

CT_EX 宏集手册

ctex.org

2016/02/02 v2.3*

简介

CT_EX 宏集是面向中文排版的通用 L^AT_EX 排版框架,为中文 L^AT_EX 文档提供了汉字输出支持、标点压缩、字体字号命令、标题文字汉化、中文版式调整、数字日期转换等支持功能,可适应论文、报告、书籍、幻灯片等不同类型的中文文档。

CT_EX 宏集支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX 和 LuaL^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 实现。

目录

第 1 节 介绍	2	第 8 节 实用命令	21
		8.1 字号与间距	21
		8.2 中文数字转换	23
		8.3 杂项	23
第 2 节 简明教程	3	第 9 节 LuaL^AT_EX 下的中文支持方式	23
2.1 CT _E X 宏集的组成	3	9.1 LuaL ^A T _E X 下替代字体的设置	24
2.2 CT _E X 宏集的安装和更新	3	第 10 节 CT_EX 宏集的配置	25
2.3 使用 CT _E X 文档类	4	10.1 修改宏包默认选项	25
2.4 使用 ctex 宏包	4	10.2 宏包载入后的配置	26
		10.3 配置标题中文翻译	26
第 3 节 宏包选项与 \ctexset 命令	5	10.4 自定义字体集	26
		第 11 节 对旧版本的兼容性	26
第 4 节 编译方式、编码与中文字库	5	11.1 CT _E X 0.8a 及以前的版本	26
4.1 编译方式	5	11.2 CT _E X 0.9–CT _E X 1.0d	27
4.2 中文编码	6	11.3 CT _E X 1.02c 以后的 SVN 开发版	28
4.3 中文字库	6	11.4 CT _E X 2.2 之前的版本	29
第 5 节 排版格式设定	8	第 12 节 宏集依赖情况与手工安装方法	29
5.1 文档默认字号	8	第 13 节 开发人员	30
5.2 章节标题风格	9	第 14 节 参考文献	30
5.3 排版方案选项	9	第 15 节 代码实现	31
第 6 节 文档汉化	11	版本历史	126
6.1 日期汉化	11	代码索引	126
6.2 文档标题汉化	11		
6.3 页面格式设置与汉化	12		
第 7 节 章节标题格式设置	13		

*ctex-kit rev. 4e02493.

第 1 节 介绍

历史

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

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

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

2014 年 3 月，为了适应 L^ATeX 的最新发展，特别是 L^ATeX3 的逐渐成熟，李清用 L^ATeX3 重构了整个宏包的代码，并重新使用 `doc` 和 `DocStrip` 工具进行代码的管理，升级版本号为 2.0，并改称 CT_EX 宏集。

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

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

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

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

关于宏集名字的说明

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

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

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

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 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三个中文文档类和四个宏包当中,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
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
	ctexcap.sty	过时宏包,仅作兼容使用,相当于开启章节标题设置功能的 <code>ctex.sty</code>
	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的(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。

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。

对于 $\text{T}_{\text{E}}\text{X}$ 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

```
tlmgr update --all
```

来完整更新已安装的宏包。

对于 $\text{M}_{\text{I}}\text{K}_{\text{T}}\text{E}_{\text{X}}$, 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 $\text{C}_{\text{T}}\text{E}_{\text{X}}$ 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 $\text{C}_{\text{T}}\text{E}_{\text{X}}$ 宏集提供的三个中文文档类。

$\text{C}_{\text{T}}\text{E}_{\text{X}}$ 宏集提供了三个中文文档类: `ctexart`、`ctexrep` 和 `ctexbook`, 分别对应 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 的标准文档类 `article`、`report` 和 `book`。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 或 LuaLaTeX 编译, 也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。
\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。

以下是使用 `beamer` 文档类编写中文演示文稿的一个示例。

例 2

```
\documentclass{beamer}
\usepackage[UTF8]{ctex}
\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 或 LuaLaTeX 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}
```

有些文档类是建立在 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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}
```

⁶使用 (pdf) $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 时也能够使用 GBK 编码, 但不推荐。(见 4.2 节)

