simhei.ttf }
3841     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3842 </windowsold>
3843 <*windowsnew>
3844     \setCJKsansfont [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
3845     \setCJKfamilyfont { zhyahei }
3846     [ BoldFont = msyhbd\l__ctex_msyh_suffix_tl ] { msyh\l__ctex_msyh_suffix_tl }
3847     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3848     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3849 </windowsnew>
3850     \setCJKmonofont { simfang.ttf }
3851     \setCJKfamilyfont { zhkai } { simkai.ttf }
3852     \setCJKfamilyfont { zhfs } { simfang.ttf }
3853     \setCJKfamilyfont { zhsong } { simsun.ttc }
3854     \setCJKfamilyfont { zhhei } { simhei.ttf }
3855     \setCJKfamilyfont { zhli } { simli.ttf }
3856     \setCJKfamilyfont { zhyou } { simyou.ttf }
3857     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3858     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3859     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3860     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3861   }
3862   {
3863     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3864     \ctex_punct_set:n { windows }
3865     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3866     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3867     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3868   }
3869   {
3870     \tl_set:Nn \CJKrmdefault { rm }
3871     \tl_set:Nn \CJKsfdefault { sf }
3872     \tl_set:Nn \CJKttdefault { tt }
3873   }
3874 }
3875 {
3876   \sys_if_engine_uptex:TF
3877   {
3878     \ctex_set_upfonts:nnnnnn

```

```

3879         {simsun.ttc} {simhei.ttf} {simkai.ttf}
3880 <windowsnew>         {msyh\l__ctex_msyh_suffix_tl} {msyhbdl__ctex_msyh_suffix_tl}
3881 <windowsold>         {simhei.ttf} {simhei.ttf}
3882         {simfang.ttf}
3883         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3884         \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
3885         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3886         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3887 <windowsnew>         \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
3888         \ctex_set_upfamily:nnn { zhli } { upschrn } {}
3889         \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
3890         \ctex_set_upfamily:nnn { zhyou } { upschg } {}
3891         \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
3892     }
3893     {
3894 <*windowsold>
3895         \setCJKmainfont
3896         [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3897         \setCJKsansfont { SimHei }
3898         \setCJKmonofont { FangSong_GB2312 }
3899         \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3900         \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3901 </windowsold>
3902 <*windowsnew>
3903         \setCJKmainfont
3904         [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3905         \setCJKsansfont
3906         [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3907         \setCJKmonofont { FangSong }
3908         \setCJKfamilyfont { zhkai } { KaiTi }
3909         \setCJKfamilyfont { zhfs } { FangSong }
3910 </windowsnew>
3911         \setCJKfamilyfont { zhsong } { SimSun }
3912         \setCJKfamilyfont { zhhei } { SimHei }
3913         \setCJKfamilyfont { zhli } { LiSu }
3914         \setCJKfamilyfont { zhyou } { YouYuan }
3915 <*windowsnew>
3916         \setCJKfamilyfont { zhyahei }
3917         [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3918 </windowsnew>
3919     }
3920 }
3921 </windowsnew|windowsold>

```

14.15.3 ctex-fontset-adobe.def

```

3922 <*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。

```

3923 \sys_if_engine_pdftex:TF
3924 {
3925     \sys_if_output_pdf:TF
3926     { \ctex_fontset_error:n { adobe } }
3927     {
3928         \ctex_zhmap_case:nnn
3929         {
3930             \setCJKmainfont
3931             [
3932                 cmap = UniGB-UTF16-H ,
3933                 BoldFont = AdobeHeitiStd-Regular.otf ,
3934                 ItalicFont = AdobeKaitiStd-Regular.otf
3935             ] { AdobeSongStd-Light.otf }
3936             \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3937             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3938             \setCJKfamilyfont { zhsong }

```

```

3939         [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3940     \setCJKfamilyfont { zhhei }
3941     [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3942     \setCJKfamilyfont { zhkai }
3943     [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3944     \setCJKfamilyfont { zhfs }
3945     [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3946     \ctex_punct_set:n { adobe }
3947     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3948     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3949     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3950     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3951     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3952 }
3953 {
3954     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3955     \ctex_punct_set:n { adobe }
3956     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3957     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3958     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3959 }
3960 { \ctex_fontset_error:n { adobe } }
3961 }
3962 }
3963 {
3964     \sys_if_engine_uptex:TF
3965     {
3966         \ctex_set_upfonts:nnnnnn
3967         {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
3968         {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
3969         {AdobeFangsongStd-Regular.otf}
3970         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3971         \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
3972         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3973         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3974     }
3975     {
3976         \setCJKmainfont
3977         [
3978             BoldFont = AdobeHeitiStd-Regular ,
3979             ItalicFont = AdobeKaitiStd-Regular
3980         ] { AdobeSongStd-Light }
3981         \setCJKsansfont { AdobeHeitiStd-Regular}
3982         \setCJKmonofont { AdobeFangsongStd-Regular}
3983         \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3984         \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3985         \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3986         \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3987     }
3988 }
3989 </adobe>
3990 <*fandol>

```

14.15.4 ctex-fontset-fandol.def

```

3991 \sys_if_engine_pdftex:TF
3992 {
3993     \sys_if_output_pdf:TF
3994     { \ctex_fontset_error:n { fandol } }
3995     {
3996         \ctex_zhmap_case:nnn
3997         {
3998             \setCJKmainfont
3999             [
4000                 cmap = UniGB-UTF16-H ,
4001                 BoldFont = FandolSong-Bold.otf ,
4002                 ItalicFont = FandolKai-Regular.otf

```



```

4003         ] { FandolSong-Regular.otf }
4004     \setCJKsansfont
4005     [
4006         cmap = UniGB-UTF16-H ,
4007         BoldFont = FandolHei-Bold.otf
4008     ] { FandolHei-Regular.otf }
4009     \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4010     \setCJKfamilyfont { zhsong }
4011     [
4012         cmap = UniGB-UTF16-H ,
4013         BoldFont = FandolSong-Bold.otf
4014     ] { FandolSong-Regular.otf }
4015     \setCJKfamilyfont { zhhei }
4016     [
4017         cmap = UniGB-UTF16-H ,
4018         BoldFont = FandolHei-Bold.otf
4019     ] { FandolHei-Regular.otf }
4020     \setCJKfamilyfont { zhfs }
4021     [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
4022     \setCJKfamilyfont { zhkai }
4023     [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
4024     \ctex_punct_set:n { fandol }
4025     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4026     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4027     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4028     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4029     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4030     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
4031 }
4032 {
4033     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4034     \ctex_punct_set:n { fandol }
4035     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4036     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4037     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4038 }
4039 { \ctex_fontset_error:n { fandol } }
4040 }
4041 }
4042 {
4043     \sys_if_engine_uptex:TF
4044     {
4045         \ctex_set_upfonts:nnnnnn
4046         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
4047         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
4048         {FandolFang-Regular.otf}
4049         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4050         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4051         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4052         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4053     }
4054     {
4055         \setCJKmainfont
4056         [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
4057         { FandolSong-Regular }
4058         \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4059         \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
4060         \setCJKfamilyfont { zhsong }
4061         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4062         \setCJKfamilyfont { zhhei }
4063         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4064         \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4065         \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4066     }
4067 }
4068 </fandol>
4069 <*mac>

```


14.15.5 ctex-fontset-mac.def

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

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

在 DVIPDFMx 下,可以通过下述方式使用华文宋体和华文楷体:

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

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

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

DVIPDFMx 不能生成 PDF,报下述错误:

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

如果将 CMap 改为 UniGB-UTF16-H,错误信息是

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

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

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

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

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

```

4070 \sys_if_engine_pdftex:TF
4071 { \ctex_fontset_error:n { mac } }
4072 {
4073   \sys_if_engine_uptex:TF
4074   { \ctex_fontset_error:n { mac } }
4075   {
4076     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4077     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4078     \setCJKmonofont { STFangsong }
4079     \setCJKfamilyfont { zhsong } { STSong }
4080     \setCJKfamilyfont { zhhei } { STHeiti }
4081     \setCJKfamilyfont { zhfs } { STFangsong }
4082     \setCJKfamilyfont { zhkai } { STKaiti }
4083   }
4084 }
4085 </mac>
4086 <*founder>

```

14.15.6 ctex-fontset-founder.def

```

4087 \sys_if_engine_pdftex:TF
4088 {
4089   \ctex_zhmap_case:nnn
4090   {
4091     \setCJKmainfont
4092       [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4093     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4094     \setCJKmonofont { FZFSK.TTF }
4095     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4096     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4097     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4098     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4099     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4100     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4101     \ctex_punct_set:n { founder }
4102     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4103     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4104     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4105     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4106     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4107     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4108     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4109   }
4110   {
4111     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4112     \ctex_punct_set:n { founder }
4113     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4114     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4115     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4116   }
4117   { \ctex_fontset_error:n { founder } }
4118 }
4119 {
4120   \sys_if_engine_uptex:TF
4121   {
4122     \ctex_set_upfonts:nnnnnn
4123       { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4124       { FZXH1K.TTF } { FZHTK.TTF }
4125       { FZFSK.TTF }
4126     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4127     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4128     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4129     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4130     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4131     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4132     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}

```

```

4133     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4134   }
4135   {
4136     \setCJKmainfont
4137       [ 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), $\text{X}_{\text{L}}\text{TeX}$ 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。

```

4138     \setCJKsansfont [ BoldFont = FZHei-B01 ]
4139       { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4140     \setCJKmonofont { FZFangSong-Z02 }
4141     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4142     \setCJKfamilyfont { zhhei } { FZHei-B01 }
4143     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4144     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4145     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4146     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4147   }
4148 }
4149 </founder>
4150 <*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。

```

4151 \sys_if_engine_pdftex:TF
4152 {
4153   \ctex_zhmap_case:nnn
4154   {
4155     \setCJKmainfont
4156       [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4157     \setCJKsansfont { wqy-zenhei.ttc }
4158     \setCJKmonofont { uming.ttc }
4159     \setCJKfamilyfont { zhsong } { uming.ttc }
4160     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4161     \setCJKfamilyfont { zhkai } { ukai.ttc }
4162     \ctex_punct_set:n { ubuntu }
4163     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4164     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4165     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4166     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4167     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4168   }
4169   {
4170     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4171     \ctex_punct_set:n { ubuntu }
4172     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4173     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4174     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4175   }
4176   { \ctex_fontset_error:n { ubuntu } }

```

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

```

4177 }
4178 {
4179   \sys_if_engine_uptex:TF
4180   {
4181     \ctex_set_upfonts:nnnnnn
4182     { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4183     { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4184     { uming.ttc }
4185     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4186     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4187     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4188   }
4189   {
4190     \setCJKmainfont
4191     [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4192     \setCJKsansfont { WenQuanYi~Zen~Hei }
4193     \setCJKmonofont { AR~PL~UMing~CN }
4194     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4195     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4196     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4197   }
4198 }
4199 </ubuntu>

```

14.15.8 中文字体命令

```

4200 <!*windows>
4201 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4202 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
4203 </ubuntu>\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4204 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4205 <*windowsnew|windowsold|founder>
4206 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4207 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4208 </windowsnew|windowsold|founder>
4209 <windowsnew>\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4210 </!windows>
4211 </fontset>
4212 <*zhmap>

```

14.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4213 \begingroup\catcode61\catcode48\catcode32=10\relax%
4214 \catcode 35=6 % #
4215 \catcode123=1 % {
4216 \catcode125=2 % }
4217 \toks0{\endlinechar=\the\endlinechar\relax}%
4218 \toks2{\endlinechar=13}%
4219 \def\x#1 #2 {%
4220   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4221   \toks2\expandafter{\the\toks2 \catcode#1=#2}}%
4222 \x 13 5 % carriage return
4223 \x 32 10 % space
4224 \x 35 6 % #
4225 \x 40 12 % (
4226 \x 41 12 % )
4227 \x 45 12 % -
4228 \x 46 12 % .
4229 \x 47 12 % /
4230 \x 58 12 % :
4231 \x 60 12 % <
4232 \x 61 12 % =
4233 \x 64 11 % @

```

```

4234 \x 91 12 % [
4235 \x 93 12 % ]
4236 \x 123 1 % {
4237 \x 125 2 % }
4238 \edef\x#1{\endgroup%
4239 \edef\noexpand#1{%
4240 \the\toks0 %
4241 \let\noexpand\noexpand\noexpand#1%
4242 \noexpand\noexpand\noexpand\noexpand\undefined%
4243 \noexpand\noexpand\noexpand\endinput}%
4244 \the\toks2}%
4245 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4246 \begingroup\expandafter\endgroup
4247 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4248 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4249 \expandafter\ifx\csname pdfoutput\endcsname\relax
4250 \iffalse\else\ifnum\pdfoutput < 1 \iffalse\else \iftrue\fi\fi
4251 \else ifctexpdf\fi
4252 \endcsname

```

提供非 \LaTeX 格式下的 `\ProvidesFile`。

```

4253 \begingroup
4254 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4255 \long\def\x#1\ProvidesFile#2[#3]{%
4256 #1%
4257 \immediate\write-1{File: #2 #3}%
4258 \expandafter\xdef\csname ver@#2\endcsname{#3}}
4259 \expandafter\x%
4260 \fi
4261 \endgroup

```

14.15.9.1 zhwindowsfonts.tex

```

4262 <*windows>
4263 \ProvidesFile{zhwindowsfonts.tex}%
4264 [2016/04/23 v2.4 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4265
4266 \ifzhmappdf
4267 \pdfmapline{=gbk@UGBK@ <simsum.ttc}
4268 \pdfmapline{=gbksong@UGBK@ <simsum.ttc}
4269 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4270 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4271 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4272 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4273 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4274
4275 \pdfmapline{=cyberb@Unicode@ <simsum.ttc}
4276 \pdfmapline{=unisong@Unicode@ <simsum.ttc}
4277 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4278 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4279 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4280 \pdfmapline{=unili@Unicode@ <simli.ttf}
4281 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4282
4283 \pdfmapline{=gbksongsl@UGBK@ <simsum.ttc}
4284 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4285 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4286 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4287 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4288 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4289
4290 \pdfmapline{=unisongsl@Unicode@ <simsum.ttc}
4291 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4292 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4293 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}

```

```

4294 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4295 \pdfmapline{=uniyouisl@Unicode@ <simyou.ttf}
4296
4297 \else
4298 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4299 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4300 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4301 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4302 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4303 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4304 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4305
4306 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4307 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4308 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4309 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4310 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4311 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4312 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4313
4314 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4315 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4316 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4317 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4318 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4319 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
4320
4321 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4322 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4323 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4324 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4325 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4326 \special{pdf:mapline uniyouisl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4327
4328 </windows>

```

14.15.9.2 zhadobefonts.tex

```

4329 <*adobe>
4330 \ProvidesFile{zhadobefonts.tex}%
4331 [2016/04/23 v2.4 Adobe font map loader for DVIPDFMx (CTEX)]
4332
4333 \ifzhmappdf
4334 %% pdfTeX does not support OTF fonts
4335
4336 \else
4337 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4338 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4339 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4340 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4341 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4342
4343 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4344 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4345 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4346 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4347 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4348
4349 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4350 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4351 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4352 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4353
4354 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4355 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4356 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4357 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4358

```


4359 </adobe>

14.15.9.3 zhfandolfonts.tex

```

4360 <*fandol>
4361 \ProvidesFile{zhfandolfonts.tex}%
4362 [2016/04/23 v2.4 Fandol font map loader for DVIPDFMx (CTEX)]
4363
4364 \ifzhmappdf
4365 %% pdfTeX does not support OTF fonts
4366
4367 \else
4368 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4369 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4370 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4371 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4372 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4373
4374 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4375 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4376 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4377 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4378 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4379
4380 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4381 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4382 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4383 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4384
4385 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4386 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4387 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4388 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4389
4390 </fandol>

```

14.15.9.4 zhfounderfonts.tex

```

4391 <*founder>
4392 \ProvidesFile{zhfounderfonts.tex}%
4393 [2016/04/23 v2.4 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4394
4395 \ifzhmappdf
4396 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF>}
4397 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF>}
4398 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF>}
4399 \pdfmapline{=gbkhei@UGBK@ <FZHKT.TTF>}
4400 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF>}
4401 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF>}
4402 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF>}
4403
4404 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF>}
4405 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF>}
4406 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF>}
4407 \pdfmapline{=unihei@Unicode@ <FZHKT.TTF>}
4408 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF>}
4409 \pdfmapline{=unili@Unicode@ <FZLSK.TTF>}
4410 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF>}
4411
4412 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF>}
4413 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF>}
4414 \pdfmapline{=gbkheisl@UGBK@ <FZHKT.TTF>}
4415 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF>}
4416 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF>}
4417 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF>}
4418
4419 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF>}
4420 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF>}

```

```

4421 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4422 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4423 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4424 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4425
4426 \else
4427 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4428 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4429 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4430 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4431 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4432 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4433 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4434
4435 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4436 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4437 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4438 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4439 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4440 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4441 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF}
4442
4443 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4444 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4445 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4446 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4447 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4448 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4449
4450 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4451 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4452 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4453 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4454 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4455 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4456
4457 </founder>

```

14.15.9.5 zhubuntufonts.tex

```

4458 <*ubuntu>
4459 \ProvidesFile{zhubuntufonts.tex}%
4460 [2016/04/23 v2.4 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4461
4462 \ifzhmappdf
4463 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4464 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4465 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4466 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4467 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4468 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4469
4470 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4471 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4472 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4473 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4474 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4475 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4476
4477 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4478 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4479 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4480 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4481 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4482
4483 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4484 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4485 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}

```



```

4486 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4487 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4488
4489 \else
4490 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4491 \special{pdf:mapline gbkkai@UGBK@ unicode :0:uming.ttc}
4492 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4493 \special{pdf:mapline gbkkai@UGBK@ unicode :0:wqy-zenhei.ttc}
4494 \special{pdf:mapline gbkkai@UGBK@ unicode :0:uming.ttc}
4495
4496 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4497 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4498 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4499 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4500 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4501
4502 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4503 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4504 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4505 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:uming.ttc -s .167}
4506
4507 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4508 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4509 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4510 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4511
4512 </ubuntu>
4513 \fi
4514
4515 \ctex@zhmap@endinput
4516 </zhmap>

```

14.15.10 制作 spa 文件

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

```

4517 <*spa>
4518 <*macro>
4519 \input expl3-generic %
4520 \ExplSyntaxOn
4521 \sys_if_engine_xetex:F
4522 {
4523   \msg_new:nn { ctex } { xetex }
4524   { XeTeX~is~required~to~compile~this~document! }
4525   \msg_fatal:nn { ctex } { xetex }
4526 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

4527 \seq_new:N \c__ctex_punct_seq
4528 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4529 {
4530   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4531   "3008 , "300A , "3016 , "3010 ,
4532   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4533   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4534   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4535 }

```

\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。

```

4536 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4537 {
4538   \iow_open:Nn \g__ctex_spa_iow {#1}
4539   \clist_map_inline:nn {#2}
4540   { \__ctex_write_family:nn ##1 }
4541   \iow_close:N \g__ctex_spa_iow
4542 }
4543 \iow_new:N \g__ctex_spa_iow
4544 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

(End definition for \ctex_make_spa:nn.)