```
\section{简介}
章节标题中文化的 \LaTeX{} 手册。
\end{document}
```

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`
New: 2014-03-18
`\ctexset` $\{ \langle \text{键值列表} \rangle \}$

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

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

例 4

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

`\ctexset` 采用 L^AT_EX 3 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)。

不同的编译方式和中文支持方式会在一定程度上影响 CT_EX 宏集的行为,比如宏包对编码的处理。在用户使用 X_YL^AT_EX 及 LuaL^AT_EX 编译时,CT_EX 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 L^AT_EX 及 pdfL^AT_EX 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。

⁷L^AT_EX、pdfL^AT_EX、X_YL^AT_EX 以及 LuaL^AT_EX。

表 2 C_TE_X 宏集的中文支持方式

(pdf)L ^A T _E X	X _Y L ^A T _E X	LuaL ^A T _E X
CJK	xeCJK	LuaTeX-ja

除去文档编码之外,选择不同的编译方式还可能影响 C_TE_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。C_TE_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 X_YL^AT_EX 或 LuaL^AT_EX 编译。

使用 X_YL^AT_EX 或 LuaL^AT_EX 编译时,C_TE_X 宏集强制使用 UTF-8 编码,此时 GBK 选项无效。使用 (pdf)L^AT_EX 编译时,C_TE_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 C_TE_X 宏集工作在 UTF-8 编码下。

4.3 中文字库

以往,为 L^AT_EX 文档配置中文支持是一件相当繁琐的事情。默认情况下,C_TE_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 C_TE_X 宏集自动配置字体策略

	Mac OS X	Windows New ¹	Windows Old ²	其他
X _Y L ^A T _E 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
LuaL ^A T _E X ⁴	LuaTeX-ja 华文字库	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja 中易字库	LuaTeX-ja Fandol 字库
pdfL ^A T _E X	不可用	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	不可用
L ^A T _E X + DVIPDFM _x	不可用	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	CJK + zhmetrics 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 LuaL^AT_EX 编译时使用 LuaTeX-ja 宏包。对此,第 9 节有特别说明。

通常,由 C_TE_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 C_TE_X 仍然提供了一系列选项,供在 C_TE_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。

zhmap ☆ zhmap = (true|false|zhmCJK)

Updated: 2014-03-08

指定字体映射机制。本选项只在使用 pdfL^AT_EX/L^AT_EX 编译时有意义。

⁸C_TE_X 宏集现在能够识别 Mac OS X 系统以及 Windows 系统。

<code>true</code>	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。
<code>false</code>	使用传统的 CJK 字库 (Type 1) ⁹ 。
<code>zhmCJK</code>	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 = (none|adobe|fandol|founder|mac|ubuntu|windows|windowsnew|windowsold|...)`

New: 2014-03-08

指定 C_TE_X 宏集加载的字库。

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

C_TE_X 预定义了以下六种中文字库。

<code>adobe</code>	使用 Adobe 公司的四款中文字体, 不支持 <i>pdf</i> L _A T _E X。
<code>fandol</code>	使用 Fandol 中文字体, 不支持 <i>pdf</i> L _A T _E X。
<code>founder</code>	使用方正公司的中文字体。
<code>mac</code>	使用 Mac OS X 系统下的华文字体, 不支持 (<i>pdf</i>)L _A T _E X。
<code>ubuntu</code>	使用 Ubuntu 系统下的文泉驿和文鼎字体。
<code>windows</code>	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。
<code>windowsnew</code>	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。
<code>windowsold</code>	使用简体中文 Windows XP 或之前系统下的中易字体。

注意: 使用 (*pdf*)L_AT_EX 编译的时候, 若设置 `zhmap = false` (比如需要使用 L_AT_EX + 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_TE_X 预定义的中文字库, 可以设置 `fontset` 为下述值之一。

<code>none</code>	不配置中文字体, 需要用户自己配置。
<code><name></code>	这里 <code><name></code> 为自定义的名字。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_TE_X 宏集预定义的中文字库还定义了一些字体命令。

<code>\songti</code>	宋体, CJK 等价命令 <code>\CJKfamily{zhsong}</code> 。
<code>\heiti</code>	黑体, CJK 等价命令 <code>\CJKfamily{zhhei}</code> 。
<code>\fangsong</code>	仿宋, CJK 等价命令 <code>\CJKfamily{zhfs}</code> 。
<code>\kaishu</code>	楷书, CJK 等价命令 <code>\CJKfamily{zhkai}</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`。

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` ☆ `zihao = (-4|5|false)`

New: 2015-05-06

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。本选项可以用于三个 \TeX 文档类和 `ctex` 宏包, 也可以用于 `ctexsize` 宏包。

`scheme = chinese` 时, 默认值为 5, 即设置 `\normalsize` 为五号字。

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` ☆ \TeX 文档类是在 \LaTeX 标准文档类之上开发的。因此, 除了可以使用 \TeX 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, \TeX 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 \TeX 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。 \TeX 会将这些选项传给标准文档类¹²。

¹²事实上, \LaTeX 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 宏集提供的三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 宏集提供的三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` ☆ 修改 `\paragraph` 和 `\subparagraph` 的格式。

`sub4section` ☆

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 `afterskip` 等选项。

注意, 上述两个选项只有在 `heading` 选项启用的时候才有意义。亦即, 只有在使用三个 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` 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”, 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;

`plain` 原样 (不调整标点间距)。

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。

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

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。

- true 总是保留汉字后的空格。此时,用户需要自行在行尾加上 % 处理换行产生的空格¹⁵。
- false 总是忽略掉汉字后面的空格,不论汉字后是什么(使用 (pdf)L^AT_EX 编译时);等同于 auto 的效果(使用 X_YL^AT_EX 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 Lua^AT_EX 编译的时候,该选项无效:汉字间的空格以及汉字与西文字符之间的空格总是有效,不会被忽略,但可以自动忽略掉由换行产生的空格。

linespread ☆ linespread = \langle 数值 \rangle

New: 2014-04-23

接受一个浮点数值,设置行距倍数。本选项的初始值与 scheme 有关。

scheme = chinese 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。

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

autoindent autoindent = \langle true|false|数值|带单位的数值 \rangle

New: 2014-03-13

在字体大小发生变化时,是否自动调整段首缩进(\parindent)的大小。

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

true 等价于设置 autoindent = 2。

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

例 8

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

linestretch ★ linestretch = \langle 数值或长度 \rangle

New: 2014-03-26

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

¹⁵L^AT_EX 将单个换行视作一个空格。

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化,使之以中文的方式显示今天的日期。如编译本文档的日期就是“2016 年 2 月 2 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 2 月 2 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年二月二日”。使用全汉字的日期格式。
`old` 效果为“February 2, 2016”。使用文档原来的(英文)日期格式。

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

例 9

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

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

6.2 文档标题汉化

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

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

```
contentsname * contentsname = <名字>
```

设置目录标题名 `\contentsname`。中文默认为“目录”。

```
listfigurename * listfigurename = <名字>
```

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

```
listtablename * listtablename = <名字>
```

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

```
figurename * figurename = <名字>
```

设置图片环境标题名 `\figurename`。中文默认为“图”。

`tablename` * `tablename = <名字>`

设置表格环境标题名 `\tablename`。中文默认为“表”。

`abstractname` * `abstractname = <名字>`

设置摘要 `abstract` 环境标题名 `\abstractname`。中文默认为“摘要”。注意 `book` 类没有摘要, 该选项无效。

`indexname` * `indexname = <名字>`

设置索引标题名 `\indexname`。中文默认为“索引”。

`appendixname` * `appendixname = <名字>`

设置附录标题名 `\appendixname`。中文默认为“附录”。

`bibname` * `bibname = <名字>`

设置参考文献标题名 `\refname`(对 `article`)或 `\bibname`(对 `report` 和 `book`)。中文默认为“参考文献”。

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:

```
\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”。

CT_EX 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)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。但注意，对 `article` 及其衍生的 `ctexart` 等文档类，没有 `chapter` 级别的标题。

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},
  }
}
```

<code>part/numbering</code>	*
<code>chapter/numbering</code>	*
<code>section/numbering</code>	*
<code>subsection/numbering</code>	*
<code>subsubsection/numbering</code>	*
<code>paragraph/numbering</code>	*
<code>subparagraph/numbering</code>	*

`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。

New: 2015-06-21

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。

<code>.../name</code>	*
-----------------------	---

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	同右	{}	
subsection	同右	{}	
subsubsection	同右	{}	
paragraph	同右	{}	
subparagraph	同右	{}	

`.../number` * `number = {(数字输出命令)}`

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

例 14

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

`number` 选项的定义同时将控制对章节计数器的交叉引用。在引用计数器时,记录在 \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 \the(标题) 等价定义
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	同右	<code>\thesection</code>	<code>\arabic{section}</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
}
```

表 7 format 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	\Large\bfseries\centering	\raggedright
part	\huge\bfseries\centering	\centering
chapter	\huge\bfseries\centering	\raggedright
section	\Large\bfseries\centering	\Large\bfseries
subsection	同右	\large\bfseries
subsubsection	同右	\normalsize\bfseries
paragraph	同右	\normalsize\bfseries
subparagraph	同右	\normalsize\bfseries

```
.../nameformat * nameformat = {{格式命令}}
.../nameformat+ * nameformat+ = {{格式命令}}
```

Updated: 2015-06-30

`nameformat` 用于控制章节名字的格式,作用域为章节名字,包括编号。它一般用于章节名(包括编号)与章节标题的字体、字号等设置不一致的情形。参见 `titleformat` 选项。

`nameformat+` 用于在已有的章节名字格式后附加内容。

`nameformat` 选项的最后一个格式命令可以是一个有一个参数的命令。这个命令的参数用于接受章节名字和编号,实现特殊效果(见例 19)。

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

表 8 nameformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\Large\bfseries
part	{}	\huge\bfseries
chapter	{}	\huge\bfseries
section	同右	{}
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	同右	<code>\par\vskip 20pt</code>
chapter	<code>\quad</code>	<code>\par\nobreak\vskip 20pt</code>
section	同右	<code>\quad</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	{}	\Huge\bfseries
chapter	{}	\Huge\bfseries
section	同右	{}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

```
.../aftertitle * aftertitle = {<代码>}
.../aftertitle+ * aftertitle+= {<代码>}
```

New: 2015-06-19

aftertitle 选项的参数 <代码> 将被插入到章节标题内容之后。

aftertitle+ 选项用于在原有代码后面附加代码。

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

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

表 11 aftertitle 选项的默认设置

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

表 12 runin 选项的默认设置

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

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

```
subsection/runin *
```

```
subsubsection/runin *
```

```
paragraph/runin *
```

```
subparagraph/runin *
```

New: 2015-06-27

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

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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

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

New: 2015-06-27

afterindent 选项用于设置章节标题后首段的缩进。

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

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

表 13 afterindent 选项的默认设置

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

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

Updated: 2015-06-27

`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	0pt plus 1fil
chapter	50pt
section	3.5ex plus 1ex minus .2ex
subsection	3.25ex plus 1ex minus .2ex
subsubsection	3.25ex plus 1ex minus .2ex
paragraph	3.25ex plus 1ex minus .2ex
subparagraph	3.25ex plus 1ex minus .2ex

`.../afterskip` * `afterskip = {<弹性间距>}`

Updated: 2015-06-27

`afterskip` 选项控制章节标题与后面下方之间的距离。

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

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

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

`.../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。
```

表 15 afterskip 选项的默认设置

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

part/pagestyle *
chapter/pagestyle *

New: 2014-03-21

pagestyle = {<页面格式>}

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

表 17 pagestyle 选项的默认设置

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

appendix/numbering *

New: 2015-06-21

numbering = true|false

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

用法与普通章节 numbering 选项类似。

appendix/name

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	\section	{}		{}	
book, report	\chapter	\appendixname\space	附录 _□	\appendixname\space	Appendix _□

`appendix/number` * `number = {<数字输出命令>}`

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

用法与普通章节的 `number` 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

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

表 19 `appendix/number` 选项的默认设置

文档类	影响命令	默认值
<code>article</code>	<code>\section</code>	<code>\Alph{section}</code>
<code>book, report</code>	<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 = {
    format      = \Huge\bfseries,
    nameformat  = \hrule height 1bp \relax\bigskip\hfill\chapternamebox,
    number      = \arabic{chapter},
    aftername   = \par\medskip,
    aftertitle  = \bigskip\hrule height 2bp \relax
  }
}
\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。

`\ziju` `\ziju {<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 `\ccwd` 的值,但不会影响英文字距。

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

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

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

8.2 中文数字转换

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

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

 $\backslash\text{chinese} \{(\text{counter})\}$

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

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

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

以中文格式输出数字。这里的数字可以是整数、小数和分数。

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

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

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 TEX 宏，不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 (number) 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 (number) 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 $\text{C}\text{T}\text{E}\text{X}$ 标志。

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

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

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

9.1 Lua \LaTeX 下替代字体的设置

AlternateFont `\setCJKfamilyfont` $\langle family \rangle$

New: 2014-04-14

```

\setCJKfamilyfont  $\langle family \rangle$ 
[
  AlternateFont =
  {
     $\langle character range_1 \rangle$  [ $\langle alternate font features_1 \rangle$ ]  $\langle alternate font name_1 \rangle$  ||
     $\langle character range_2 \rangle$  [ $\langle alternate font features_2 \rangle$ ]  $\langle alternate font name_2 \rangle$  ||
    .....
  } ,
   $\langle base font features \rangle$ 
]  $\langle base font name \rangle$ 

```

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

CharRange `\setCJKfamilyfont` $\langle family \rangle$

New: 2014-04-14

```

\setCJKfamilyfont  $\langle family \rangle$ 
[
  CharRange =  $\langle character range \rangle$  ,
   $\langle alternate font features \rangle$ 
]  $\langle alternate font name \rangle$ 

```

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

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

例 20

```

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

```

declarecharrange *

New: 2014-04-14

```

\ctexset
{
  declarecharrange =
  {
     $\langle name_1 \rangle$   $\langle character range_1 \rangle$  ,
     $\langle name_2 \rangle$   $\langle character range_2 \rangle$  ,
    ...
  }
}

```

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

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

```

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

```

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

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

C_TE_X 宏集的配置文件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-ja 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。

<code>cs4size</code> <code>c5size</code>	分别相当于 <code>zihao=-4</code> 和 <code>zihao=5</code> , 过时选项。
<code>CCT</code> <code>CCTfont</code>	相关选项已删除。
<code>indent</code> <code>noindent</code>	<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。
<code>zhmap</code> <code>nozhmap</code>	<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。
<code>winfonts</code> <code>adobefonts</code> <code>nofonts</code>	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)。
<code>punct</code> <code>nopunct</code>	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。
<code>cap</code> <code>nocap</code>	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。
<code>space</code> <code>nospace</code>	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。
<code>fancyhdr</code>	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。

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

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

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

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

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

在新版本中，上述由 `fntef` 衍生的相关宏都成为过时命令。

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

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

\CTEXnoindent 设置 `\parindent=0pt`。过时命令。

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

\CTEXoptions `\CTEXoptions[⟨选项⟩]`
相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

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

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

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

例 24

```
% 代替 \CTEXoptions[captiondelimiter={:}]
\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 pdfL_AT_EX 或 L_AT_EX + 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 X_LL_AT_EX 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。

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。

第 14 节 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 15 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 . \@current } { 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 \end{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 \end{ctex}
25 \end{class|ctex}

26 (*class|style)
27 \RequirePackage { xparse , l3keys2e }
28 \end{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 \end{class}

```

\c__ctex_engine_file_str 引擎检查,暂不支持 pTeX 系。

```

43 \msg_new:nnnn { ctex } { engine-not-supported }
44 { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }
45 { You~can~switch~to~xelatex,~lualatex~or~pdflatex. }
46 \file_if_exist:nTF { ctex-engine- \c_sys_engine_str .def }
47 {
48   \str_const:Nx \c__ctex_engine_file_str
49   { ctex-engine- \c_sys_engine_str .def }
50 }
51 { \msg_critical:nnx { ctex } { engine-not-supported } { \c_sys_engine_str } }

```

(End definition for \c__ctex_engine_file_str.)