4545 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4546 {
4547   \group_begin:
4548   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4549   \l__ctex_punct_font
4550   \clist_clear:N \l__ctex_punct_bounds_clist
4551   \seq_map_inline:Nn \c__ctex_punct_seq
4552   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4553   \iow_now:Nx \g__ctex_spa_iow
4554   {
4555     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4556       { \l__ctex_punct_bounds_clist , , , }
4557     }
4558   \group_end:
4559 }
4560 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4561 {
4562   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4563   {
4564     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4565     \__ctex_calc_bounds:nn { \c_three } {#1}
4566   }
4567 }
4568 \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。

```

4569 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4570 {
4571   \fp_eval:n
4572   {
4573     round
4574     (
4575       \dim_to_decimal_in_unit:nn
4576       { 100 \xetex_glyphbounds:D #1 ~ #2 }
4577       { 1 em }
4578     )
4579   }
4580 }
4581 \ExplSyntaxOff
4582 </macro>

```

下面是 CTeX 定义的一些字体。

```

4583 <*make>
4584 \input ctexspamacro %
4585
4586 \MAKESPA {ctexpunct.spa}
4587 {
4588   {adobezhsong}      {AdobeSongStd-Light} ,
4589   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4590   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4591   {adobezhfs}        {AdobeFangsongStd-Regular} ,

```

```

4592 {fandolzh song} {FandolSong} ,
4593 {fandolzh songb} {FandolSong-Bold} ,
4594 {fandolzhhei} {FandolHei} ,
4595 {fandolzhheib} {FandolHei-Bold} ,
4596 {fandolzhkai} {FandolKai} ,
4597 {fandolzhfs} {FandolFang} ,
4598 {founderzh song} {FZShuSong-Z01} ,
4599 {founderzh songb} {FZXiaoBiaoSong-B05} ,
4600 {founderzhhei} {FZHei-B01} ,
4601 {founderzhheil} {FZXiHeiI-Z08} ,
4602 {founderzhkai} {FZKai-Z03} ,
4603 {founderzhfs} {FZFangSong-Z02} ,
4604 {founderzhli} {FZLiShu-S01} ,
4605 {founderzhyou} {FZXiYuan-M01} ,
4606 {founderzhyoub} {FZZhunYuan-M02} ,
4607 {ubuntuzh song} {AR PL Uming CN} ,
4608 {ubuntuzhhei} {WenQuanYi Zen Hei} ,
4609 {ubuntuzhkai} {AR PL UKai CN} ,
4610 {windowszh song} {SimSun} ,
4611 {windowszhhei} {SimHei} ,
4612 {windowszhkai} {KaiTi} ,
4613 {windowszhfs} {FangSong} ,
4614 {windowszhli} {LiSu} ,
4615 {windowszhyou} {YouYuan} ,
4616 {windowszhyahei} {Microsoft YaHei} ,
4617 {windowszhyaheib} {Microsoft YaHei Bold}
4618 }
4619
4620 \primitive\end
4621 </make>
4622 </spa>

```

14.16 translator 宏包的中文字典

```
4623 <*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。

```

4624 <*theorem>
4625 \providetranslation{Comments}{评论}
4626 \providetranslation{comments}{评论}
4627 \providetranslation{Comment}{评论}
4628 \providetranslation{comment}{评论}
4629 \providetranslation{Corollaries}{推论}
4630 \providetranslation{corollaries}{推论}
4631 \providetranslation{Corollary}{推论}
4632 \providetranslation{corollary}{推论}
4633 \providetranslation{Definitions}{定义}
4634 \providetranslation{definitions}{定义}
4635 \providetranslation{Definition}{定义}
4636 \providetranslation{definition}{定义}
4637 \providetranslation{Examples}{例}
4638 \providetranslation{examples}{例}
4639 \providetranslation{Example}{例}
4640 \providetranslation{example}{例}
4641 \providetranslation{Exercises}{练习}
4642 \providetranslation{exercises}{练习}
4643 \providetranslation{Exercise}{练习}
4644 \providetranslation{exercise}{练习}
4645 \providetranslation{Facts}{事实}
4646 \providetranslation{facts}{事实}
4647 \providetranslation{Fact}{事实}
4648 \providetranslation{fact}{事实}
4649 \providetranslation{Key Lemmas}{关键引理}
4650 \providetranslation{key lemmas}{关键引理}

```

```

4651 \providetranslation{Key Lemma}{关键引理}
4652 \providetranslation{key lemma}{关键引理}
4653 \providetranslation{Key Observations}{关键观察}
4654 \providetranslation{key observations}{关键观察}
4655 \providetranslation{Key Observation}{关键观察}
4656 \providetranslation{key observation}{关键观察}
4657 \providetranslation{Lemmas}{引理}
4658 \providetranslation{lemmas}{引理}
4659 \providetranslation{Lemma}{引理}
4660 \providetranslation{lemma}{引理}
4661 \providetranslation{Main Theorems}{主要定理}
4662 \providetranslation{main theorems}{主要定理}
4663 \providetranslation{Main Theorem}{主要定理}
4664 \providetranslation{main theorem}{主要定理}
4665 \providetranslation{Observations}{观察}
4666 \providetranslation{observations}{观察}
4667 \providetranslation{Observation}{观察}
4668 \providetranslation{observation}{观察}
4669 \providetranslation{Problems}{问题}
4670 \providetranslation{problems}{问题}
4671 \providetranslation{Problem}{问题}
4672 \providetranslation{problem}{问题}
4673 \providetranslation{Proofs}{证明}
4674 \providetranslation{proofs}{证明}
4675 \providetranslation{Proof}{证明}
4676 \providetranslation{proof}{证明}
4677 \providetranslation{Proof Sketch}{证明提要}
4678 \providetranslation{Proof sketch}{证明提要}
4679 \providetranslation{proof sketch}{证明提要}
4680 \providetranslation{Proof Sketches}{证明提要}
4681 \providetranslation{Proof sketches}{证明提要}
4682 \providetranslation{proof sketches}{证明提要}
4683 \providetranslation{Sketch of Proof}{证明提要}
4684 \providetranslation{Sketch of Proofs}{证明提要}
4685 \providetranslation{Sketch of proof}{证明提要}
4686 \providetranslation{Sketch of proofs}{证明提要}
4687 \providetranslation{sketch of proof}{证明提要}
4688 \providetranslation{sketch of proofs}{证明提要}
4689 \providetranslation{Propositions}{命题}
4690 \providetranslation{propositions}{命题}
4691 \providetranslation{Proposition}{命题}
4692 \providetranslation{proposition}{命题}
4693 \providetranslation{Remarks}{注}
4694 \providetranslation{remarks}{注}
4695 \providetranslation{Remark}{注}
4696 \providetranslation{remark}{注}
4697 \providetranslation{Solutions}{解}
4698 \providetranslation{solutions}{解}
4699 \providetranslation{Solution}{解}
4700 \providetranslation{solution}{解}
4701 \providetranslation{Theorems}{定理}
4702 \providetranslation{theorems}{定理}
4703 \providetranslation{Theorem}{定理}
4704 \providetranslation{theorem}{定理}
4705 </theorem>
4706 </dict>

```

14.17 ctexcap 宏包

```
4707 <*ctexcap>
```

ctexcap 是过时宏包。

```

4708 \clist_new:N \l__ctex_ctexcap_options_clist
4709 \clist_set:Nx \l__ctex_ctexcap_options_clist
4710 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
4711 \msg_new:nnn { ctexcap } { deprecated }

```

```

4712 {
4713   Package~`ctexcap'~is~deprecated.\\
4714   Please~use~package~`ctex'~with~option~`#1'~instead: \\ \\ \\
4715   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4716 }
4717 \msg_warning:nxx { ctexcap } { deprecated }
4718 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

4719 \PassOptionsToPackage { heading = true } { ctexcap }
4720 \RequirePackageWithOptions { ctex }
4721 </ctexcap>

```

14.18 ctexhook 宏包

```

4722 <*ctexhook>

```

\ctex_at_end_preamble:n 实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

\ctex_after_end_preamble:n
4723 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4724 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
4725 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4726 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
4727 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4728 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4729 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4730 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4731 \cs_set_nopar:Npx \document
4732 {
4733   \CTEX@document@left@hook
4734   \exp_not:o { \document }
4735   \CTEX@document@right@hook
4736 }
4737 \tl_new:N \g__ctex_end_preamble_hook_tl
4738 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

(End definition for \ctex_at_end_preamble:n and \ctex_after_end_preamble:n.)

\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。

```

4739 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4740 {
4741   \@ifpackageloaded {#1}
4742   {#2}
4743   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
4744 }

```

(End definition for \ctex_at_end_package:nn.)

\ctex_gadd_hook:Nn 给钩子附加内容。

```

\ctex_gadd_hook:cn
4745 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4746 {
4747   \tl_if_exist:NF #1 { \tl_new:N #1 }
4748   \tl_gput_right:Nn #1 {#2}
4749 }
4750 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

(End definition for \ctex_gadd_hook:Nn and \ctex_gadd_hook:cn.)

\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
4751 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
4752 {

```

```

4753 \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
4754 { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
4755 }
4756 \cs_generate_variant:Nn \ctex_package_end_hook:n { o }

```

(End definition for \ctex_package_end_hook:n and \ctex_package_end_hook:o.)

对 \@popfilename 做补丁来实现 \ctex_at_end_package:nn 的功能。

```

4757 \tl_put_left:Nn \@popfilename
4758 {
4759   \cs_if_eq:NNT \@currentx \pkgextension
4760   { \ctex_package_end_hook:o { \@currname } }
4761 }
4762 </ctexhook>

```

14.19 ctexpatch 宏包

```

4763 <*ctexpatch>

```

```

4764 \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。

```

4765 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
4766 {
4767   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4768   { once } {#2} { \use_i:nn } { \use_ii:nn }
4769 }

```

(End definition for \ctex_patch_cmd_once:NnnnTF.)

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

```

4770 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
4771 {
4772   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4773   { all } {#2} { \use_i:nn } { \use_ii:nn }
4774 }

```

(End definition for \ctex_patch_cmd_all:NnnnTF.)

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

```

4775 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
4776 {
4777   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4778   { once }
4779   {
4780     \ExplSyntaxOff
4781     \char_set_catcode_letter:n { 64 }
4782   }
4783   { }
4784   { \ctex_patch_failure:N #1 }
4785 }
4786 \cs_new_protected:Npn \ctex_patch_failure:N #1
4787 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
4788 \msg_new:nnn { ctex } { patch-failure }
4789 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

(End definition for \ctex_patch_cmd:Nnn.)

\ctex_preto_cmd:NnnTF 在宏的原本定义前面增加钩子。

```

4790 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2

```

```

4791 {
4792   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnw #1
4793   { left } {#2} { \use_i:nn } { \use_ii:nn }
4794 }

```

(End definition for \ctex_preto_cmd:NnnTF.)

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```

4795 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
4796 {
4797   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnw #1
4798   { right } {#2} { \use_i:nn } { \use_ii:nn }
4799 }

```

(End definition for \ctex_appto_cmd:NnnTF.)

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

```

4800 \cs_new_protected:Npn \ctex_patch_boot:NnnTF #1#2#3#4#5#6
4801 {
4802   \tl_set:Nn \__ctex_patch_true:w {#5}
4803   \tl_set:Nn \__ctex_patch_false:w {#6}
4804   \group_begin:
4805     \char_set_catcode_other:n { 35 }
4806     \ctex_parse_name:NN #1 #2 {#3} {#4}
4807   }
4808   \tl_new:N \__ctex_patch_true:w
4809   \tl_new:N \__ctex_patch_false:w

```

(End definition for \ctex_patch_boot:NnnTF.)

\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。

```

4810 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
4811 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
4812 \group_begin:
4813 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
4814 {
4815   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
4816   {
4817     \bool_if:nTF
4818     {
4819       \cs_if_exist_p:c { ##3 ~ } ||
4820       \cs_if_exist_p:c { #1##3 }
4821     }
4822     {
4823       \group_begin:
4824       \use:x

```

```

4825         {
4826             \group_end:
4827             \__ctex_parse_name:nNNNnN
4828             { \token_get_replacement_spec:N ##2 }
4829             \exp_not:N ##2
4830             \exp_not:c { ##3 ~ }
4831             \exp_not:c { #1##3 }
4832         } {##3} ##1
4833     }
4834     { ##1##2 }
4835 }
4836 \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
4837 {
4838     \exp_args:Nc ##6
4839     {
4840         \str_case:nnTF {##1}
4841         {
4842             { \protect ##3 } { }
4843             { \x@protect ##2 \protect ##3 } { }
4844         }
4845         {
4846             \str_if_eq_x:nnTF
4847             { \exp_not:n { #1@protected@ ##3 #1##3 } }
4848             {
4849                 \exp_last_unbraced:Nf \__ctex_parse_name:w
4850                 \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop
4851             }
4852             { #1##5 ~ } { ##5 ~ }
4853         }
4854         {
4855             \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
4856             {
4857                 { #1@protected@ ##2 ##4 } { }
4858                 { #1@ ##4 } { }
4859             }
4860             { #1##5 } {##5}
4861         }
4862     }
4863 }
4864 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
4865 }
4866 \use:x
4867 {
4868     \__ctex_tmp:w
4869     { \c_backslash_str }
4870     { \c_left_brace_str }
4871     { \tl_to_str:n { testopt } }
4872 }
4873 \group_end:
4874 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

(End definition for \ctex_parse_name:NN.)