```

52 \end{class|ctex}

53 (*class|ctex|ctexheading)

```

```

54 \RequirePackage { ctexhook , ctexpatch }
55 <ctexheading>\RequirePackage { fix-cm , everysel }

```

15.1 内部函数与变量

```

\l__ctex_tmp_tl
\l__ctex_tmp_int 56 \tl_new:N \l__ctex_tmp_tl
\l__ctex_tmp_dim 57 \int_new:N \l__ctex_tmp_int
\l__ctex_tmp_box 58 <ctexheading>\dim_new:N \l__ctex_tmp_dim
59 \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。

```

60 \cs_new_protected:Npn \ctex_file_wrapper:nnn #1#2#3
61 {
62   \use:x
63   {
64     \ExplSyntaxOff
65     \char_set_catcode_letter:n { 64 }
66     #1
67     \exp_not:n {#2}
68     \bool_if:NTF \l__kernel_expl_bool
69     { \ExplSyntaxOn }
70     { \ExplSyntaxOff }
71     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
72     #3
73   }
74 }

```

(End definition for \ctex_file_wrapper:nnn.)

`\ctex_file_input:n` 输入文件。

```

75 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
76 { \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。

```

77 \cs_new_protected_nopar:Npn \ctex_scheme_input:n #1
78 {
79   \ctex_file_wrapper:nnn
80   { }
81   {
82     \tl_if_exist:NTF \c__ctex_class_tl
83     {
84       \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
85       { \file_input:n { ctex-scheme- #1 .def } }
86     }
87     { \file_input:n { ctex-scheme- #1 .def } }
88   }
89   { }
90 }
91 \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。

```

92 \cs_new_eq:NN \g__ctex_section_depth_flag \c_two

```


(End definition for `\g__ctex_section_depth_flag`.)

```
93 </class|ctex|ctexheading>
94 <*class|ctex>
```

对旧版本的宏包给出错误信息。

```
95 \msg_new:nnon { ctex } { package-too-old }
96 { Support~package~`#1'~too~old. }
97 {
98   Please~update~an~up~to~date~version~of~the~package~`#1'\
99   using~your~TeX~package~manager~or~from~CTAN.
100 }
```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```
101 \sys_if_output_pdf:TF
102 { \cs_new_eq:NN \ifctexpdf \if_true: }
103 { \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。

```
104 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
105 \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。

```
106 \cs_new:Npn \ctex_set_default_ccwd:Nn #1#2
107 {
108   \dim_compare:nNnTF
109     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
110     { \tl_set:Nn #1 {#2} }
111     { \tl_set:Nn #1 { #2 \ccwd } }
112 }
```

(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`。

```
113 \cs_new:Npn \ctex_default_pt:n #1
114 {
115   \exp_after:wN \__ctex_default_pt:w
116   \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
117 }
118 \use:x
119 {
120   \cs_new:Npn \exp_not:N \__ctex_default_pt:w
121     ##1 \tl_to_str:n { pt } ##2 \exp_not:N \q_stop
122     { ##1 \tl_to_str:n { pt } }
123 }
```

(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。

```
124 \tl_new:N \l__ctex_encoding_tl
125 \tl_set:Nx \l__ctex_encoding_tl
126 { \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。

```
127 \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。

```
128 \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。

```
129 \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。

```
130 \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。

```
131 <class>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
```

```
132 <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。

```
133 <*class>
```

```
134 \clist_new:N \g__ctex_std_options_clist
```

```
135 </class>
```

(End definition for \g__ctex_std_options_clist.)

对无效选项给出警告。

```
136 \msg_new:nnn { ctex } { invalid-option }
```

```
137 { Option~`\l_keys_key_tl'~is~invalid~in~current~mode. }
```

```
138 \msg_new:nnn { ctex } { invalid-value }
```

```
139 { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_tl'. }
```

对过时选项或命令给出警告。

```
140 \msg_new:nnn { ctex } { deprecated-option }
```

```
141 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
```

```
142 \msg_new:nnn { ctex } { deprecated-command }
```

```
143 { Command~ #1 is~ deprecated.\\ #2 }
```

```
144 \msg_new:nnn { ctex } { deprecated-environment }
```

```
145 { Environment~`#1'~is~ deprecated.\\ #2 }
```

```
146 </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,其它值则不作修改。

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

```
148 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one
```

```
149 </class|ctex|ctexsize>
```

(End definition for \g__ctex_font_size_flag.)

15.2 宏包选项

```

150 <*class|style>
151 \keys_define:nn { ctex / option }
152 {
153 </class|style>

```

zihao

```

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

```

(End definition for zihao. This function is documented on page 21.)

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

```

177   linespread .fp_set:N = \l__ctex_line_spread_fp ,
178   linespread .initial:n = { \c_nan_fp } ,
179   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。

```

180   autoindent .choice: ,
181   autoindent .default:n = { true } ,
182   autoindent / true .code:n =
183   {
184     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
185     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
186   } ,
187   autoindent / false .code:n =
188   {
189     \tl_clear:N \l__ctex_autoindent_tl
190     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
191   } ,
192   autoindent / unknown .code:n =
193   {
194     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
195     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
196   } ,

```

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

indent 仅为兼容性保留, 已过时。

```

197 indent .code:n =
198   {
199     \msg_warning:nnn { ctex } { deprecated-option }
200     {
201       The~ functionality~ has~ been~ removed.\\
202       It's~ better~ to~ set~ the~ heading~ styles~ via~ `afterindent'~
203       options.~
204     }
205   } ,
206 indent .value_forbidden:n = true ,
207 noindent .code:n =
208   {
209     \msg_warning:nnn { ctex } { deprecated-option }
210     {
211       The~ functionality~ has~ been~ removed.\\
212       It's~ better~ to~ set~ the~ heading~ styles~ via~ `afterindent'~
213       options.
214     }
215   } ,
216 noindent .value_forbidden:n = true ,

```

(End definition for `indent`. This function is documented on page 27.)

GBK UTF8