```

\l__ctex_prefix_str  分别保存宏的 \meaning 中的前缀、参数文本和替换文本。
\l__ctex_parameter_str
\l__ctex_replacement_str
4875 \str_new:N \l__ctex_prefix_str
4876 \str_new:N \l__ctex_parameter_str
4877 \str_new:N \l__ctex_replacement_str

```

(End definition for \l__ctex_prefix_str, \l__ctex_parameter_str, and \l__ctex_replacement_str.)

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

```

\__ctex_get_macro_meaning:w
4878 \group_begin:
4879 \cs_set_protected:Npn \__ctex_tmp:w #1
4880 {
4881     \prg_new_protected_conditional:Npnn

```



```

4882 \ctex_get_macro_meaning:N ##1 { TF }
4883 {
4884   \exp_after:wN \__ctex_get_macro_meaning:w
4885   \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
4886 }
4887 \cs_new_protected:Npn \__ctex_get_macro_meaning:w
4888   ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
4889 {
4890   \tl_if_empty:nTF { ##4 }
4891     { \prg_return_false: }
4892     {
4893       \str_set:Nn \l__ctex_prefix_str { ##1 }
4894       \str_set:Nn \l__ctex_parameter_str { ##2 }
4895       \str_set:Nn \l__ctex_replacement_str { ##3 }
4896       \prg_return_true:
4897     }
4898 }
4899 }
4900 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
4901 \group_end:

```

(End definition for \ctex_get_macro_meaning:NTF.)

\ctex_if_rescanable:NnTF 检查宏是否可以重建。

```

4902 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
4903 {
4904   \ctex_get_macro_meaning:NTF #1
4905   {
4906     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
4907     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
4908   }
4909   {#4}
4910 }
4911 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

```

(End definition for \ctex_if_rescanable:NnTF.)

__ctex_patch_rebuild:Nn 使用 \tl_rescan:nn 来重新记号化 \meaning 字符串。

```

4912 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
4913 {
4914   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
4915   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
4916   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
4917   \use:x
4918   {
4919     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
4920     \exp_not:o { \l__ctex_parameter_tl }
4921     { \exp_not:o { \l__ctex_replacement_tl } }
4922   }
4923 }
4924 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
4925 {
4926   \str_if_empty:NTF #2
4927     { \tl_clear:N #1 }
4928     { \tl_set_rescan:Nno #1 {#3} {#2} }
4929 }
4930 \tl_new:N \l__ctex_prefix_tl
4931 \tl_new:N \l__ctex_parameter_tl
4932 \tl_new:N \l__ctex_replacement_tl

```

(End definition for __ctex_patch_rebuild:Nn.)

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

```

4933 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnnw #1#2#3#4#5
4934 {
4935   \group_end:
4936   \ctex_if_rescanable:NnTF #1 {#3}
4937   {
4938     \use:x
4939     {
4940       \__ctex_patch_replace:nnnTF {#2}
4941       { \tl_to_str:n {#4} }
4942       { \tl_to_str:n {#5} }
4943     }
4944     {
4945       \__ctex_patch_rebuild:Nn #1 {#3}
4946       \__ctex_patch_true:w
4947     }
4948     { \__ctex_patch_false:w }
4949   }
4950   { \__ctex_patch_false:w }
4951 }

```

(End definition for __ctex_patch_cmd:Nnnnnw.)

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

```

4952 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
4953 {
4954   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
4955   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
4956 }

```

(End definition for __ctex_patch_replace:nnnTF.)

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

4957 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
4958 {
4959   \group_end:
4960   \ctex_get_macro_meaning:NnTF #1
4961   {
4962     \str_if_empty:NnTF \l__ctex_parameter_str
4963     { \__ctex_hookto_cmd_parameterless:Nnnnw }
4964     { \__ctex_hookto_cmd_parameter:Nnnnw }
4965     #1 {#2} {#3} {#4}
4966   }
4967   { \__ctex_patch_false:w }
4968 }

```

(End definition for __ctex_hookto_cmd:Nnnnw.)

__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。

```

4969 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
4970 {
4971   \str_if_empty:NnTF \l__ctex_prefix_str
4972   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
4973   \tex_edef:D #1
4974   {
4975     \use:c { __ctex_ #2 _hook_aux:nn }
4976     { \exp_not:o {#1} }
4977     { \exp_not:n {#4} }
4978   }
4979   \__ctex_patch_true:w
4980 }
4981 \cs_generate_variant:Nn \tl_rescan:nn { no }
4982 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
4983 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

(End definition for _ctex_hookto_cmd_parameterless:Nnnnw.)

_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。

```

4984 \cs_new_protected:Npn \_ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
4985 {
4986   \_ctex_patch_rebuild:Nn \_ctex_rebuild_cmd:w {#3}
4987   \cs_if_eq:NNTF #1 \_ctex_rebuild_cmd:w
4988   {
4989     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
4990     \_ctex_patch_rebuild:Nn #1 {#3}
4991     \_ctex_patch_true:w
4992   }
4993   { \_ctex_patch_false:w }
4994 }

```

(End definition for _ctex_hookto_cmd_parameter:Nnnnw.)

4995 </ctexpatch>

版本历史

v2.0

General: c5size, cs4size 是过时选项。	36
captiondelimiter 是过时选项。	109
fancyhdr 成为过时选项, 原选项功能总是打开。	39
fntef 成为过时选项, 原选项功能总是打开。	39
hyperref 成为过时选项, 原选项功能总是打开。	39
indent, noindent 是过时选项。	37
nofonts, adobefonts, winfonts 是过时选项。	37
nopunct 是过时选项。	38
nospace 是过时选项。	38
nozhmap 是过时选项。	38
punct 选项可以设置标点格式。	38
ctex 宏包新增 heading 选项。	39
\CTEXindent, \CTEXnoindent 是过时命令。	81
\CTEXsetup, \CTEXoptions 是过时命令。	14, 41
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; \CTEXfilltwosides 是过时环 境。	74
标题设置新增 pagestyle 选项。	85
调整 \footnotesep 的大小, 以适合行距的变化。	109
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号 选项。	40
将标题汉化功能加入 ctex.sty。	82
将中文字号功能提取到可以独立使用的 ctexsize。 . . .	106
解决 etoolbox 与 breqn 关于 \end 的冲突。	110
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	74
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jan 宏包支持 Lua _{La} TeX。	48
新增 autoindent 选项。	36
新增 fontset 选项。	37
新增 linespread 选项。	36
新增 linestretch 选项。	79
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。 .	39
新增 zhmcjk 支持选项。	38
新增 zihao 选项。	36
新增统一设置接口 \ctexset。	41
应用 L _A T _E X3 重新整理代码。	1
中文字号不再采用近似值。	106
自动检测操作系统, 载入对应的字体配置。	71
\CJK@surr: 解决与 \nouppercase 的冲突。	44

v2.0.1

General: 修复 10pt、11pt 等选项无效的问题。	40
---	----

v2.0.2

General: 修复加载 ctex 宏包后章节标题后第一段无段首 缩进的问题。	105
---	-----

v2.1

General: format+, nameformat+ 等带加号的选项, 加号与 前面的文字之间可以有可选的空格。	84
nameformat 可以接受章节名字为参数。	83
不依赖 ifpdf 宏包。	34
不再设置 hyperref 宏包的 colorlinks 选项。	72
给 pdf _{La} TeX 下的非 UTF8 编码 CJK 字体族加上 CMap。 42	

将章节标题设置功能提取到可以独立使用的宏包

ctexheading 中。	1
新的标题格式选项 aftertitle。	84
修复 ctexbook 和 ctexrep 类的中文 part/number 选项 初值为空的错误。	99
\ctex_if_macosx:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	72
hyperref: 补充定义 \hypersetup。	40

v2.2

General: before skip 和 after skip 选项的符号不再有特 殊意义。	84
before skip、after skip 和 indent 选项支持表达式。 . .	86
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/before skip 和 part/after skip 选项有意义。	86
给 enumitem 宏包注册 \chinese 和 \zhnum。	81
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。 .	130
将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。	99
删去 etoolbox 与 breqn 的兼容补丁。	110
新的标题格式选项 afterindent。	84
新的标题格式选项 numbering。	84
新的标题格式选项 runin。	84
新增子宏包 ctexpatch 实现给宏打补丁功能。	131

v2.3

General: .value_required: 和 .value_forbidden: 已 过时。	36
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。 .	34
更新 LuaTeX-jan 支持 (20150922.0)。	48
更新 unicode-math 宏包补丁。	48
兼容 titleps 宏包。	95
修复 nameformat 作用域问题。	83
与 L _A T _E X3 (2015/12/20) 同步。	32

v2.4

General: 初步支持 up _{La} TeX。	68
加强 beamer 宏包支持。	1
提供 translator 宏包的中文定理名称翻译。	128
提供 up _{La} TeX 的 NFSS 字体定义。	112
正确更新 CJK 包的 \CJKfamilydefault。	70
正确设置 up _{TeX} 下字体命令。	69
\ctex_parse_name:NN: 修复宏名解析错误。	132

v2.4.1

General: beamer 不调整默认行距。	105
beamer 不调整默认字体大小。	105
使用 bootfont.bin 判断 Windows XP 以避免权限问 题。	113
随字体更新 up _{TeX} 的 \xkanjiskip。	77
新的标题格式选项 part/fixbefore skip 和 chapter/fixbefore skip。	85
正确更新 up _{La} TeX 的 \CJKfamilydefault。	70
\ccwd: 正确设置 up _{TeX} 下的 \ccwd。	75
\chinese: 支持 \pagenumbering。	81
zihao: 不允许无参 zihao 选项。	36

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

_	2955, 2959, 2975, 2979
\#	491, 2876, 2922
.../afterindent	20
.../aftername	18
.../aftername+	18
.../afterskip	21
.../aftertitle	19
.../aftertitle+	19
.../beforeskip	20
.../format	16
.../format+	16
.../indent	21
.../name	15
.../nameformat	16
.../nameformat+	16
.../number	15
.../numberformat	18
.../numberformat+	18
.../titleformat	18
.../titleformat+	18
\/	1629
\\	15, 33, 34, 102, 145, 147, 149, 205, 215, 677, 1120, 1202, 2249, 3320, 3321, 3434, 3435, 3655, 3685, 4713, 4714, 4715, 4789
\{	1131, 1132, 4715
\}	1131, 1132, 4715
10pt	8, <u>40</u>
11pt	8, <u>40</u>
12pt	8, <u>40</u>

A

\abovedisplayshortskip	3525, 3533, 3544, 3564, 3572, 3583
\abovedisplayskip	3524, 3527, 3532, 3539, 3543, 3550, 3563, 3566, 3571, 3578, 3582, 3589
\abstractname	2233
abstractname	12
\addCJKfontfeature	1202, 1225
\addCJKfontfeatures	<u>59</u> , 1220, 1225
\addcontentsline	2455, 2456, 2459, 2462, 2485, 2486, 2489, 2492, 2566, 2571, 2574, 2577, 2683, 2699
\AddEnumerateCounter	2193, 2195, 2196, 2197
\addpenalty	2650, 2651
\addtocontents	2580, 2581
\addvspace	2418, 2419, 2580, 2581, 2650, 2651, 2991, 3001, 3009, 3019, 3027, 3037
adobefonts	28
\advance	2407, 2408
\algorithmname	2240, 2256
algorithmname	12
AlternateFont	25, <u>61</u>

\appendix	<u>93</u> , 2791, 2792
appendix/name	22, <u>93</u>
appendix/number	22, <u>93</u>
appendix/numbering	21, <u>93</u>
\appendixname	2235, 3305
appendixname	12
\arabic	3210
\AtBeginDocument	579, 1618
\AtBeginDvi	449
\AtBeginShipoutFirst	451, 1821, 1837
\AtBeginUTFCommand	721
\AtEndOfClass	135
\AtEndOfPackage	136
\AtEndUTFCommand	727
autoindent	10, <u>36</u> , <u>78</u>

B

backslash commands:

\c_backslash_str	4869
\baselinestretch	2103
\begin	2997, 3015, 3033
\begingroup	2674, 2713, 2743, 2987, 3005, 3023, 4213, 4246, 4253
\belowdisplayshortskip	3526, 3534, 3545, 3565, 3573, 3584
\belowdisplayskip	3527, 3539, 3550, 3566, 3578, 3589
\bfdefault	983, 989, 1315, 1322
\bfseries	2468, 2471, 2499, 2502, 2513, 2526, 2601, 2607, 2625, 3098, 3100, 3104, 3116, 3117, 3120, 3142, 3144, 3149, 3166, 3170, 3191, 3224, 3240, 3253
\bibname	2238, 2241, 2249, 2257, 2263, 2264
bibname	12

bool commands:

\bool_gset_false:N	265, 270
\bool_gset_true:N	260
\bool_if:cTF	2842, 2852, 2863
\bool_if:NT	1160
\bool_if:NTF	70, 3324
\bool_if:nTF	4817
\bool_new:N	131, 1197
\bool_set_false:N	1172
\bool_set_true:N	1190

box commands:

\box_ht:N	3605
\box_new:N	61
\box_wd:N	1909, 2918

C

c5size	28
cap	28
captiondelimiter	29, 109
\catcode	4213, 4214, 4215, 4216, 4220, 4221
CCT	28

- CCTfont 28
- \ccwd 24, 75, 115, 188, 1909, 1913, 1916, 1918, 1955, 1956, 1957, 2054, 2092, 2111, 2112, 2119, 2146, 2147, 2148, 2151, 2153, 2154, 2165, 2176, 2177, 3622, 3623, 3624
- \centering 2496, 2524, 2988, 3006, 3024, 3104, 3115, 3120, 3124, 3149, 3170, 3178, 3212
- \chapter 88, 2548, 3330
- chapter/beforeskip 30
- chapter/fixbeforeskip 20, 85
- chapter/numbering 15
- chapter/pagestyle 21, 85
- \chaptermark 2579
- \chaptername 3139
- char commands:
- \char_generate:nn 747
- \char_set_catcode:nn 73, 552, 553
- \char_set_catcode_active:N 1350
- \char_set_catcode_letter:n 67, 1621, 2936, 4781
- \char_set_catcode_math_superscript:n 547
- \char_set_catcode_other:N 491, 916, 1333, 2876, 2922
- \char_set_catcode_other:n 546, 4805
- \char_value_catcode:n 73, 552, 553
- CharRange 25, 61
- \Chinese 29, 2189
- \chinese 24, 81, 2187, 2189, 2197, 3091, 3148
- \CJK 1874
- \CJKbold 3735, 3736, 3746, 3747, 3761, 3762, 3764, 3765, 3767, 3768, 3772, 3773, 3775, 3776, 3778, 3779, 3791, 3792, 3794, 3795, 3797, 3798, 3802, 3803, 3805, 3806, 3808, 3809
- \CJKfamily 57, 526, 530, 534, 538, 1057, 1079, 1259, 1263, 1267, 1271, 1419, 1640, 4201, 4202, 4203, 4204, 4206, 4207, 4209
- \CJKfamilydefault 520, 521, 538, 594, 1253, 1254, 1271, 1279, 1281, 1285, 1292, 1305, 1636, 1645, 1711, 1715, 1723, 1736, 1738
- \CJKfilltwosides 1882
- \CJKfontspec 59, 1215
- \CJKglue 1908, 1922, 1939, 1952
- \CJKhook 46, 609
- \CJKnormal 3728, 3729, 3730, 3731, 3732, 3733, 3734, 3739, 3740, 3741, 3742, 3743, 3744, 3745, 3760, 3763, 3766, 3771, 3774, 3777, 3790, 3793, 3796, 3801, 3804, 3807
- \CJKpunctmapfamily 642, 643, 644, 645, 646, 647, 651, 652, 653, 654, 655, 656
- \CJKrmdefault 514, 521, 522, 526, 1127, 1138, 1228, 1254, 1255, 1259, 1281, 1284, 1285, 1633, 1636, 1637, 1719, 3857, 3859, 3860, 3865, 3866, 3867, 3870, 3947, 3950, 3951, 3956, 3957, 3958, 4025, 4028, 4029, 4035, 4036, 4037, 4102, 4105, 4106, 4113, 4114, 4115, 4163, 4166, 4167, 4172, 4173, 4174
- \CJKsfdefault 515, 523, 530, 1128, 1139, 1234, 1256, 1263, 1634, 1638, 1720, 3841, 3847, 3848, 3871, 3948, 4026, 4030, 4103, 4107, 4164
- \CJKtilde 595
- \CJKttdefault 516, 524, 534, 1129, 1140, 1239, 1257, 1267, 1635, 1639, 1721, 3858, 3872, 3949, 4027, 4104, 4165
- clearalternatefont 26, 65
- \cleardoublepage 2430, 2548
- \clearpage 2432, 2548
- clist commands:
- \clist_clear:N 1004, 4550
- \clist_concat:NNN 1026
- \clist_const:Nn 622
- \clist_gput_right:Nn 382, 404, 406
- \clist_gput_right:No 394
- \clist_gset:Nn 1245
- \clist_if_empty:NT 1338, 1597
- \clist_if_empty:NTF 1357
- \clist_map_break:n 2821, 3315
- \clist_map_function:nN 1548, 1562
- \clist_map_function:xN 1514, 1516
- \clist_map_inline:Nn 612, 1450
- \clist_map_inline:nn 367, 634, 1555, 1866, 1892, 2816, 3312, 3452, 3487, 4539
- \clist_new:N 138, 1023, 1246, 1392, 4568, 4708
- \clist_put_left:Nn 3381, 3407
- \clist_put_left:Nx 1027
- \clist_put_right:Nn 1189
- \clist_put_right:Nx 1438, 4562
- \clist_set:Nn 1375, 1376
- \clist_set:No 1008
- \clist_set:Nx 4709
- \clist_use:Nn 4718
- \clubpenalty 2742, 2749
- colon commands:
- \c_colon_str 941, 942, 943
- \contentsname 2228
- contentsname 11
- continuation 12
- cs commands:
- \cs:w 3062
- \cs_end: 3062
- \cs_generate_variant:Nn 93, 467, 479, 764, 968, 1091, 1105, 1178, 1198, 1199, 1371, 1408, 1510, 1548, 1556, 4750, 4756, 4874, 4981
- \cs_gset_eq:cc 1165, 1168
- \cs_gset_eq:NN 161, 162, 163, 261, 266, 271, 309, 311, 377, 381, 499, 811, 815, 3387, 3414
- \cs_gset_nopar:Npx 812
- \cs_gset_protected_nopar:cpx 1053, 2770
- \cs_gset_protected_nopar:Npn 752, 2401, 2792
- \cs_gset_protected_nopar:Npx 850, 1412, 1506
- \cs_if_eq:cNF 952
- \cs_if_eq:NNT 4759
- \cs_if_eq:NNTF 4907, 4987
- \cs_if_exist:cF 806, 848, 886
- \cs_if_exist:cTF 3049
- \cs_if_exist:NF 358, 476, 818, 1411, 1503, 1973, 3332, 4764
- \cs_if_exist:NTF 44, 3330
- \cs_if_exist_p:c 4819, 4820

- \cs_if_exist_use:c 1075, 1076
- \cs_if_exist_use:cF 4, 789
- \cs_if_exist_use:cT 1525, 4753
- \cs_if_exist_use:cTF 1097
- \cs_if_free:cF 461, 614, 629, 1162, 2835, 2901, 2906
- \cs_if_free:cTF 1314
- \cs_if_free:Nf 471, 2081, 2193, 2910
- \cs_if_free:Nf 2067
- \cs_new:Npn
110, 117, 124, 907, 913, 917, 1000, 1078, 2860, 4864, 4982
- \cs_new_eq:cc 1901
- \cs_new_eq:cN 6, 7, 8, 9, 20, 23, 1887
- \cs_new_eq:Nc 1964
- \cs_new_eq:NN 95, 106, 107, 108, 133, 134,
152, 359, 468, 599, 660, 827, 852, 928, 959, 1225, 1231,
1732, 1888, 1889, 1952, 1963, 2188, 2189, 2190, 2762,
2763, 2764, 2765, 2791, 3067, 3328, 3333, 4544, 4911, 4983
- \cs_new_nopar:Npn
867, 1070, 1123, 1134, 2187, 2899, 2904, 4569
- \cs_new_protected:Npn 62,
745, 750, 1001, 1020, 1336, 1427, 1441, 1592, 1958,
2368, 2912, 3507, 3513, 4545, 4723, 4725, 4739, 4745,
4765, 4770, 4775, 4786, 4790, 4795, 4800, 4810, 4815,
4836, 4887, 4902, 4912, 4924, 4933, 4952, 4957, 4969, 4984
- \cs_new_protected_nopar:cpx 1871
- \cs_new_protected_nopar:Npn
77, 79, 135, 136, 447, 459,
469, 474, 480, 512, 542, 559, 601, 607, 610, 627, 632,
640, 649, 658, 729, 787, 796, 809, 814, 816, 828, 842,
869, 884, 897, 921, 930, 947, 961, 978, 1024, 1032, 1051,
1071, 1081, 1106, 1144, 1156, 1158, 1180, 1182, 1275,
1300, 1309, 1355, 1366, 1393, 1402, 1409, 1415, 1435,
1448, 1467, 1481, 1488, 1500, 1520, 1536, 1554, 1557,
1559, 1572, 1584, 1626, 1650, 1663, 1677, 1709, 1741,
1759, 1774, 1790, 1801, 1805, 1844, 1851, 1855, 1878,
1884, 1905, 1919, 1954, 1988, 2066, 2068, 2093, 2128,
2138, 2143, 2292, 2322, 2399, 2403, 2755, 2808, 2828,
2833, 2838, 2848, 2858, 2877, 3045, 3054, 3059, 3425,
3440, 3472, 3477, 3627, 3631, 4536, 4560, 4727, 4729, 4751
- \cs_new_protected_nopar:Npx 484, 1211
- \cs_set:Npn 1334
- \cs_set_eq:cc 616, 630
- \cs_set_eq:cN 1969
- \cs_set_eq:Nc 2757, 2758, 2759, 2760
- \cs_set_eq:NN 109, 189, 194, 199,
539, 557, 577, 604, 608, 767, 770, 821, 822, 929, 960,
998, 1057, 1272, 1411, 1419, 1429, 1430, 1431, 1503,
1504, 1505, 1629, 1714, 2204, 2209, 2212, 2217, 2812, 2822
- \cs_set_nopar:cpx 1902
- \cs_set_nopar:Npn 1685
- \cs_set_nopar:Npx 2850, 4731
- \cs_set_protected:Npn 732, 4813, 4879
- \cs_set_protected_nopar:Npn 603, 771, 1922, 3514
- \cs_set_protected_nopar:Npx 1960
- \cs_to_str:N 54, 1210, 4811
- \cs_undefine:c 1036, 1037, 1041, 1042, 4754
- \cs_undefine:N 1432
- cs4size 28
- \csname 2657, 2658,
2663, 2679, 2680, 2682, 2685, 2686, 2687, 2696, 2697,
2698, 2701, 2702, 2703, 4245, 4247, 4248, 4249, 4254, 4258
- \CTeX 24, 109, 3608, 3611
- ctex commands:
- \ctex_add_cmap:cn 472
- \ctex_add_cmap:n 42, 464, 469
- \ctex_add_cmap:Nn 474, 479
- \ctex_add_to_selectfont:n . 76, 1958, 1979, 2007, 2080
- \ctex_after_end_preamble:n 34, 109, 130, 4725
- \g__ctex_after_end_preamble_hook_tl 4726, 4730, 4738
- \ctex_appto_cmd:NnnTF 132, 495, 1620, 4795
- \ctex_assign_heading_name:nn ... 85, 2328, 2365, 2784