```

217 GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
218 UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
219 GBK .value_forbidden:n = true ,
220 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。

```

221 fontset .tl_gset:N = \g__ctex_fontset_tl ,
222 nofonts .code:n =
223   {
224     \msg_warning:nnn { ctex } { deprecated-option }
225     {
226       Option~ `fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
227       fontset~ option.
228     }
229     \keys_set:nn { ctex / option } { fontset = none }
230   } ,
231 adobefonts .code:n =
232   {
233     \msg_warning:nnn { ctex } { deprecated-option }
234     {
235       Option~ `fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
236       fontset~ option.
237     }
238     \keys_set:nn { ctex / option } { fontset = adobe }
239   } ,
240 winfonts .code:n =
241   {
242     \msg_warning:nnn { ctex } { deprecated-option }
243     {
244       Option~ `fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
245       fontset~ option.
246     }
247     \keys_set:nn { ctex / option } { fontset = windows }
248   } ,
249 nofonts .value_forbidden:n = true ,
250 winfonts .value_forbidden:n = true ,
251 adobefonts .value_forbidden:n = true ,

```

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

zhmap

```

252 zhmap .choice: ,
253 zhmap .default:n = { true } ,
254 zhmap / zhmCJK .code:n =
255   {
256     \bool_gset_true:N \g__ctex_zhmCJK_bool
257     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
258   } ,
259 zhmap / true .code:n =
260   {
261     \bool_gset_false:N \g__ctex_zhmCJK_bool
262     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
263   } ,
264 zhmap / false .code:n =
265   {
266     \bool_gset_false:N \g__ctex_zhmCJK_bool
267     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
268   } ,
269 nozhmap .code:n =
270   {
271     \msg_warning:nnn { ctex } { deprecated-option }
272     { Option~ `zhmap=false'~ is~ set. }
273     \keys_set:nn { ctex / option } { zhmap = false }
274   } ,
275 nozhmap .value_forbidden:n = true ,

```

(End definition for `zhmap`. This function is documented on page 27.)

punct 设置标点符号输出格式。

```

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

```

(End definition for `punct`. This function is documented on page 27.)

space

```

286 space .choices:nn =
287   { true , auto , false }
288   {
289     \exp_args:Nx \ctex_at_end:n
290     { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
291   } ,
292 space .default:n = { true } ,
293 nospace .code:n =
294   {
295     \msg_warning:nnn { ctex } { deprecated-option }
296     { Option~ `space=false'~ is~ set. }
297     \keys_set:nn { ctex / option } { space = false }
298   } ,
299 nospace .value_forbidden:n = true ,

```

(End definition for `space`. This function is documented on page 27.)

heading

```

300 heading .bool_set:N = \l__ctex_heading_bool ,

```

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

```

301 </class|ctex>
302 <*class|ctex|ctexheading>

sub3section
sub4section
303 sub3section .code:n =
304   { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
305 sub4section .code:n =
306   { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
307 sub3section .value_forbidden:n = true ,
308 sub4section .value_forbidden:n = true ,

```

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

scheme

```

309 scheme .tl_set:N = \l__ctex_scheme_tl ,
310 <*ctexheading>
311   scheme .default:n = { plain } ,
312   scheme .initial:n = { plain }
313 }
314 </ctexheading>
315 <!ctexheading>
316   scheme .default:n = { chinese } ,
317   scheme .initial:n = { chinese } ,
318 </!ctexheading>

319 </class|ctex|ctexheading>
320 <*class|ctex>

321 cap .code:n =
322   {
323     \msg_warning:nnn { ctex } { deprecated-option }
324     { Option~ `scheme = chinese' ~ is~ set. }
325     \keys_set:nn { ctex / option } { scheme = chinese }
326   } ,
327 nocap .code:n =
328   {
329     \msg_warning:nnn { ctex } { deprecated-option }
330     { Option~ `scheme = plain' ~ is~ set. }
331     \keys_set:nn { ctex / option } { scheme = plain }
332   } ,
333 cap .value_forbidden:n = true ,
334 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

```

335 fntef .code:n =
336   {
337     \msg_warning:nnn { ctex } { deprecated-option }
338     { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
339   } ,
340 fancyhdr .code:n =
341   {
342     \msg_warning:nnn { ctex } { deprecated-option }
343     { `fancyhdr'~ package~ is~ loaded. }
344     \RequirePackage { fancyhdr }
345   } ,
346 hyperref .code:n =
347   {
348     \msg_warning:nnn { ctex } { deprecated-option }
349     { `hyperref'~ package~ will~ be~ loaded. }
350   }
\ctex_at_end:n

```

```

351     {
352         \cs_if_exist:NF \hypersetup
353         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
354     }
355     \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
356 } ,
357 }

```

(End definition for *fntef*, *fancyhdr*, and *hyperref*. These functions are documented on page 28.)

```

358 </class>ctex)
359 <*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

```

360 \tl_clear:N \l__ctex_tmp_tl
361 \clist_map_inline:nn
362 {
363     10pt , 11pt , 12pt ,
364     8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
365 }
366 {
367     \tl_put_right:Nn \l__ctex_tmp_tl
368     {
369         #1 .code:n =
370 <!*class>
371         { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
372 </!class>
373 <*class>
374         {
375             \cs_gset_eq:NN \g__ctex_font_size_flag \c_two
376             \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
377         } ,
378 </class>
379         #1 .value_forbidden:n = true ,
380     }
381 }
382 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
383 \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。

```

384 <*class>
385 \keys_define:nn { ctex / option }
386 {
387     unknown .code:n =
388         { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
389 }
390 </class>
391 <!ctexsize>\ctex_file_input:n { ctexopts.cfg }
392 </class>ctex|ctexsize)
393 <*class>style)
394 \ProcessKeysOptions { ctex / option }
395 </class>style)
396 <*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。

```

397 \if_case:w \g__ctex_font_size_flag
398     \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
399 \or:
400     \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
401 \fi:

```


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

```

402 (*article)
403 \tl_const:Nn \c__ctex_class_tl { article }
404 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
405 \LoadClass { article }
406 /article)
407 (*book)
408 \tl_const:Nn \c__ctex_class_tl { book }
409 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
410 \LoadClass { book }
411 /book)
412 (*report)
413 \tl_const:Nn \c__ctex_class_tl { report }
414 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
415 \LoadClass { report }
416 /report)
417 /class)

```

15.3 用户设置接口

`\ctexset`

```

418 (*class|ctex|ctexheading)
419 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
420 /class|ctex|ctexheading)

```

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

`\CTEXsetup` 过时命令。
`\CTEXoptions`

```

421 (*class|ctex)
422 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
423 {
424   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
425   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
426   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
427 }
428 \NewDocumentCommand \CTEXoptions { +o }
429 {
430   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
431   { \ctexset~ {~ #1~ }~ is~ set. }
432   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
433 }
434 /class|ctex)

```

(End definition for `\CTEXsetup` and `\CTEXoptions`. These functions are documented on page 28.)