- __ctex_assign_heading_name:nnn 85, 2367, 2368
- \ctex_at_end:n 35, 135, 136, 293, 356, 1952, 3694
- \ctex_at_end_package:nn
130, 131, 450, 498, 765, 776, 1815, 1817, 1835,
2191, 2400, 2868, 2909, 2911, 2933, 2963, 3073, 3610, 4739
- \ctex_at_end_preamble:n
130, 361, 584, 591, 1274, 1646, 4723
- \ctex_auto_ignorespaces: 46, 599, 608, 2017
- \l__ctex_autoindent_tl
35, 132, 188, 193, 198, 2054, 2058, 2061, 2130, 2133
- __ctex_calc_bounds:nn 4564, 4565, 4569
- __ctex_ccglue: 1939, 1952
- \l__ctex_ccglue_skip 1923, 1927,
1930, 1933, 1935, 1943, 1947, 2104, 2115, 2124, 2160, 2168
- \ctex_CJK_input:n 44, 542, 557
- \c__ctex_class_tl
84, 86, 409, 414, 419, 424, 3328, 3334, 3336, 3337, 3339
- __ctex_clear_fntef_color:n ... 1844, 1851, 1855, 1870
- \c__ctex_cmap_encoding_seq 42, 456, 457, 463
- \l__ctex_ctexcap_options_clist 4708, 4709, 4718
- \ctex_declare_math_sizes:nnnn 107, 3472, 3506
- __ctex_def_heading_keys:n 84, 2322, 2391
- \ctex_default_pt:n 34, 113, 117
- __ctex_default_pt:w 119, 124
- \ctex_detected_platform: 71, 1741, 1759, 1774, 3635, 3644
- \l__ctex_encoding_tl 35, 128, 129, 221,
222, 505, 508, 586, 594, 1811, 2043, 2185, 3369, 3396, 3410
- \g__ctex_end_preamble_hook_tl 4724, 4728, 4737
- \c__ctex_engine_file_str 32, 50, 2047
- \c__ctex_engine_str 32, 43, 48, 51, 53
- \ctex_family_cmap:nn 42, 459, 467
- \ctex_family_cmap:xn 468
- \l__ctex_family_default_init_tl
70, 1711, 1731, 1733, 1738
- __ctex_family_default_wrap:n 1714, 1732, 1735
- \ctex_file_input:n 33, 77, 397,
661, 2047, 3339, 3374, 3379, 3384, 3385, 3400, 3405,
3411, 3412, 3516, 3518, 3636, 3640, 3648, 3694, 3819, 3820
- \ctex_file_wrapper:nnn 33, 62, 78, 81, 544
- __ctex_fix_varioref_label:n 99
- \ctex_fix_varioref_label:n 3057, 3059

`\g_ctex_font_size_flag` 36, 152,
 161, 162, 163, 377, 381, 403, 3386, 3387, 3413, 3414, 3515
`\c_ctex_font_size_prop` 106, 3427, 3438, 3444, 3482, 3509
`\c_ctex_font_size_seq` 3436, 3439, 3450
`\l_ctex_font_size_tl` 3427, 3428, 3474, 3475, 3509, 3510
`\ctex_fontset_error:n`
 110, 3627, 3926, 3960, 3994, 4039, 4071, 4074, 4117, 4176
`\g_ctex_fontset_tl` 225, 1743, 1761, 1786, 1793, 1794,
 3633, 3636, 3639, 3640, 3646, 3648, 3668, 3670, 3675, 3692
`\ctex_gadd_hook:cn` 130, 4743
`\ctex_gadd_hook:Nn` 130, 609, 4745, 4750
`__ctex_get_font_sizes:Nn` 107, 3474, 3477
`\ctex_get_macro_meaning:N` 4882
`\ctex_get_macro_meaning:NTF` 133, 4904, 4960
`__ctex_get_macro_meaning:w` 133, 4884, 4887
`\l_ctex_heading_bool` 304, 3324
`\c_ctex_headings_seq`
 83, 2282, 2284, 2285, 2286, 2289, 2388, 3056
`__ctex_hookto_cmd:Nnnnw` 135, 4792, 4797, 4957
`__ctex_hookto_cmd_parameter:Nnnnw` .. 136, 4964, 4984
`__ctex_hookto_cmd_parameterless:Nnnnw`
 135, 4963, 4969
`\ctex_hypersetup:n`
 ... 72, 72, 359, 1801, 1805, 1810, 1813, 1828, 1831, 1834
`\ctex_if_autoindent_touched:F`
 35, 133, 189, 194, 199, 3366
`\ctex_if_ccglue_touched:` 1936
`\ctex_if_ccglue_touched:TF` 75, 2099
`\ctex_if_ccglue_touched_p:` 75
`\ctex_if_macosx:TF` 72, 1781, 1782, 1784, 1790
`\ctex_if_preamble:TF` 34, 108, 109, 3663
`\ctex_if_rescanable:NnTF` 134, 4902, 4936
`\ctex_ignorespaces_case:N` .. 46, 601, 2014, 2017, 2020
`__ctex_initial_heading:n` 83, 2292, 2390
`__ctex_left_hook_aux:nn` 4982
`\l_ctex_line_spread_fp`
 109, 181, 3388, 3389, 3415, 3416, 3601, 3603
`\l_ctex_line_stretch_tl` 2086, 2091, 2092, 2096
`\ctex_load_fontset:` 110, 3631, 3652, 3671, 3693
`\ctex_load_zhmap:nnnn`
 43, 512, 519, 3863, 3954, 4033, 4111, 4170
`\l_ctex_ltj_add_alternate_bool` 1160, 1172, 1190, 1197
`\ctex_ltj_add_font_features:n` 59, 1180, 1198
`\ctex_ltj_add_font_features:nn` 59, 1182, 1199
`\ctex_ltj_add_font_features:x` 1222
`\ctex_ltj_add_font_features:xn` 1181
`__ctex_ltj_alternate_cs:n`
 1037, 1040, 1041, 1042, 1075, 1076, 1078, 1163,
 1166, 1167, 1169, 1170, 1361, 1398, 1460, 1485, 1525, 1542
`\l_ctex_ltj_alternate_family_tl` 1423, 1426
`\l_ctex_ltj_alternate_options_clist`
 1376, 1377, 1388, 1392
`\l_ctex_ltj_alternate_seq`
 1005, 1345, 1346, 1354, 1360, 1368, 1616
`\l_ctex_ltj_base_CJKfamily_tl`
 1006, 1022, 1073, 1460, 1485
`\l_ctex_ltj_base_family_tl` 65,
 1038, 1068, 1074, 1404, 1445, 1504, 1523, 1528, 1529, 1541
`__ctex_ltj_change_encoding:` 55, 978, 1056, 1418
`\l_ctex_ltj_char_range_clist` 1004,
 1329, 1338, 1357, 1375, 1387, 1399, 1422, 1444, 1595, 1597
`\ctex_ltj_char_range_key:nn` 67, 1589, 1592
`__ctex_ltj_char_range_parse_feature:w`
 67, 1600, 1604
`\g_ctex_ltj_char_range_prop` ... 66, 1452, 1563, 1568
`\l_ctex_ltj_char_range_tl`
 1452, 1455, 1561, 1563, 1565, 1567, 1574
`__ctex_ltj_check_family:n` 56, 1032, 1359
`\ctex_ltj_clear_alternate_font:n` 1514, 1520
`\l_ctex_ltj_current_font_tl`
 50, 789, 793, 794, 806, 848, 853
`\l_ctex_ltj_current_shape_tl` 871, 876, 881, 882
`\ctex_ltj_declare_alternate_shape:nnnnnn`
 64, 1421, 1441, 1443
`\ctex_ltj_declare_char_range:n` 1554, 1556
`__ctex_ltj_declare_char_range:nn` 1555, 1557
`\ctex_ltj_declare_char_range:nn` 66, 1558, 1559
`\ctex_ltj_declare_char_range:x` 1551
`\ctex_ltj_def_char_range_key:n` 67, 1564, 1584
`\g_ctex_ltj_default_features_clist` 1026, 1245, 1246
`\ctex_ltj_ensure_default_family:` ... 60, 1275, 1728
`\ctex_ltj_extract_font:` 51, 821, 828
`__ctex_ltj_family_csname:n` ... 1036, 1053, 1070, 1097
`\g_ctex_ltj_family_font_name_prop`
 56, 1011, 1030, 1034, 1108, 1184, 1277, 1289
`\g_ctex_ltj_family_font_options_prop`
 56, 1012, 1031, 1187
`\ctex_ltj_family_if_exist:nTF` 57
`\ctex_ltj_family_if_exist:xN` 1092
`\ctex_ltj_family_if_exist:xNF` 1279
`\ctex_ltj_family_if_exist:xNT` 1305
`\ctex_ltj_family_if_exist:xNTF` 1083, 1284, 1302, 1523
`\g_ctex_ltj_family_int` . 1150, 1152, 1437, 1438, 1440
`\g_ctex_ltj_family_name_prop`
 56, 1029, 1038, 1060, 1094, 1404
`\l_ctex_ltj_family_tl`
 1085, 1090, 1147, 1148, 1163, 1167, 1170, 1181, 1517, 1518
`__ctex_ltj_family_unknown_warning:n`
 57, 1088, 1106, 1533
`\l_ctex_ltj_font_options_clist` 1008, 1013,
 1014, 1015, 1023, 1059, 1188, 1189, 1192, 1395, 1396, 1400
`\ctex_ltj_fontspec:nn` 58, 1144, 1157
`__ctex_ltj_fontspec:nnn` 1158, 1178
`__ctex_ltj_fontspec:xnn` 1151
`\ctex_ltj_fontspec:xx` 1156, 1191, 1217
`\g_ctex_ltj_fontspec_family_tl`
 1058, 1061, 1062, 1063, 1069, 1074, 1099
`\g_ctex_ltj_fontspec_prop` 1146, 1174, 1179
`\ctex_ltj_get_and_define_fonts:nN` 54, 921, 929
`\ctex_ltj_get_and_define_fonts_al:nN` 925, 928
`\ctex_ltj_get_and_define_fonts_ja:nN` 924, 930
`__ctex_ltj_gset_family_cs:nn` 56, 1016, 1051

- \ctex_ltj_if_alter_nate_shape_exist:n 862
- \ctex_ltj_if_alter_nate_shape_exist:nF 844
- \ctex_ltj_if_alter_nate_shape_exist:nT 831, 871
- \ctex_ltj_if_alter_nate_shape_exist:nTF 52
- \ctex_ltj_if_jfont:nTF 53, 55, 907, 918, 923
- \ctex_ltj_if_jfont_math:NTF 53, 913, 954
- __ctex_ltj_if_jfont_math:w 914, 917
- \l__ctex_ltj_jfm_tl ... 54, 964, 965, 967, 982, 984, 1027
- \c__ctex_ltj_math_fam_int 990, 991
- \c__ctex_ltj_math_family_tl
 - 1311, 1313, 1315, 1318, 1322
- \ctex_ltj_math_group_hook: 54, 771, 953, 959
- \c__ctex_ltj_math_tl
 - 985, 986, 988, 990, 1243, 1302, 1312, 1317, 1321
- __ctex_ltj_msg_def_family_map:n 1121, 1123
- __ctex_ltj_msg_family_map:n 1050, 1120, 1134
- __ctex_ltj_noembed_wrap:n 998, 1000
- __ctex_ltj_patch_external_font:w 52, 834, 867
- \ctex_ltj_pickup_font:
 - 50, 802, 816, 827, 891, 933, 935, 937
- __ctex_ltj_pop_fontname: 805, 814, 893
- __ctex_ltj_push_fontname:n 801, 809, 890
- \ctex_ltj_reset_alter_nate_font:n 1516, 1536
- \g__ctex_ltj_reset_alter_nate_prop
 - 1043, 1527, 1539, 1547
- \ctex_ltj_save_alter_nate_family:Nnnn
 - 63, 1386, 1406, 1409
- \ctex_ltj_save_alter_nate_seq:cn 1361
- \ctex_ltj_save_alter_nate_seq:Nn 62, 1366, 1371
- \ctex_ltj_save_alter_nate_seq:Nnnwn 62
- \ctex_ltj_save_alter_nate_seq:Nnnwnw 1369, 1372
- __ctex_ltj_save_alter_nate_shape:cn 1459, 1484
- __ctex_ltj_save_alter_nate_shape:Nnn 65, 1500, 1510
- \ctex_ltj_save_char_range:n 66, 1562, 1569
- \ctex_ltj_save_char_range:nn 1571, 1572
- __ctex_ltj_save_fontname:w 811, 815
- \ctex_ltj_select_alter_nate_font: 52, 869, 1982
- \ctex_ltj_select_font: 50, 787, 1981
- __ctex_ltj_select_font_aux: 50, 790, 796, 807
- \ctex_ltj_set_alter_nate_family:coon 1397
- __ctex_ltj_set_alter_nate_family:n 1064, 1071
- \ctex_ltj_set_alter_nate_family:nn .. 63, 1364, 1393
- \ctex_ltj_set_alter_nate_family:nnF .. 62, 1009, 1355
- \ctex_ltj_set_alter_nate_family:nnn
 - 63, 1405, 1413, 1415
- \ctex_ltj_set_alter_nate_family:Nnnnn ... 1402, 1408
- \ctex_ltj_set_alter_nate_seq:n 61, 1327, 1336
- \ctex_ltj_set_alter_nate_shape:n ... 1469, 1481, 1492
- \ctex_ltj_set_alter_nate_shape:nnN .. 65, 1454, 1488
- \ctex_ltj_set_alter_nate_shape:nnn .. 64, 1457, 1464
- \ctex_ltj_set_alter_nate_shape:nnnn 1466, 1467
- \ctex_ltj_set_alter_nate_shape:Nnnnnnn 64, 1444, 1448
- \ctex_ltj_set_family:nnn 55, 1001, 1021, 1175
- \ctex_ltj_set_family:xxx
 - 1020, 1206, 1213, 1228, 1234, 1239, 1243
- \ctex_ltj_subst_font: 51, 822, 842
- \ctex_ltj_swap_cs:NN 63, 1420, 1427
- \ctex_ltj_switch_family:n 1081, 1091, 1176, 1212
- \ctex_ltj_switch_family:x 1080, 1148
- __ctex_ltj_tmp:w 1334, 1351, 1429, 1431, 1432
- \l__ctex_ltj_tmp_seq 1342, 1343, 1346, 1353
- \l__ctex_ltj_tmp_tl .. 1007, 1008, 1034, 1045, 1048,
 - 1185, 1193, 1209, 1212, 1213, 1302, 1303, 1305, 1306,
 - 1340, 1341, 1342, 1379, 1381, 1382, 1389, 1599, 1607, 1616
- \l__ctex_ltj_tmpa_tl 1279, 1284
- __ctex_ltj_um_char:Nn 740, 745
- __ctex_ltj_um_char_aux:Nn 750, 764
- __ctex_ltj_um_char_aux:Nx 747
- __ctex_ltj_um_define_math_chars: 729, 767
- \g__ctex_ltj_unknown_family_seq ... 1110, 1112, 1117
- __ctex_ltj_update_family_uid:N 64, 1014, 1395, 1435
- \ctex_ltj_update_mathfont: 61, 1297, 1300
- \ctex_ltj_update_mathfont:n 1303, 1306, 1309
- __ctex_ltj_use_global_options:N
 - 1015, 1024, 1377, 1396
- \ctex_ltj_use_math_group:Nn 54, 770, 947, 960
- \c__ctex_macosx_file_str 1792, 1796
- \ctex_make_spa:nn 126, 4536, 4544
- \ctex_mono_jfm:n 54, 961, 968
- \ctex_mono_jfm:o 969, 2037
- \l__ctex_msyh_suffix_tl
 - 3823, 3824, 3828, 3844, 3846, 3880
- \ctex_package_end_hook:n 130, 4751, 4756
- \ctex_package_end_hook:o 130, 4760
- \l__ctex_parameter_str ... 133, 4876, 4894, 4915, 4962
- \l__ctex_parameter_tl 4915, 4920, 4931
- \ctex_parse_name:NN 132, 4806, 4810
- \ctex_parse_name:NNn 4815, 4874
- __ctex_parse_name:nNNnN 4827, 4836
- \ctex_parse_name:NNx 4811
- __ctex_parse_name:w 4849, 4855, 4864
- \ctex_patch_boot:NNnnTF
 - 132, 4767, 4772, 4777, 4792, 4797, 4800
- \ctex_patch_cmd:Nnn
 - 131, 1647, 1687, 2879, 2885, 2891, 2944, 2948,
 - 2954, 2958, 2966, 2969, 2974, 2978, 3622, 3623, 3624, 4775
- __ctex_patch_cmd:Nnnnnw .. 134, 4767, 4772, 4777, 4933
- \ctex_patch_cmd_all:NnnnTF 131, 4770
- \ctex_patch_cmd_once:NnnnTF 131, 4765
- \ctex_patch_failure:N 501, 528, 532, 536, 540,
 - 1261, 1265, 1269, 1273, 1624, 1971, 2929, 2939, 4784, 4786
- __ctex_patch_false:w 4803, 4809, 4948, 4950, 4967, 4993
- __ctex_patch_rebuild:Nn
 - 134, 4906, 4912, 4945, 4986, 4990
- __ctex_patch_replace:nnnTF 135, 4940, 4952
- __ctex_patch_rescan:NNn 4914, 4915, 4916, 4924
- __ctex_patch_true:w 4802, 4808, 4946, 4979, 4991
- \ctex_plane_to_utfxvibe:Nn 44, 559, 577
- \l__ctex_prefix_str .. 133, 4875, 4893, 4914, 4971, 4972
- \l__ctex_prefix_tl 4914, 4919, 4930
- \ctex_preto_cmd:NnnTF 131, 525, 529,
 - 533, 537, 1258, 1262, 1266, 1270, 1965, 2926, 2935, 4790

- \l_ctex_punct_bounds_clist ... 4550, 4556, 4562, 4568
- \c_ctex_punct_family_clist 612, 622
- \l_ctex_punct_font 4548, 4549
- \ctex_punct_map_bfseries:nn
..... 47, 632, 3848, 3860, 3866, 3951,
3957, 4029, 4030, 4036, 4106, 4107, 4108, 4114, 4167, 4173
- \ctex_punct_map_family:nn ... 46, 627, 3841, 3847,
3857, 3858, 3865, 3947, 3948, 3949, 3956, 4025, 4026,
4027, 4035, 4102, 4103, 4104, 4113, 4163, 4164, 4165, 4172
- \ctex_punct_map_itshape:nn 47, 649,
3859, 3867, 3950, 3958, 4028, 4037, 4105, 4115, 4166, 4174
- \ctex_punct_map_series:nnn 636, 637, 640
- \c_ctex_punct_seq 4527, 4528, 4551
- \ctex_punct_set:n 46, 610,
3836, 3864, 3946, 3955, 4024, 4034, 4101, 4112, 4162, 4171
- \ctex_punct_space:nn 47, 658, 660
- \l_ctex_punct_tl
..... 280, 581, 582, 669, 969, 2034, 2035, 2036, 2037
- _ctex_rebuild_cmd:w ... 4906, 4907, 4911, 4986, 4987
- \l_ctex_replacement_str
..... 133, 4877, 4895, 4916, 4954, 4955, 4989
- \l_ctex_replacement_tl 4916, 4921, 4932
- _ctex_right_hook_aux:nn 4983
- _ctex_save_bounds:n 4552, 4560
- _ctex_save_cmap:Nn 476, 480
- _ctex_save_font_size:nn 106, 3440, 3471
- _ctex_save_selectfont: 1964, 1969
- \ctex_scheme_input:n 33, 79, 93
- \ctex_scheme_input:o 3075, 3341
- \l_ctex_scheme_tl 315, 3075, 3341
- \g_ctex_section_depth_flag
..... 34, 95, 309, 311, 3263, 3279, 3295
- \c_ctex_section_headings_seq
..... 83, 2278, 2279, 2284, 2767, 2826
- \ctex_select_size:.. 2055, 2062, 2065, 2066, 2087, 2141
- \ctex_set_default_ccwd:Nn ... 34, 110, 198, 2061, 2086
- \ctex_set_font_size:Nnn 107, 3507, 3522,