15.4 特定引擎支持与设置

15.4.1 ctex-engine-pdftex.def

`\c__ctex_cmap_encoding_seq`

需要加上 CMap 的 CJK 字体编码。

```

436 \seq_new:N \c__ctex_cmap_encoding_seq
437 \seq_set_from_clist:Nn \c__ctex_cmap_encoding_seq
438 { 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。

```

439 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2
440 {
441   \cs_if_free:cF { #1 + #2 }
442   {
443     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
444     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
445   }
446 }
447 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
448 \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。

```

449 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
450 {
451   \cs_if_free:NF \CJK@plane
452   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
453 }
454 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
455 {
456   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
457   #1
458 }
459 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
460 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
461 {
462   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
463   \tex_immediate:D \pdfobj stream ~ file { \l__ctex_tmp_tl .cmap }
464   \cs_new_protected_nopar:Npx #1
465   {
466     \exp_not:N \pdffontattr \exp_not:N \font
467     { /ToUnicode ~ \int_use:N \pdflastobj \c_space_tl 0 ~ R }
468   }
469 }

```

(End definition for \ctex_add_cmap:n.)

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

```

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

```

(End definition for \DeclareFontFamily.)

首先检查选项，决定是否载入 zhmCJK 宏包。

```

484 \if_bool:N \g__ctex_zhmCJK_bool
485 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
486 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

487 \else:
488 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }

```

```

489     { \RequirePackage { CJK } }
490     { \RequirePackage { CJKutf8 } }
491     \RequirePackage { CJKpunct , CJKspace }

```

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

```

492     \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
493     {
494         \tl_set:Nn \CJKrmdefault {#1}
495         \tl_set:Nn \CJKsfdefault {#2}
496         \tl_set:Nn \CJKttdefault {#3}
497         \AtBeginDvi { \file_input:n {#4} }
498         \ctex_at_end_package:nn { atbegshi }
499         { \AtBeginShipoutFirst { \file_input:n {#4} } }
500     }
501     \@onlypreamble \ctex_load_zhmap:n

```

(End definition for `\ctex_load_zhmap:nnnn`)

```

502     \tl_if_exist:NF \CJKfamilydefault
503     { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
504     \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
505     \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
506     \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
507     \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
508     { \CJKfamily { \CJKrmdefault } }
509     { }
510     { \ctex_patch_failure:N \rmfamily }
511     \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
512     { \CJKfamily { \CJKsfdefault } }
513     { }
514     { \ctex_patch_failure:N \sffamily }
515     \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
516     { \CJKfamily { \CJKttdefault } }
517     { }
518     { \ctex_patch_failure:N \ttfamily }
519     \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
520     { \CJKfamily { \CJKfamilydefault } }
521     { \cs_set_eq:NN \reset@font \normalfont }
522     { \ctex_patch_failure:N \normalfont }

```

`zhmCJK` 判断结束。

```
523 \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。

```

524     \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
525     {
526         \ctex_file_wrapper:nnn
527         {
528             \char_set_catcode_other:n { 60 } % <
529             \char_set_catcode_math_superscript:n { 94 } % ^
530             \int_set_eq:NN \tex_endlinechar:D \c_minus_one
531         }
532         { \file_input:n {#1} }
533         {
534             \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
535             \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
536             \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
537         }
538     }
539     \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。

```

540 \if_cs_exist:N \CJK@surr
541   \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
542     {
543       \tl_set:Nx \l__ctex_tmp_tl {#2}
544       \int_set:Nn \l__ctex_tmp_int
545         { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
546       \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
547         { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
548         {
549           \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
550           \tl_gset:Nx #1
551             {
552               \int_to_Hex:n
553                 { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
554               \int_to_Hex:n
555                 { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
556             }
557         }
558     }
559   \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
560 \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`。

```

561 \AtBeginDocument
562 {
563   \str_if_eq_x:nnF { \l__ctex_punct_tl } { quanjiao }
564   { \punctstyle { \l__ctex_punct_tl } }
565 }

```

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。

```

566 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
567   \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
568   {
569     \CJK@makeActive
570     \CJK@loadBinding { standard }
571   }

```

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}

```

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

```

\else
  \CJKenc{#2}
\fi
\CJKfontenc{#2}{#1}
\CJKfamily{#3}
\def\CJK@series{\f@series}
\def\CJK@shape{\f@shape}%
\csname CJKhook\endcsname}

```

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

```

572 \exp_args:Nx \ctex_at_end_preamble:n
573 {
574   \exp_not:N \CJK@envStart
575   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
576   \exp_not:N \CJKtilde
577 }

```

`zhmCJK` 判断结束。

```
578 \fi:
```

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

```

579 <@@=>
580 \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。

```
581 <@@=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。

```

582 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
583 {
584   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
585   { \cs_set_eq:NN \CJK@ignorespaces #1 }
586   \ctex_set_ignorespaces:
587 }
588 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
589 { \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。

```
590 \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。

```

591 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
592 {
593   \clist_map_inline:Nn \c__ctex_punct_family_clist
594   {
595     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
596     {
597       \cs_set_eq:cc
598       { CJKpunct@ ##1 @spaces }

```

```

599         { c__ctex_ #1 ##1 _punct_spaces_tl }
600     }
601 }
602 }
603 \clist_const:Nn \c__ctex_punct_family_clist
604 {
605     zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
606     zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
607 }

```

(End definition for \ctex_punct_set:n.)

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

```

608 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
609 {
610     \cs_if_free:cF { CJKpunct@ #2 @spaces }
611     { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
612 }

```

(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。

```

613 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
614 {
615     \clist_map_inline:nn {#1}
616     {
617         \ctex_punct_map_series:nnn { ##1 } { b } {#2}
618         \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
619     }
620 }
621 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
622 {
623     \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
624     \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
625     \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
626     \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
627     \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
628     \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
629 }

```

(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。

```

630 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
631 {
632     \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
633     \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
634     \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
635     \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
636     \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
637     \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
638 }

```

(End definition for \ctex_punct_map_itshape:nn.)