3530, 3541, 3552, 3553, 3554, 3555, 3556, 3557, 3558,
3561, 3569, 3580, 3591, 3592, 3593, 3594, 3595, 3596, 3597
- _ctex_set_font_size:nnNn 3510, 3513
- \ctex_set_ignorespaces: 46, 603, 605, 607, 609
- \ctex_set_upfamily:nnn
..... 69, 1650, 1662, 3883, 3884, 3885, 3886, 3887,
3888, 3890, 3970, 3971, 3972, 3973, 4049, 4050, 4051,
4052, 4126, 4127, 4128, 4129, 4130, 4132, 4185, 4186, 4187
- \ctex_set_upfonts:nnnnnn
..... 69, 1677, 1684, 3878, 3966, 4045, 4122, 4181
- \ctex_set_upmap:nnn 69,
1663, 1676, 1679, 1680, 1681, 1682, 3889, 3891, 4131, 4133
- \ctex_set_zhmap:n 42, 447, 453, 517, 1665
- \g_ctex_spa_iow 4538, 4541, 4543, 4553
- \c_ctex_std_class_tl 103, 3315, 3327, 3328, 3394, 3417
- \g_ctex_std_options_clist
..... 35, 138, 382, 394, 404, 406, 410, 415, 420, 425
- \ctex_titleps_hook: 95, 2877, 2910, 2911
- _ctex_titlesec_format:n 2822, 2830, 2833
- _ctex_titlesec_format:Nn 2812, 2836, 2838
- _ctex_titlesec_format_explicit:Nn 2813, 2848
- \ctex_titlesec_hook: 93, 2808, 2872
- _ctex_titlesec_hook:n 2822, 2826, 2828
- _ctex_titlesec_spacing:Nn 2831, 2858
- _ctex_titlesec_spacing:nnnnnn 2859, 2860
- _ctex_tmp:w 4813, 4868, 4879, 4900
- \l_ctex_tmp_box 33, 61, 1908, 1909, 2914, 2918
- \l_ctex_tmp_dim 33, 60,
2096, 2097, 2102, 2105, 2111, 2150, 2152, 2154, 2163, 2165
- \l_ctex_tmp_int 33, 59, 562,
564, 565, 567, 571, 573, 2110, 2112, 2113, 2119, 2120,
2152, 2153, 2154, 2157, 2158, 2163, 2165, 2766, 2769, 2773
- \l_ctex_tmp_tl . 33, 58, 366, 373, 388, 389, 482, 483,
561, 563, 2324, 2377, 2386, 2396, 3393, 3394, 3482, 3483
- \ctex_update_ccglue: 75, 1919, 2106, 2125, 2169
- \ctex_update_ccwd: 75, 1905, 2100, 2177
- \ctex_update_default_family:
..... 45, 70, 584, 1274, 1646, 1709
- \ctex_update_em_unit: 76, 1954, 2095, 2145
- \ctex_update_parindent: 80, 2073, 2128, 2170
- \ctex_update_size: 79, 2067, 2068, 2081
- \ctex_update_stretch: 79, 2072, 2093
- \ctex_update_xkanjiskip: 77, 1988, 2007
- \ctex_update_ziju: 80, 2075, 2143
- \c_ctex_upper_case_file_str 1780, 1788
- \ctex_varioref_hook: 99, 3054, 3065, 3073
- \c_ctex_version_tl 3, 6, 7, 8, 9, 20, 23
- _ctex_write_family:nn 4540, 4545
- \l_ctex_xkanjiskip_skip
..... 77, 1991, 1992, 1994, 1995, 1996, 2003, 2004
- \l_ctex_xkanjiskip_tl 1994, 1999, 2000
- \c_ctex_zero_tl 2070, 2077, 2079
- \ctex_zhmap_case:nnn
..... 35, 134, 261, 266, 271, 3834, 3928, 3996, 4089, 4153
- \g_ctex_zhmCJK_bool ... 35, 131, 260, 265, 270, 504, 585
- \ctex_zihao:n 106, 3424, 3425
- \ctex_ziju:n 2137, 2138
- \l_ctex_ziju_dim 2146, 2147, 2151, 2162, 2164, 2168, 2172
- \l_ctex_ziju_tl 2070, 2078, 2079, 2140, 2146
- \ctexbibname 2250, 2270
- \CTEXcounter 2190
- \CTEXdigits 24, 82, 2202
- \CTEXfilltwosides 29, 1878, 1888
- \CTEXindent 29, 81, 2173, 2175
- \CTEXnoindent 29, 81, 2179, 2181
- \CTEXnumber 24, 82, 2200
- \CTEXoptions 29, 41, 439, 441
- \ctexset 5, 41, 430, 436, 442
- \CTEXsetfont 29, 78, 2065
- \CTEXsetup 29, 41, 433, 435
- \CTEXsout 29
- \ctexspadef 47, 660, 4555
- \CTEXthechapter 16, 2565, 2568, 2956, 2976, 2977
- \CTEXtheparagraph 16
- \CTEXthepart 16, 2456, 2486

`\CTEXthesection` 16, 2946, 2960, 2968, 2980
`\CTEXthesubparagraph` 16
`\CTEXthesubsection` 16, 2950, 2971
`\CTEXthesubsubsection` 16
`\CTEXunderdblline` 29
`\CTEXunderdot` 29
`\CTEXunderline` 29
`\CTEXunderwave` 29
`\CTEXxout` 29
`\CurrentOption` 394

D

`declarecharrange` 25, 66
`\DeclareFontEncoding` 971
`\DeclareFontFamily`
 ... 43, 495, 501, 980, 3712, 3713, 3714, 3715, 3716, 3717
`\DeclareFontShape`
 ... 981, 983, 1654, 1655, 1658, 1659, 3728, 3729,
 3730, 3731, 3732, 3733, 3734, 3735, 3736, 3739, 3740,
 3741, 3742, 3743, 3744, 3745, 3746, 3747, 3751, 3752,
 3753, 3754, 3755, 3756, 3760, 3761, 3762, 3763, 3764,
 3765, 3766, 3767, 3768, 3771, 3772, 3773, 3774, 3775,
 3776, 3777, 3778, 3779, 3783, 3784, 3785, 3786, 3790,
 3791, 3792, 3793, 3794, 3795, 3796, 3797, 3798, 3801,
 3802, 3803, 3804, 3805, 3806, 3807, 3808, 3809, 3813, 3814
`\DeclareFontSubstitution` 974
`\DeclareKanjiFamily`
 ... 1652, 1653, 3719, 3720, 3721, 3722, 3723, 3724
`\DeclareMathSizes` 3475
`\DeclareSymbolFont` 986, 1312
`\def` 2451, 2481, 2508,
 2522, 2532, 2559, 2588, 2614, 2633, 2656, 2659, 2692,
 2709, 2722, 2723, 2727, 3535, 3546, 3574, 3585, 4219, 4255
`\defaultCJKfontfeatures` 59, 1244, 1252
`\defbeamertemplate` 2986, 3004, 3022
`\defbeamertemplatealias` 3040, 3041, 3042
dim commands:
 `\dim_abs:n` 2164
 `\dim_add:Nn` 2147
 `\dim_compare:nNnF` 2132, 2155
 `\dim_compare:nNnTF` 112, 2097, 2148
 `\dim_eval:n` 2117, 2163
 `\dim_max:nn` 2917
 `\dim_min:nn` 2164
 `\dim_new:N` 60, 1918, 2172
 `\dim_set:Nn` 1909, 1955,
 1956, 1957, 2096, 2102, 2133, 2146, 2150, 2177, 2915, 3605
 `\dim_sub:Nn` 2154
 `\dim_to_decimal:n` 34, 3446, 3447, 3484
 `\dim_to_decimal_in_unit:nn` 4575
 `\dim_use:N` 120, 3605
 `\dim_zero:N` 2183
`\dimexpr` 2466, 2511, 2596, 2622, 2992, 3010, 3028
`\directlua` 707, 708
`\document` 4731, 4734

E

`\edef` 4238, 4239
`\else` 1648, 1649, 2422, 2431,
 2439, 2457, 2461, 2487, 2491, 2548, 2555, 2569, 2573,
 2576, 2584, 2644, 2649, 2662, 2666, 2684, 2691, 2700,
 2721, 2735, 2748, 4250, 4251, 4297, 4336, 4367, 4426, 4489
else commands:
 `\else:` 507, 865, 911, 1940, 3609
`\em` 68, 1647
`\eminnershape` 1649
`\end` 3000, 3018, 3036, 4620
`\endCJKfilltwosides` 1884
`\endcsname` 2657, 2658, 2663,
 2679, 2680, 2682, 2685, 2686, 2687, 2696, 2697, 2698,
 2701, 2702, 2703, 4245, 4247, 4248, 4249, 4252, 4254, 4258
`\endCTEXfilltwosides` 1884, 1889
`\endgroup` 2681, 2720, 2743, 3002, 3020, 3038, 4238, 4246, 4261
`\endinput` 4243
`\endlinechar` 4217, 4218
etex commands:
 `\etex_dimexpr:D` 120, 2111
 `\etex_glueexpr:D` 2103
`\everypar` 2648, 2738, 2750
`\EverySelectfont` 1697, 1973, 1976
exp commands:
 `\exp_after:wN`
 119, 818, 834, 836, 911, 914, 2859, 3428, 3475, 3510, 4884
 `\exp_args:cc` 3050
 `\exp_args:Nc` 2831, 2836, 4838
 `\exp_args:Nnc` 99
 `\exp_args:NNf` 1607
 `\exp_args:NNNo` 3604
 `\exp_args:Nno` 3062
 `\exp_args:No` 563, 1789, 2855, 4552, 4900
 `\exp_args:Nx` 293, 591, 665, 2137, 3424, 3603
 `\exp_last_unbraced:Nf` 4849
 `\exp_not:c` 853, 1873,
 1875, 1902, 2298, 2299, 2300, 2305, 2307, 2309, 2310,
 2313, 2314, 2317, 2329, 2330, 2331, 2332, 2333, 2334,
 2335, 2336, 2337, 2338, 2339, 2341, 2342, 2345, 2347,
 2349, 2351, 2353, 2355, 2774, 2775, 2776, 2777, 4830, 4831
 `\exp_not:N` 124, 125, 486,
 593, 594, 595, 780, 781, 783, 889, 974, 1060, 1061, 1062,
 1063, 1386, 1507, 1719, 1720, 1721, 1735, 1874, 2308,
 2772, 2777, 2843, 2844, 2853, 2854, 3444, 3646, 4829, 4919
 `\exp_not:n` 69, 1057, 1058, 1714, 2862, 4847, 4977
 `\exp_not:o` 388,
 782, 784, 1059, 1192, 1193, 1387, 1388, 1389, 1413,
 1507, 1736, 1961, 2396, 3051, 4734, 4919, 4920, 4921, 4976
 `\exp_not:v` 4710
`\expandafter`
 4220, 4221, 4245, 4246, 4247, 4248, 4249, 4254, 4258, 4259
`\ExplSyntaxOff` 66, 72, 495, 525, 529, 533, 537,
 689, 1258, 1262, 1266, 1270, 1965, 2926, 2985, 4581, 4780
`\ExplSyntaxOn` 71, 718, 3043, 4520

F

false commands:

\c_false_bool 2765
 \familydefault 1717
 fancyhdr 28, 40
 \fangsong 4203
 \fi 1701, 2410,
 2416, 2424, 2433, 2441, 2446, 2460, 2463, 2470, 2490,
 2493, 2501, 2541, 2542, 2545, 2548, 2557, 2572, 2575,
 2578, 2587, 2593, 2603, 2604, 2605, 2619, 2634, 2640,
 2646, 2652, 2669, 2670, 2688, 2689, 2704, 2705, 2707,
 2725, 2751, 2752, 2946, 2950, 2956, 2960, 2968, 2971,
 2976, 2977, 2980, 2996, 3014, 3032, 4250, 4251, 4260, 4513

fi commands:

\fi: 407, 541, 578, 597, 865, 911, 1940, 2951, 2961, 3519, 3609
 \figurename 2231
 figurename 11

file commands:

\g_file_current_name_tl 681, 1789
 \file_if_exist:nF 3825, 3827
 \file_if_exist:nTF 48, 1776, 1778, 1780, 1792, 3639, 3818
 \file_if_exist_input:nF 86
 \file_input:n 78, 87, 89, 517, 550

five commands:

\c_five 3447
 fntef 29, 40
 \font 3712, 3713, 3714, 3715, 3716, 3717
 fontset 7, 37, 111
 \fontsize 3428

fontspec commands:

\g_fontspec_encoding_tl 979
 \l_fontspec_family_tl 1446
 __fontspec_namewrap:n 55, 998
 \fontspec_set_family:Nnn 1058, 1423
 \footnotesep 109, 3605
 \footnotesize 3541, 3580, 3604

four commands:

\c_four 311, 571, 573

fp commands:

\fp_compare:nNnF 3601
 \fp_compare:nNnT 3388, 3415
 \fp_eval:n 2140, 4571
 \fp_set:Nn 3389, 3416
 \fp_use:N 2077, 3603

G

GBK 6, 37
 \gdef 1692, 2796, 2797, 2798, 2799, 2802, 2803, 2804, 2805
 \global 2551, 2737, 2740
 \globaljfont 836
 \glueexpr 2419, 2443, 2477, 2518,
 2534, 2590, 2612, 2616, 2630, 2651, 2676, 2693, 2715,
 2722, 2723, 2733, 2747, 2991, 3001, 3009, 3019, 3027, 3037

group commands:

\group_begin:
 490, 723, 731, 798, 820, 846, 888, 915, 1003,

1055, 1332, 1417, 1502, 1522, 1538, 1713, 2304, 2312,
 2375, 2875, 2921, 3604, 4547, 4728, 4804, 4812, 4823, 4878
 \group_end: 494,
 503, 727, 743, 803, 824, 859, 892, 919, 1018, 1065,
 1352, 1424, 1508, 1534, 1545, 1725, 2315, 2319, 2395,
 2898, 2925, 3604, 4558, 4728, 4826, 4873, 4901, 4935, 4959
 \group_insert_after:N 856, 857
 \gtfamily 1648

H

hbox commands:

\hbox_set:Nn 1908, 2914
 heading 9, 39
 \heiti 4202
 \hskip ... 2676, 2693, 2715, 2722, 2723, 2746, 2747, 2967, 2970
 \hspace 2455, 2456, 2485, 2486, 2568
 \Huge 2502, 2526, 2607, 2625, 3117, 3144, 3558, 3597
 \huge 2471,
 2499, 2513, 2601, 3100, 3116, 3120, 3142, 3149, 3557, 3596
 hyperref 29, 40
 \hypersetup 358, 359, 1802
 \hyphenchar 3712, 3713, 3714, 3715, 3716, 3717

I

if commands:

\if_bool:N 504, 585
 \if_case:w 403, 3515
 \if_cs_exist:N 558
 \if_false: 107
 \if_meaning:w 1939
 \if_true: 106
 \ifctexpdf 34, 106, 107
 \ifdim 2406, 2638, 2672, 2711, 2729
 \ifincsname 3609
 \IfNoValueF 437, 443
 \IfNoValueTF 1210, 1471, 1576, 1609, 1611, 2371
 \ifnum 1699,
 2452, 2467, 2482, 2498, 2560, 2598, 2660, 2684, 2700, 4250
 \ifodd 2420, 2444,
 2453, 2467, 2483, 2498, 2553, 2562, 2591, 2600, 2617,
 2642, 2663, 2673, 2685, 2701, 2712, 2730, 2946, 2950,
 2956, 2960, 2968, 2971, 2976, 2977, 2980, 2994, 3012, 3030
 \ifx 4248, 4249, 4254
 \ifzhmappdf 4266, 4333, 4364, 4395, 4462
 \ignorespaces 2753
 \immediate 4257
 indent 28, 37
 \indexname 2234
 indexname 12
 \input 4519, 4584
 \insertcontinuationtext 2243, 2252, 2259
 \insertpart 2998, 2999
 \insertromanpartnumber 2989, 3087
 \insertsection 3016, 3017
 \insertsectionnumber 3007, 3179
 \insertsubsection 3034, 3035
 \insertsubsectionnumber 3025, 3207

int commands:

<code>\int_compare:nNnF</code>	2157, 3386, 3413
<code>\int_compare:nNnTF</code>	564, 2113, 3263, 3279, 3295
<code>\int_const:Nn</code>	990
<code>\int_div_truncate:nn</code>	79, 571
<code>\int_eval:n</code>	1472, 1474, 1475, 1577, 1579, 1580
<code>\int_from_hex:n</code>	563
<code>\int_gincr:N</code>	1150, 1437
<code>\int_incr:N</code>	2769
<code>\int_mod:nn</code>	573
<code>\int_new:N</code>	59, 1440
<code>\int_set:Nn</code>	554, 562, 2110, 2152
<code>\int_set_eq:NN</code>	548
<code>\int_sub:Nn</code>	567, 2158
<code>\int_to_Hex:n</code>	565, 570, 572
<code>\int_use:N</code>	487, 554, 903, 1152, 1438, 2773, 4552
<code>\int_zero:N</code>	2766

<code>\interlinepenalty</code>	2464, 2495, 2509, 2523, 2606, 2624, 2677, 2678, 2716, 2717
--------------------------------------	--

iow commands:

<code>\iow_close:N</code>	4541
<code>\iow_indent:n</code>	4715
<code>\iow_new:N</code>	4543
<code>\iow_now:Nx</code>	4553
<code>\iow_open:Nn</code>	4538
<code>\itemsep</code>	3538, 3549, 3577, 3588
<code>\itshape</code>	1648, 1649

J

<code>\jfam</code>	955, 991
--------------------------	----------

K

<code>\kaishu</code>	4204
<code>\kanjifamily</code>	1637, 1638, 1639, 1642
<code>\kanjifamilydefault</code>	1645

kernel commands:

<code>\l_kernel_expl_bool</code>	70
--	----

keys commands:

<code>\l_keys_choice_tl</code>	294
<code>\keys_define:nn</code>	155, 388, 391, 995, 1325, 1434, 1511, 1549, 1588, 2009, 2030, 2048, 2082, 2205, 2226, 2254, 2264, 2267, 2270, 2396, 2780, 3612, 3659
<code>\keys_if_exist:nnF</code>	1586
<code>\l_keys_key_tl</code>	141, 143, 145
<code>\keys_set:nn</code>	27, 171, 177, 233, 242, 251, 277, 287, 294, 301, 331, 337, 430, 437, 443, 3078, 3132, 3156, 3187, 3221, 3237, 3250, 3265, 3273, 3281, 3289, 3296, 3297, 3299, 3365, 3367, 3701
<code>\keys_set_known:nn</code>	3344
<code>\keys_set_known:nnN</code>	1007

L

<code>\labelformat</code>	3062
<code>\languagealias</code>	3373, 3378, 3399, 3404
<code>\LARGE</code>	3556, 3595
<code>\Large</code>	2468, 3098, 3104, 3166, 3170, 3555, 3594
<code>\large</code>	3191, 3554, 3593

<code>\lastbox</code>	2741
<code>\leavevmode</code>	2416, 2634

left commands:

<code>\c_left_brace_str</code>	4870
<code>\leftmargin</code>	3535, 3546, 3574, 3585
<code>\leftmargini</code>	3535, 3546, 3574, 3585
<code>\let</code>	1705, 2661, 2668, 3528, 3567, 4241, 4247
<code>\linespread</code>	3603
<code>linespread</code>	10, 36
<code>linestretch</code>	10, 79
<code>\linewidth</code>	2111, 2119, 2151
<code>\lishu</code>	4206
<code>\listfigurename</code>	2229
<code>listfigurename</code>	11
<code>\listtablename</code>	2230
<code>listtablename</code>	11
<code>\LoadClass</code>	411, 416, 421, 426
<code>\long</code>	4255
<code>\lstlistingname</code>	781, 782
<code>\lstlistlistingname</code>	783, 784
<code>\ltjdefcharrange</code>	690, 691, 692, 694, 698, 699, 702, 705
<code>LTJFONTUID</code>	64
<code>\ltjgetparameter</code>	1913, 1943, 1991, 2005
<code>\ltjsetkanjiskip</code>	1930
<code>\ltjsetmathletter</code>	748
<code>\ltjsetparameter</code>	706, 712, 1627
<code>\ltjsetxkanjiskip</code>	1995

lua commands:

<code>\lua_escape_x:n</code>	864, 876, 903, 910, 1462, 1477, 1478, 1495, 1496, 1505
<code>\lua_now_x:n</code>	724, 758, 838, 864, 868, 873, 878, 900, 909, 977, 1483, 1507, 1745

M

<code>\MAKESPA</code>	4544, 4586
-----------------------------	------------

mark commands:

<code>\q_mark</code>	4885, 4888
<code>\markboth</code>	2473, 2494
<code>\mathalpha</code>	736
<code>\mathbin</code>	736
<code>\mathfence</code>	737
<code>\mathgroup</code>	955
<code>\mathop</code>	737
<code>\mathord</code>	736
<code>\mathpunct</code>	737
<code>\mathrel</code>	736

max commands:

<code>\c_max_dim</code>	2097
<code>\maxdimen</code>	2406
<code>\mcfamily</code>	1648
<code>\mddefault</code>	975, 981, 987, 1313, 1318

minus commands:

<code>\c_minus_one</code>	152, 548, 3386, 3413
---------------------------------	----------------------

mode commands:

<code>\mode_if_math:T</code>	949
<code>\mode_if_math:TF</code>	754

msg commands:

\msg_critical:nnn 3628
 \msg_critical:nnx 53, 681
 \msg_error:nn 3679
 \msg_error:nnn .. 19, 22, 38, 41, 672, 685, 688, 3429, 3511
 \msg_error:nnx 2221
 \msg_error:nnxx 3645, 3674
 \msg_fatal:nn 4525
 \msg_new:nn 4523
 \msg_new:nnn 140, 142, 144, 146, 148, 675,
 1049, 1118, 1200, 1860, 2247, 3318, 3629, 3688, 4711, 4788
 \msg_new:nnnn 12, 30, 45, 99, 2223, 3431, 3653, 3682
 \msg_warning:nn 1195, 1864, 2027, 2038, 2269
 \msg_warning:nnn 169, 175, 203, 213, 228, 237,
 246, 275, 285, 299, 329, 335, 343, 348, 354, 1113, 3616, 3666
 \msg_warning:nnnn 435, 441, 1873, 1880, 2175, 2181
 \msg_warning:nnx 3337, 4717, 4787
 \msg_warning:nnxx 1045

N

nan commands:

\c_nan_fp 182
 \newCJKfontfamily 59, 1207
 \NewDocumentCommand ... 430, 433, 439, 1079, 1205, 1207,
 1215, 1220, 1226, 1232, 1237, 1242, 1244, 1372, 1464,
 1569, 1604, 1640, 2065, 2136, 2173, 2179, 2200, 2202,
 2365, 3423, 3608, 4201, 4202, 4203, 4204, 4206, 4207, 4209
 \newfontfeature 992, 993, 994
 \newpage 2535, 2540
 \ngostype 44, 1763
 \nobreak 2468,
 2475, 2516, 2601, 2607, 2610, 2625, 2628, 2731, 3099, 3143
 \nobreakspace 2468, 2499
 nocap 28
 \noexpand 4239, 4241, 4242, 4243
 nofonts 28
 noindent 28
 \nointerlineskip 2409
 nopunct 28
 \normalem 1843
 \normalfont 537, 539, 540, 1229,
 1235, 1240, 1270, 1272, 1273, 1296, 2465, 2466, 2496,
 2497, 2510, 2511, 2524, 2525, 2595, 2596, 2621, 2622, 2777
 \normalsize 3224, 3240, 3253, 3522, 3561, 3599, 3607
 nospace 28
 nozhmap 28
 \null 2442, 2538
 \numberline 96,
 2567, 2568, 2686, 2687, 2702, 2703, 2926, 2929, 2935, 2939

O

one commands:

\c_one 162, 1465, 1570, 2165, 2366, 4564
 \onecolumn 2437

or commands:

\or: 405, 3517

P

\pagegoal 2406
 \pagestyle 3364
 \par 2405, 2417, 2468, 2499, 2502, 2513,
 2526, 2601, 2607, 2625, 2635, 2731, 2990, 2991, 2998,
 3001, 3008, 3009, 3016, 3019, 3026, 3027, 3034, 3037,
 3080, 3099, 3110, 3127, 3135, 3143, 3182, 3184, 3215, 3217
 paragraph/afterskip 30
 paragraph/beforeskip 30
 paragraph/numbering 15
 paragraph/runin 20
 \parindent
 2132, 2133, 2155, 2176, 2177, 2182, 2183, 2465, 2466,
 2510, 2511, 2595, 2596, 2621, 2622, 2992, 3010, 3028, 3297
 \parsep 3537, 3538, 3548, 3549, 3576, 3577, 3587, 3588
 \parskip 2408
 \part 86, 2415, 2428
 part/beforeskip 30
 part/fixbeforeskip 20, 85
 part/numbering 15
 part/pagestyle 21, 85
 \partname 2468, 2499, 2989, 3082
 \PassOptionsToClass 410, 415, 420, 425
 \PassOptionsToPackage 505, 1806, 2185, 4719
 \pdfmapline 4267,
 4268, 4269, 4270, 4271, 4272, 4273, 4275, 4276, 4277,
 4278, 4279, 4280, 4281, 4283, 4284, 4285, 4286, 4287,
 4288, 4290, 4291, 4292, 4293, 4294, 4295, 4396, 4397,
 4398, 4399, 4400, 4401, 4402, 4404, 4405, 4406, 4407,
 4408, 4409, 4410, 4412, 4413, 4414, 4415, 4416, 4417,
 4419, 4420, 4421, 4422, 4423, 4424, 4463, 4464, 4465,
 4466, 4467, 4468, 4470, 4471, 4472, 4473, 4474, 4475,
 4477, 4478, 4479, 4480, 4481, 4483, 4484, 4485, 4486, 4487
 \pdfoutput 4250
 \pdfstringdefDisableCommands 3611

pdftex commands:

\pdftex_pdffontattr:D 486
 \pdftex_pdflastobj:D 487
 \pdftex_pdfobj:D 483
 \phantomsection 2401

prg commands:

\prg_do_nothing:
 959, 1411, 1503, 1889, 1963, 2014, 2763, 4911
 \prg_new_conditional:Npnn 862, 1936
 \prg_new_protected_conditional:Npnn 1092, 4881
 \prg_return_false: ... 865, 1102, 1940, 1944, 1948, 4891
 \prg_return_true: 865, 1095, 1100, 1940, 1944, 1948, 4896
 \primitive 4620
 \ProcessKeysOptions 400
 \proofname 2236
 proofname 12

prop commands:

\prop_get:NnN 1187
 \prop_get:NnNT 1404
 \prop_get:NnNTF 1105, 1146, 1184, 1452, 3427, 3482, 3509
 \prop_get:NxNTF 1094

`\prop_gpop:NnNT` 1034, 1038, 1539
`\prop_gput:Nnn` 1011, 1174, 3444
`\prop_gput:Nno` 1012, 1060, 1527, 1563
`\prop_gremove:Nn` 1043
`\prop_if_empty:Nf` 1108, 1277
`\prop_map_break:n` 1291
`\prop_map_inline:Nn` 1289
`\prop_new:N` 1029, 1030, 1031, 1179, 1547, 1568, 3438
`\protect` .. 2567, 2568, 2580, 2581, 2686, 2687, 2702, 2703,
 2880, 2882, 2886, 2888, 2892, 2894, 2902, 2907, 4842, 4843
`\ProvidesFile` 4255, 4263, 4330, 4361, 4392, 4459
`\providetranslation`
 4625, 4626, 4627, 4628, 4629, 4630, 4631,
 4632, 4633, 4634, 4635, 4636, 4637, 4638, 4639, 4640,
 4641, 4642, 4643, 4644, 4645, 4646, 4647, 4648, 4649,
 4650, 4651, 4652, 4653, 4654, 4655, 4656, 4657, 4658,
 4659, 4660, 4661, 4662, 4663, 4664, 4665, 4666, 4667,
 4668, 4669, 4670, 4671, 4672, 4673, 4674, 4675, 4676,
 4677, 4678, 4679, 4680, 4681, 4682, 4683, 4684, 4685,
 4686, 4687, 4688, 4689, 4690, 4691, 4692, 4693, 4694,
 4695, 4696, 4697, 4698, 4699, 4700, 4701, 4702, 4703, 4704
ptex commands:
`\ptex_kanjiskip:D` 1916, 1933, 1947
`\ptex_xkanjiskip:D` 1992, 1996, 2006
`punct` 9, 28, 38, 78
`\punctstyle` 582, 2035

Q

`\quad` . 2657, 2945, 2946, 2949, 2950, 2956, 2960, 2968, 2971,
 2976, 2977, 2980, 3105, 3150, 3160, 3192, 3225, 3241, 3254
`\quotation` 110, 3624

R

`\raggedright` 2465, 2510, 2595, 2621, 3097, 3141
`\refname` 2237, 2242, 2249, 2258, 2266, 2267
`refname` 12
`\refstepcounter` 98, 2454, 2484, 2563, 2664, 3067, 3068
`\relax` 2419, 2443, 2466, 2477, 2482, 2498, 2511, 2518, 2534,
 2590, 2596, 2612, 2616, 2622, 2630, 2636, 2651, 2665,
 2671, 2676, 2693, 2710, 2715, 2722, 2723, 2728, 2733,
 2745, 2747, 2967, 2970, 2991, 2992, 3001, 3009, 3010,
 3019, 3027, 3028, 3037, 4213, 4217, 4220, 4248, 4249, 4254
`\renewcommand` 2415, 2428, 2548
`\RenewDocumentCommand` 3068
`\RequirePackage` 27,
 56, 57, 350, 361, 506, 509, 510, 511, 664, 683, 686,
 720, 728, 780, 1815, 1824, 1842, 1848, 2186, 3418, 4764
`\RequirePackageWithOptions` 4720
`resetalternatfont` 26, 65
reverse commands:
`\reverse_if:N` 585
`\rmdefault` 1719
`\rmfamily` 525, 528, 1258, 1261, 1637

S

scan commands:

`\scan_stop:`
 ... 65, 120, 836, 955, 1504, 1505, 2103, 2111, 4548, 4730
`scheme` 9, 39
`\scriptsize` 3552, 3591
`\secdef` 2425, 2448, 2558
`section/afterskip` 30
`section/beforeskip` 30
`section/numbering` 15
`section/runin` 20
`\sectionname` 3007, 3176
`\selectfont`
 1086, 1530, 1543, 1643, 1687, 1965, 1971, 1984, 2067, 3428

seq commands:

`\seq_clear:N` 1005
`\seq_concat:NNN` 1345
`\seq_gput_left:Nn` 2285, 2286
`\seq_gput_right:Nn` 1112, 3450
`\seq_gset_eq:NN` 2284
`\seq_gset_from_clist:Nn` 2279, 2289, 4528
`\seq_if_empty:Nf` 1360
`\seq_if_in:NnF` 1110
`\seq_if_in:NnT` 463
`\seq_map_function:NN` 2826
`\seq_map_inline:Nn` 1368, 2388, 2767, 3056, 4551
`\seq_new:N` .. 456, 1117, 1353, 1354, 2278, 2282, 3439, 4527
`\seq_put_right:No` 1616
`\seq_set_filter:NNn` 1343
`\seq_set_from_clist:Nn` 457
`\seq_set_split:NnV` 1342
`\seq_use:Nnnn` 3436
`\setbox` 2741
`\setCJKfamilyfont` 59, 1131,
 1205, 3845, 3851, 3852, 3853, 3854, 3855, 3856, 3899,
 3900, 3908, 3909, 3911, 3912, 3913, 3914, 3916, 3938,
 3940, 3942, 3944, 3983, 3984, 3985, 3986, 4010, 4015,
 4020, 4022, 4060, 4062, 4064, 4065, 4079, 4080, 4081,
 4082, 4095, 4096, 4097, 4098, 4099, 4100, 4141, 4142,
 4143, 4144, 4145, 4146, 4159, 4160, 4161, 4194, 4195, 4196
`\setCJKmainfont` .. 59, 1127, 1226, 1231, 1247, 3837, 3895,
 3903, 3930, 3976, 3998, 4055, 4076, 4091, 4136, 4155, 4190
`\setCJKmathfont` 59, 1242, 1250
`\setCJKmonofont` 59, 1129, 1237, 1249, 3850, 3898,
 3907, 3937, 3982, 4009, 4059, 4078, 4094, 4140, 4158, 4193
`\setCJKromanfont` 1231, 1251
`\setCJKsansfont` .. 59, 1128, 1232, 1248, 3840, 3844, 3897,
 3905, 3936, 3981, 4004, 4058, 4077, 4093, 4138, 4157, 4192
`\SetSymbolFont` 988, 1317, 1321
`\sfdefault` 1720
`\sffamily` 529, 532, 1262, 1265, 1638
six commands:
`\c_six` 3447
skip commands:
`\skip_horizontal:N` 1923, 1927
`\skip_if_eq:nnT` 1990

\@minus	3162, 3194, 3227, 3242, 3255, 3524, 3526, 3532, 3534, 3536, 3537, 3543, 3545, 3547, 3548, 3563, 3565, 3571, 3573, 3575, 3576, 3582, 3584, 3586, 3587
\@namedef	2880, 2882, 2886, 2888, 2892, 2894, 2902, 2907
\@nameuse	2880, 2882, 2892, 2894, 2902
\@ne	2863
\@nil	889
\@nobreakfalse	2736
\@noskipsecfalse	2740
\@noskipsectrue	2737
\@onlypreamble	34, 453, 519, 1247, 1248, 1249, 1250, 1251, 1252, 1662, 1676, 1684, 3652
\@part	87, 2425, 2448, 2451, 2481
\@pkgextension	6, 7, 8, 9, 20, 23, 682, 4759
\@plus	3111, 3112, 3162, 3163, 3194, 3195, 3227, 3228, 3242, 3255, 3268, 3284, 3524, 3525, 3526, 3532, 3533, 3534, 3536, 3537, 3543, 3544, 3545, 3547, 3548, 3563, 3564, 3565, 3571, 3572, 3573, 3575, 3576, 3582, 3583, 3584, 3586, 3587
\@popfilename	131, 4757
\@schapter	2558
\@secntformat	90, 2656, 2665
\@secpenalty	2650, 2651
\@sect	91, 2655, 2659
\@setfontsize	3514
\@spart	88, 2425, 2448, 2508, 2522
\@ssect	91, 2654, 2709
\@startsection	90, 92, 93, 99, 2633, 2772
\@svsec	2661, 2665, 2668, 2676, 2694, 2695
\@svsechd	2692, 2722, 2723, 2743
\@tempdima	2915, 2917
\@tempskipa	2407, 2408, 2443, 2447, 2590, 2594, 2616, 2620, 2636, 2638, 2639, 2650, 2671, 2672, 2710, 2711, 2728, 2729, 2732, 2745, 2746
\@tempswafalse	2440
\@tempswatrue	2438
\@topnewpage	2583
\@topnum	2551
\@xsect	92, 2708, 2726, 2727
\@zhdig	2196
\@zhnum	2188, 2195
\abstractname	12
\addCJKfontfeature	65
\AfterEndPreamble	130
\algorithmname	12
\appendixname	12, 22, 22, 22
\AtBeginDocument	34, 45
\AtBeginDvi	42
\AtBeginShipoutFirst	42
\AtEndOfClass	35
\AtEndOfPackage	35
\AtEndOfPackageFile*	130
\AtEndPreamble	130
\baselineskip	10, 11, 77, 79, 79
\bfseries	47
\bibname	12
\c@chapter	3306
\c@secnumdepth	2452, 2467, 2482, 2498, 2560, 2598, 2660, 2684, 2700
\c@section	3301
\caption	29
\catcode	33, 33, 44, 44, 121, 131
\ccwd	10, 10, 11, 24, 24, 29, 29, 34, 75, 76, 76, 79, 79, 80
\chapter	9, 15, 20, 21, 22, 22, 86, 103
\chaptermark	13
\chaptername	15
\char	49
\CheckCommand	76
\Chinese	29, 29
\chinese	24, 24, 29, 29, 81
\CJK@ignorespaces	46, 599
\CJK@encoding	55, 55, 56, 64, 795, 799, 883, 970, 971, 975, 977, 979, 980, 981, 983, 987, 989, 1312, 1315, 1317, 1321, 1462, 1477, 1478, 1495, 1496
\CJK@envStart	45, 45, 45, 45, 593
\CJK@family	50, 50, 790, 792, 795, 800, 847, 883, 1083, 1529, 1539, 1541, 1984, 1985
\CJK@ignorespaces	46, 46, 604, 608
\CJK@input	44, 45, 557
\CJK@loadBinding	44, 45, 589
\CJK@loadEncoding	44
\CJK@makeActive	45, 588
\CJK@plane	44, 76, 471, 472, 482, 1973
\CJK@surr	44, 44, 44, 558, 577
\CJK@upperReset	45
\CJKfamily	56
\CJKfamilydefault	45, 45, 60, 60, 68, 70, 70, 70, 70, 70
\CJKglue	79, 79, 109
\CJKhook	46, 46
\CJKrmdefault	43
\CJKsymbol	76
\CJKunderdotbasesep	29
\CJKunderline	29
\clearpage	45
\contentsname	11, 11
\CS	51, 51
\csname	109
\CTEX@afterindent	2642, 2759, 2764
\CTEX@aftertitle	2719, 2724, 2758, 2763
\CTEX@appendix@number	2785, 2797, 2803
\CTEX@appendix@numbering	2786, 2799, 2805
\CTEX@chapter@afterindent	2553
\CTEX@chapter@aftername	2602
\CTEX@chapter@afterskip	2612, 2630
\CTEX@chapter@aftertitle	2609, 2627
\CTEX@chapter@beforeskip	2590, 2616
\CTEX@chapter@fixbeforeskip	2382, 2591, 2617
\CTEX@chapter@format	2597, 2623
\CTEX@chapter@indent	2596, 2622
\CTEX@chapter@numbering	2562, 2600, 2805, 2956, 2976, 2977
\CTEX@chapter@pagestyle	2380, 2550

<code>\CTEX@chapter@titleformat</code>	2608, 2626	<code>\CTEX@toc@width@n</code>	2912, 2927, 2937
<code>\CTEX@chaptername</code>	2602	<code>\CTEX@todayold</code>	2204, 2209
<code>\CTEX@document@left@hook</code>	4727, 4733	<code>\CTEX@update@sectionformat@n</code>	92, 2641, 2755
<code>\CTEX@document@right@hook</code>	4729, 4735	<code>\CTEX@verbatim@font@hook</code>	1622, 1626
<code>\CTEX@Family@CMap</code>	468, 496, 499	<code>\ctex@zhmap@endinput</code>	4515
<code>\CTEX@fixbeforeskip</code>	86, 2403, 2445, 2592, 2618	<code>\CTEX@digits</code>	24, 24