\ctex_punct_space:nn 定义标点的边界信息。

\ctexspadef

```

639 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
640 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
641 \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。

```
642 \ctex_file_input:n { ctexspa.def }
643 </pdfTeX>
```

15.4.2 ctex-engine-xetex.def

```
644 <(*xetex)
645 \RequirePackage { xeCJK }
646 \exp_args:Nx \xeCJKsetup
647 {
648   LoadFandol = false ,
649   AutoFakeBold = true ,
650   PunctStyle = \l__ctex_punct_tl
651 }
```

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。

```
652 \@ifpackagelater { fontspec } { 2014/05/25 } { }
653 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
654 </xetex>
```

15.4.3 ctex-engine-luatex.def

```
655 <(*luatex)
```

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

```
656 \msg_new:nnn { ctex } { luatexja-loaded }
657 {
658   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
659   Loading~file~`#1'~will~abort!
660 }
661 \@ifpackageloaded { luatexja }
662 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
663 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
664 \RequirePackage { luatexja }
665 \@ifpackagelater { luatexja } { 2015/09/21 } { }
666 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
667 \RequirePackage { fontspec }
668 \@ifpackagelater { fontspec } { 2014/05/25 } { }
669 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
```

15.4.3.1 LuaTeX-ja 的默认设置

```
670 \ExplSyntaxOff
```

以下设置抄录自 llTjdefs.sty。

```
671 \ltjdefcharrange{1}{"80-"36F, "1E00-"1EFF}
672 \ltjdefcharrange{2}{"370-"4FF, "1F00-"1FFF}
673 \ltjdefcharrange{3}{%
674   "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
675 \ltjdefcharrange{4}{%
676   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
677   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
678   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
679 \ltjdefcharrange{5}{"D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
680 \ltjdefcharrange{6}{%
681   "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
682   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
683 \ltjdefcharrange{7}{
684   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
685   "AC00-"D7FF}
```



```

686 \ltjdefcharrange{8}{"A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
687 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, -8}}
688 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 ltj-latex.sty。

```

689 \directlua{
690   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
691   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
692 }
693 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .4pt,
694   xkanjiskip=.25\zw plus 1pt minus 1pt,
695   autospacing, autoxspacing, jacharrange={-1},
696   yalbaselineshift=\z@, yjabaselineshift=\z@,
697   jcharwidowpenalty=500, differentjfm=paverage
698 }
699 \ExplSyntaxOn

```

15.4.3.2 LuaTeX-ja 的补丁

```
700 <@@=ctex_ltj>
```

在 L^AT_EX 下, LuaTeX-ja 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。

```

701 \RequirePackage { xunicode-addon }
702 \AtBeginUTFCommand
703 {
704   \group_begin:
705   \lua_now_x:n { tex.globaldefs = 0 }
706   \ltj@allalchar
707 }
708 \AtEndUTFCommand { \group_end: }

```

对 fontspec 沿用 LuaTeX-ja 的补丁。

```
709 \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。

```

710 \cs_new_protected_nopar:Npn \__ctex_ltj_um_define_math_chars:
711 {
712   \group_begin:
713   \cs_set_protected:Npn \__um_sym:nnn ##1##2##3
714   {
715     \tl_if_in:nnT
716     {
717       \mathord \mathalpha \mathbin \mathrel
718       \mathpunct \mathop \mathfence
719     }
720     { ##3 }
721     { \__ctex_ltj_um_char:Nn ##2 { ##1 } }
722   }
723   \__um_input_math_symbol_table:
724   \group_end:
725 }
726 \cs_new_protected:Npn \__ctex_ltj_um_char:Nn #1#2
727 {
728   \__ctex_ltj_um_char_aux:Nx #1 { \char_generate:nn {#2} { 12 } }
729   \ltjsetmathletter {#2}
730 }
731 \cs_new_protected:Npn \__ctex_ltj_um_char_aux:Nn #1#2
732 {
733   \cs_gset_protected_nopar:Npn #1
734   {
735     \mode_if_math:TF
736     {#2}

```

```

737     {
738     {
739         \lua_now_x:n { tex.globaldefs = 0 }
740         \ltj@allalchar #2
741     }
742     }
743 }
744 }
745 \cs_generate_variant:Nn \__ctex_ltj_um_char_aux:Nn { Nx }
746 \ctex_at_end_package:nn { unicode-math }
747 {
748     \cs_set_eq:NN \__um_define_math_chars: \__ctex_ltj_um_define_math_chars:
749     \@ifpackagelater { unicode-math } { 2015/06/28 }
750     {
751         \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
752         \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
753         { \__um_switchto_literal: }
754     }
755     { }
756 }

```

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。

```

757 \ctex_at_end_package:nn { listings }
758 {
759     \use:x
760     {
761         \exp_not:N \RequirePackage { lltjp-listings }
762         \tl_set:Nn \exp_not:N \lstlistingname
763         { \exp_not:o { \lstlistingname } }
764         \tl_set:Nn \exp_not:N \lstlistlistingname
765         { \exp_not:o { \lstlistlistingname } }
766     }
767 }

```

15.4.3.3 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。

```

768 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
769 {
770     \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
771     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
772 }
773 \tl_new:N \CJK@family
774 \tl_new:N \l__ctex_ltj_current_font_tl
775 \tl_set:Nn \l__ctex_ltj_current_font_tl
776 { \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。

```

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

```

787 \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
788   { \__ctex_ltj_select_font_aux: }
789 }
790 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
791   {
792   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
793   \cs_gset_nopar:Npx \font@name {#1}
794   }
795 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
796   { \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`。

```

797 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
798   {
799   \exp_after:wN \cs_if_exist:NF \font@name
800   {
801   \group_begin:
802   \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
803   \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
804   \define@newfont
805   \group_end:
806   }
807 }
808 \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:

```

%%%%\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`。

```

809 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
810   {
811   \get@external@font
812   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
813   {
814   \tl_set:Nx \external@font
815   { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
816   }
817   \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。

```

818     \font@name
819     \lua_now_x:n { font.current(tex.getattribute('ltj@curjfnt')) }
820     \use:c { \f@encoding + \f@family }
821     \use:c { \curr@fontshape }
822   }

```

(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。

```

823 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
824 {
825   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
826   {
827     \group_begin:
828     \tl_set_eq:NN \CJK@family \f@family
829     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
830     {
831       \cs_gset_protected_nopar:Npx \subst@correction
832       {
833         \cs_new_eq:NN
834         \exp_not:c { \l__ctex_ltj_current_font_tl }
835         \font@name
836       }
837       \group_insert_after:N \group_insert_after:N
838       \group_insert_after:N \subst@correction
839     }
840     \group_end:
841   }
842 }

```

(End definition for `\ctex_ltj_subst_font:`)

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

```

843 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
844 {
845   \lua_now_x:n { luatexja.jfont.does_alt_set ('lua_escape_x:n {#1}') }
846   \prg_return_true: \else: \prg_return_false: \fi:
847 }

```

(End definition for `\ctex_ltj_if_alternate_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, ';ltjaltpfont' .. 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。

```

848 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
849   { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for __ctex_ltj_patch_external_font:w)