<code>\CTEX@makeheadinganchor</code>		<code>\CTEX@number</code>	24, 24, 24, 24, 24
.....	86, 2399, 2401, 2458, 2488, 2570, 2667	<code>\CTEX@options</code>	29, 29
<code>\CTEX@part@afterindent</code>	2420	<code>\ctexset</code> .	5, 5, 5, 5, 5, 5, 5, 7, 11, 14, 14, 25, 26, 29, 29, 111
<code>\CTEX@part@aftername</code>	2469, 2500, 2995	<code>\CTEX@setup</code>	29
<code>\CTEX@part@afterskip</code>	2477, 2518, 2534, 3001	<code>\CTEX@thechapter</code>	12, 13, 13, 16
<code>\CTEX@part@aftertitle</code> ...	2474, 2504, 2515, 2528, 2999	<code>\CTEX@thesection</code>	12, 13, 16, 97
<code>\CTEX@part@beforeskip</code>	2419, 2443, 2991	<code>\CTEX@underdotbasesep</code>	29
<code>\CTEX@part@fixbeforeskip</code>	2381, 2444	<code>\CTEX@underline</code>	29
<code>\CTEX@part@format</code>	2466, 2497, 2512, 2525, 2993	<code>\curr@fontshape</code>	801, 831, 840, 844, 890
<code>\CTEX@part@indent</code>	2466, 2511, 2992	<code>\DeclareAlternateKanjiFont</code>	64
<code>\CTEX@part@numbering</code>	2453, 2467, 2483, 2498, 2994	<code>\DeclareFontEncoding</code>	53
<code>\CTEX@part@pagestyle</code>	2379, 2435	<code>\DeclareFontFamily</code>	42, 51
<code>\CTEX@part@titleformat</code> ..	2472, 2503, 2514, 2527, 2999	<code>\DeclareFontShape</code>	51
<code>\CTEX@partname</code>	2469, 2500, 2995	<code>\DeclareFontShape@</code>	64, 1421
<code>\CTEX@postappendix</code>	2790, 2798, 2804	<code>\DeclareRobustCommand</code>	132
<code>\CTEX@postchapter</code>	2804	<code>\define@newfont</code>	50, 823
<code>\CTEX@postsection</code>	2798	<code>\do@subst@correction</code>	50, 51, 822
<code>\CTEX@preappendix</code>	2789, 2796, 2802	<code>\em</code>	68, 68, 68
<code>\CTEX@prechapter</code>	2802	<code>\enc@update</code>	1688, 1693
<code>\CTEX@presection</code>	2796	<code>\escapechar</code>	54
<code>\CTEX@runin</code>	2673, 2712, 2730, 2760, 2765	<code>\EverySelectfont</code>	76
<code>\CTEX@save@appendix</code>	2791, 2794	<code>\external@font</code>	833, 834, 836
<code>\CTEX@save@refstepcounter</code>	3067, 3070	<code>\extract@font</code>	50, 821
<code>\CTEX@section@aftername</code>	3013	<code>\f@baselineskip</code>	2103
<code>\CTEX@section@afterskip</code>	3019	<code>\f@encoding</code>	799, 839, 952
<code>\CTEX@section@aftertitle</code>	3017	<code>\f@family</code>	50, 800, 839, 847
<code>\CTEX@section@beforeskip</code>	3009	<code>\f@series</code>	795, 883
<code>\CTEX@section@format</code>	3011	<code>\f@shape</code>	795, 883
<code>\CTEX@section@indent</code>	3010	<code>\f@size</code>	795, 801, 878, 886, 889, 890, 1909, 1955, 2918
<code>\CTEX@section@numbering</code>		<code>\fangsong</code>	8, 8
.....	2799, 2946, 2960, 2968, 2980, 3012	<code>\figurename</code>	11
<code>\CTEX@section@titleformat</code>	3017	<code>\font</code>	51, 51
<code>\CTEX@sectionname</code>	3013	<code>\font@name</code>	50, 51, 804, 811, 812,
<code>\CTEX@selectfont@hook</code>		815, 818, 836, 837, 854, 932, 933, 934, 935, 936, 943, 1700	
.....	76, 1960, 1961, 1963, 1966, 1973, 1976	<code>\fontcharwd</code>	76, 76
<code>\CTEX@setcurrentlabel@n</code>	3045, 3071	<code>\footnotesep</code>	109, 109, 109
<code>\CTEX@subsection@aftername</code>	3031	<code>\footnotesize</code>	109
<code>\CTEX@subsection@afterskip</code>	3037	<code>\get@external@font</code>	830
<code>\CTEX@subsection@aftertitle</code>	3035	<code>\getanddefine@fonts</code>	54, 928, 929
<code>\CTEX@subsection@beforeskip</code>	3027	<code>\globaljfont</code>	51
<code>\CTEX@subsection@format</code>	3029	<code>\hbox</code>	86
<code>\CTEX@subsection@indent</code>	3028	<code>\heiti</code>	8
<code>\CTEX@subsection@numbering</code>	2950, 2971, 3030	<code>\hrule</code>	86
<code>\CTEX@subsection@titleformat</code>	3035	<code>\Hy@driver</code>	1819
<code>\CTEX@subsectionname</code>	3031	<code>\Hy@unicodetrue</code>	73
<code>\CTEX@thechapter</code>	2803	<code>\HyPsd@ConvertToUnicode</code>	73
<code>\CTEX@thesection</code>	2797	<code>\HyPsd@LoadUnicode</code>	73
<code>\CTEX@titleformat@n</code>	92, 2718, 2724, 2757, 2762	<code>\HyPsd@pdfencoding</code>	73, 73
<code>\CTEX@titlepslabel@clear</code>	95, 2889, 2904	<code>\HyPsd@ToBigChars</code>	73
<code>\CTEX@titlepslabel@set</code>	95, 2883, 2895, 2899	<code>\if@mainmatter</code>	2561, 2599, 2976, 3332, 3333

\if@nobreak	2647	\parindent	10, 29, 79, 81, 109
\if@noskipsec	2416, 2634, 2739	\parskip	86, 86, 86
\if@openright	2429, 2537, 2548	\part	14, 20, 20, 21, 86
\if@tempswa	2543	\partname	15
\if@twocolumn	2436, 2582	\partpage	14, 97
\if@twoside	2536, 2947, 2957	\PassOptionsToClass	41
\iffalse	53	\PassOptionsToPackage	72
\ifin@	53, 911	\pdfmapline	115, 118
\iftrue	53	\pdfstringdef	73
\indexname	12	\pickup@font	50
\insertcontinuationtext	12	\pickup@jfont	827
\itdefault	51	\ProcessKeysOptions	27
\itshape	47	\proofname	12
\kaishu	8	\protected	49, 95
\kanjiskip	77	\protected@edef	2201, 2203, 2665, 3047
\labelformat	99, 99	\ProvidesFile	122
\leftmark	13	\ps@fancy	97, 2966, 2969, 2974, 2978
\linewidth	79, 79	\ps@headings	96, 2944, 2948, 2954, 2958
\lishu	8, 8	\refname	12, 12
\listfigurename	11	\refstepcounter	99
\listtablename	11	\relax	44, 79
\ltj@@does@alt@set	52	\reset@font	539, 1272
\ltj@@getjfontnumber	53, 899	\rightmark	13
\ltj@@set@stackfont	941, 942, 943	\rmfamily	47
\ltj@allalchar	725, 759	\roman	24
\ltj@curjfont	51, 76	\scantokens	132
\ltj@pickup@altfont@auxy	53, 884	\scriptfont@name	935, 942
\ltj@pickup@altfont@copy	53, 897	\section	9, 9, 14, 15, 20, 20, 21, 21, 22, 22, 30, 83
\ltj@setpar@global	940	\section*	15
\ltj@tempcntc	53, 903	\sectionmark	13, 13
\ltjalchar	49	\sectionname	15
\m@ne		\sectionpage	14, 98
	2452, 2467, 2560, 2598, 3712, 3713, 3714, 3715, 3716, 3717	\selectfont	52, 69, 76, 76, 76, 77, 79
\markboth	13	\setCJKfamilyfont	25, 25, 25
\markright	13	\setCJKmainfont	25
\math@bgroup	951	\sf@size	934
\math@egroup	956	\sffamily	47
\math@fonts	938	\size@update	79, 79, 2067, 2081
\mathbb	109	\sldefault	51
\maxdimen	11, 79	\songti	7
\meaning	132, 133, 133, 134	\special	7, 72
\newcommand	132	\split@name	889
\newrobustcmd	132	\SplitArgument	85
\newtitlemark	96, 96	\ssf@size	936
\nfss@catcodes	55	\strutbox	109
\ngostype	71	\subparagraph	9, 9, 9, 9, 20, 34
\normalsize	8, 8, 105	\subsectionname	15
\nouppercase	44	\subsectionpage	14, 98
\p@	1909, 1955, 2499, 2580, 2581, 2589, 2601, 2611, 2615, 2629, 2918, 3110, 3136, 3137, 3143, 3524, 3525, 3526, 3532, 3533, 3534, 3536, 3537, 3543, 3544, 3545, 3547, 3548, 3563, 3564, 3565, 3571, 3572, 3573, 3575, 3576, 3582, 3583, 3584, 3586, 3587	\subst@correction	850, 857
\pagenumbering	24	\tablename	12
\pagestyle	9	\textfont@name	933, 941
\paragraph	9, 9, 9, 9, 9, 20, 34, 34	\tf@size	932
		\thechapter	13, 16, 22
		\thesection	13, 15, 16, 22, 97
		\titleformat	93
		\titlespacing	94

- \today 11, 11
- \trans@languagepath 3381, 3407
- \ttfamily 47
- \ttl@extract 94
- \ttl@setsubmark 95, 2885, 2891
- \ttl@settopmark 95, 2879, 2910
- \ttlh@hang 2844, 2854
- \ttlh@runin 2843, 2853
- \tw@ 1699
- \uppercase 44, 44
- \use@mathgroup 54, 770, 960
- \usepackage 28, 29
- \verbatim@font 1620, 1624
- \wrong@fontshape 50
- \x@protect 4843
- \xdef 95
- \XeTeXglyphbounds 126
- \xkanjiskip 77, 77, 77, 77
- \yahei 8, 8
- \youyuan 8, 8
- \z@ . 712, 715, 2465, 2510, 2551, 2595, 2621, 2638, 2672,
2711, 2729, 2741, 2863, 3525, 3533, 3544, 3564, 3572, 3583
- \zhdig 81
- \zhdigits 24
- \zhnum 81
- \zhnumber 24, 24, 24, 24
- \zihao 23, 23
- \ziju 24
- \zw 76
- tex commands:
 - \tex_def:D 4919
 - \tex_divide:D 79, 2112, 2153
 - \tex_edef:D 4973
 - \tex_endlinechar:D 548, 554
 - \tex_font:D 486, 4548
 - \tex_iftrue:D 3333
 - \tex_ignorespaces:D
..... 1080, 1218, 1223, 2020, 2137, 3424, 4730
 - \tex_immediate:D 483
- \the 4217, 4220, 4221, 4240, 4244
- \thechapter 2564, 2567, 2601, 2955, 2975, 3140
- \theparagraph 3239
- \thepart 2455, 2468, 2485, 2499, 3084
- \thesection 2945, 2959, 2967, 2979, 3159
- \thesubparagraph 3252
- \thesubsection 2949, 2970, 3190
- \thesubsubsection 3223
- \thispagestyle 2434, 2435, 2539, 2549, 2550
- three commands:
 - \c_three 309, 2157, 3279, 4565
- \tiny 3553, 3592
- tl commands:
 - \tl_clear:c 1845, 1856, 2372
 - \tl_clear:N
.. 193, 366, 389, 1561, 1565, 1985, 2058, 2386, 3479, 4927
 - \tl_const:cn 659, 682
 - \tl_const:cx 2296, 2302
 - \tl_const:Nn 409, 414, 419,
424, 521, 970, 985, 1254, 1255, 1256, 1257, 3315, 3334, 3336
 - \tl_const:Nx 3, 1311, 2077
 - \tl_gput_right:cn 464
 - \tl_gput_right:Nn 4724, 4726, 4748
 - \tl_gset:Nn 1285, 1786, 1793, 1794
 - \tl_gset:Nx 565, 568, 932, 934, 936, 1715, 1743, 1761, 3670
 - \tl_gset_eq:NN 1062, 1738
 - \tl_gset_rescan:Nnn 1292
 - \tl_head:N 3483
 - \tl_if_blank:nTF 1378, 1474, 1475, 1579, 1580, 1594, 1612
 - \tl_if_blank_p:n 1344
 - \tl_if_empty:cT 3061
 - \tl_if_empty:NF 790, 2130
 - \tl_if_empty:nF 1656, 1669
 - \tl_if_empty:NTF 2308, 3633
 - \tl_if_empty:nTF 4890
 - \tl_if_eq:NNT 1711
 - \tl_if_eq:NNTF 2070, 3394
 - \tl_if_exist:NF
..... 520, 522, 523, 524, 1253, 1255, 1256, 1257, 4747
 - \tl_if_exist:NT 3417
 - \tl_if_exist:NTF 84, 2252, 2263, 2266, 3327
 - \tl_if_in:nnT 734
 - \tl_if_in:NnTF 4954
 - \tl_map_inline:Nn 1490
 - \tl_map_inline:nn 3480
 - \tl_new:c 2294, 2295
 - \tl_new:N 58, 128,
132, 522, 523, 524, 792, 793, 881, 967, 1022, 1048, 1068,
1069, 1090, 1426, 1567, 1731, 1999, 2078, 2091, 2789,
2790, 3823, 4737, 4738, 4747, 4808, 4809, 4930, 4931, 4932
 - \tl_put_left:Nn 1637, 1638, 1639, 1968, 4757
 - \tl_put_right:Nn
..... 373, 1607, 2345, 2347, 2349, 2351, 2353, 2355
 - \tl_put_right:Nx 938, 1574, 2324, 3483, 3484
 - \tl_replace_all:Nnn 1341, 1382
 - \tl_rescan:nn 134, 4981
 - \tl_rescan:no 4972
 - \tl_set:cn 2370, 2373
 - \tl_set:Nn 114, 115, 188, 221,
222, 514, 515, 516, 781, 783, 794, 882, 964, 965, 1006,
1073, 1085, 1340, 1379, 1381, 1422, 1595, 1599, 1633,
1634, 1635, 1636, 1645, 1984, 2000, 2043, 2054, 2092,
2377, 3393, 3611, 3824, 3828, 3870, 3871, 3872, 4802, 4803
 - \tl_set:Nx
.. 129, 482, 561, 833, 1209, 1733, 2034, 2140, 2840, 2859
 - \tl_set_eq:NN
799, 800, 847, 933, 935, 979, 1074, 1099, 1529, 1541, 2079
 - \tl_set_rescan:Nno 4928
 - \tl_tail:N 2845
 - \tl_tail:n 2855
 - \tl_to_str:n 125, 126, 4871, 4900, 4941, 4942
 - \tl_trim_spaces:n 1558
 - \today 2204, 2209, 2212, 2217

- today 11, 82
- token commands:
- \token_get_replacement_spec:N 4828, 4850
 - \token_to_meaning:N 4885
 - \token_to_str:N
.. 54, 914, 923, 932, 934, 936, 1127, 1128, 1129, 1131,
1138, 1139, 1140, 1202, 2249, 2250, 3432, 4555, 4715, 4787
 - \toks 4217, 4218, 4220, 4221, 4240, 4244
 - \topsep 3536, 3547, 3575, 3586
 - \topskip 2407
 - \tracingfonts 1699
 - \TrimSpaces 433
- true commands:
- \c_true_bool 2764
 - \ttdefault 1721
 - \ttfamily 533, 536, 1266, 1269, 1639
- two commands:
- \c_two 95, 163, 377, 381, 2158, 2918, 3263, 3295
 - \c_two_hundred_fifty_six 564, 567
 - \twocolumn 2544
 - \typeout 2564, 2565
- U**
- um commands:
- __um_define_math_chars: 767
 - __um_input_math_symbol_table: 742
 - __um_switchto_literal: 772
 - __um_sym:nnn 732
 - \undefined 1705, 4242
 - \unless 2673, 2712, 2730
 - \unskip 2744
 - \updefault ... 975, 981, 983, 987, 989, 1313, 1315, 1318, 1322
 - \upshape 1648
- use commands:
- \use:c 801, 839,
840, 890, 932, 934, 936, 990, 1040, 1542, 4955, 4975, 4989
 - \use:n 133, 1282, 1286, 1732, 1887, 2762, 2923, 3325
 - \use:nn 4983
 - \use:x 64, 122, 388, 778, 889, 972, 1021,
1157, 1384, 1558, 2393, 3442, 3642, 4824, 4866, 4917, 4938
 - \use_i:nn 108, 911, 4768, 4773, 4793, 4798
 - \use_i:nnn 261
 - \use_ii:nn 109, 911, 4768, 4773, 4793, 4798
 - \use_ii:nnn 134, 266
 - \use_iii:nnn 271
 - \use_none:n .. 189, 194, 199, 1057, 1285, 1419, 1888, 2190
 - \use_none:nn 499
 - \usebeamercolor 2989, 3007, 3025, 3126, 3181, 3214
 - \usebeamerfont 2989, 2998,
3007, 3016, 3025, 3034, 3125, 3128, 3180, 3183, 3213, 3216
 - \uselanguage 3372, 3377, 3398, 3403
 - \usepackage 4715
 - UTF8 6, 37
- V**
- \verse 110, 3622, 3623
 - \vfil 2442, 2533
 - \vskip 2476, 2477,
2499, 2517, 2518, 2534, 2601, 2611, 2612, 2629, 2630,
2732, 2733, 2990, 3008, 3026, 3110, 3127, 3143, 3182, 3215
 - \vspace 2447, 2589, 2594, 2615, 2620
- W**
- winfonts 28
 - \write 4257
- X**
- \x 4219, 4222,
4223, 4224, 4225, 4226, 4227, 4228, 4229, 4230, 4231,
4232, 4233, 4234, 4235, 4236, 4237, 4238, 4245, 4255, 4259
 - \xdef 4258
 - \xeCJKsetup 665, 1852, 1927, 2015, 2018, 2021, 2036
- xetex commands:
- \xetex_charglyph:D 4552
 - \xetex_glyphbounds:D 4576
- Y**
- \yahei 4209
 - \youyuan 4207
- Z**
- zero commands:
- \c_zero 161, 2113, 3387, 3414
 - \c_zero_dim 2105, 2117, 2132, 2148, 2155, 2359, 3296
 - \c_zero_fp 2077, 3388, 3415, 3601
 - \c_zero_skip 2357, 2358
 - \zhdig 2196
 - \zhdigits 24, 2203
 - zhmap 7, 28, 38
 - \zhnum 2195
- zhnum commands:
- \zhnum_counter:n 2187
 - \zhnumber 24, 2201
 - \zhnumsetup 2213, 2218
 - \zhtoday 2212, 2217
 - \zihao 23, 106, 3423, 3432
 - zihao 8, 36
 - \ziju 24, 80, 2136
 - \zw 713, 1913, 1956, 2001