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

```

850 \cs_new_protected_nopar:Npn \ctex_ltj_select_alterfont:
851   {
852     \ctex_ltj_if_alterfont_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
853     {
854       \lua_now_x:n
855       {
856         luatexja.jfont.output_alt_font_cmd
857         ('y', '\lua_escape_x:n { \l__ctex_ltj_current_shape_tl }')
858       }
859       \lua_now_x:n { luatexja.jfont.pickup_alt_font_a ('f@size') }
860     }
861   }
862 \tl_new:N \l__ctex_ltj_current_shape_tl
863 \tl_set:Nn \l__ctex_ltj_current_shape_tl
864   { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for \ctex_ltj_select_alterfont:.)

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

```

865 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
866   {
867     \cs_if_exist:cF { #1/f@size }
868     {
869       \group_begin:
870       \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
871       \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
872       \ctex_ltj_pickup_font:
873       \group_end:
874       \__ctex_ltj_pop_fontname:
875     }
876   }

```

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

877 <@@=>

\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。

```

878 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
879   {
880     \ltj@@getjfontnumber #1
881     \lua_now_x:n
882     {
883       luatexja.jfont.pickup_alt_font_b
884       ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
885     }
886   }

```

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

15.4.3.4 数学字体族

887 <@@=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 L^AT_EX_{2_ε} 编码名称或者字体命令。L^AT_EX_{2_ε} 字体命令的一般形式是:

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

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

```
888 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
889 {
890   \lua_now_x:n
891   { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[~/]+' ) ) }
892   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
893 }
```

(End definition for $\backslash ctex_ltj_if_jfont:nTF$.)

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

```
894 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
895 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
896 \group_begin:
897   \char_set_catcode_other:N M
898   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
899     { \ctex_ltj_if_jfont:nTF {#3} }
900 \group_end:
```

(End definition for $\backslash ctex_ltj_if_jfont_math:NTF$.)

901 $\langle @@=\rangle$

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

```
902 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
903 {
904   \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
905     { \ctex_ltj_get_and_define_fonts_ja:nN }
906     { \ctex_ltj_get_and_define_fonts_al:nN }
907     {#1} #2
908 }
909 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
910 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
911 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
912 {
913   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
914   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
915   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
916   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
917   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
918   \ctex_ltj_pickup_font:
919   \tl_put_right:Nx \math@fonts
920     {
921       \ltj@setpar@global
922       \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
923       \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
924       \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
925     }
926 }
```

(End definition for $\backslash getanddefine@fonts$ and $\backslash ctex_ltj_get_and_define_fonts:nN$.)

927 $\langle @@=ctex_ltj\rangle$

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

```
\ctex_ltj_use_math_group:Nn
928 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
929 {
930   \mode_if_math:T
```

```

931     {
932       \math@bgroup
933       \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
934       \ctex_ltj_math_group_hook:
935       \ctex_ltj_if_jfont_math:NTF #1
936       { \jfam } { \mathgroup } #2 \scan_stop:
937       \math@egroup
938     }
939   }
940 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
941 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

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

15.4.3.5 字体族的定义与使用

```

\ctex_mono_jfm:n LuaTeX-ja 中与标点格式 plain 对应的 JFM 是 mono。
\l__ctex_ltj_jfm_tl
942 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
943   {
944     \str_if_eq:nnTF {#1} { plain }
945     { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
946     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
947   }
948 \tl_new:N \l__ctex_ltj_jfm_tl
949 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
950 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

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

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

```

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

```

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

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

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

```


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

```

976 \keys_define:nn { fontspec-preparse-external }
977   {
978     NoEmbed .code:n =
979       { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
980   }
981 \cs_new:Npn \__ctex_ltj_noembed_wrap:n #1 { psft: #1 }

```

`\ctex_ltj_set_family:nnn` 将自定义的字体族名与 `fontspec` 实际设置的名字对应起来。

```

982 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
983   {
984     \group_begin:
985     \clist_clear:N \l__ctex_ltj_char_range_clist
986     \seq_clear:N \l__ctex_ltj_alternate_seq
987     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
988     \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
989     \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
990     \ctex_ltj_set_alternate_family:nnF {#1} {#3}
991     {
992       \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
993       \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
994         {#1} { \l__ctex_ltj_font_options_clist }
995       \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
996       \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
997       \__ctex_ltj_gset_family_cs:nn {#1} {#3}
998     }
999     \group_end:
1000   }
1001 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1002   { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1003 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1004 \clist_new:N \l__ctex_ltj_font_options_clist
1005 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1006   {
1007     \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1008     \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1009   }

```

(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
1010 \prop_new:N \g__ctex_ltj_family_name_prop
1011 \prop_new:N \g__ctex_ltj_family_font_name_prop
1012 \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.)

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

```

1013 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1014   {
1015     \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1016     {
1017       \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1018       \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1019       \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1020       {
1021         \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1022         \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1023         \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1024         \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1025       }
1026       \msg_warning:nxxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }

```

```

1027     }
1028   }
1029   \tl_new:N \l__ctex_ltj_tmp_tl
1030   \msg_new:nnn { ctex } { redefine-family }
1031   { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#2). }

```

(End definition for __ctex_ltj_check_family:n.)

__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。

```

1032 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1033 {
1034   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }
1035   {
1036     \group_begin:
1037     \__ctex_ltj_change_encoding:
1038     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1039     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1040     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1041     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1042     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1043     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1044     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1045     \__ctex_ltj_set_alternate_family:n {#1}
1046     \group_end:
1047   }
1048 }
1049 \tl_new:N \l__ctex_ltj_base_family_tl
1050 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1051 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1052 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1053 {
1054   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1055   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1056   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1057   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1058 }
1059 \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

(End definition for __ctex_ltj_gset_family_cs:nn.)

\CJKfamily 切换字体。

```

1060 \NewDocumentCommand \CJKfamily { m }
1061 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1062 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1063 {
1064   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1065   {
1066     \tl_set:Nn \l__ctex_ltj_family_tl {#1}
1067     \selectfont
1068   }
1069   { \__ctex_ltj_family_unknown_warning:n {#1} }
1070 }
1071 \tl_new:N \l__ctex_ltj_family_tl
1072 \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。

```

1073 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1074 {
1075   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1076   { \prg_return_true: }

```

```

1077     {
1078       \cs_if_exist_use:CTF { \__ctex_ltj_family_csname:n {#1} }
1079       {
1080         \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1081         \prg_return_true:
1082       }
1083       { \prg_return_false: }
1084     }
1085   }
1086 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

__ctex_ltj_family_unknown_warning:n

```

1087 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1088 {
1089   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1090   {
1091     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1092     {
1093       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1094       \msg_warning:nnn { ctex } { family-unknown } {#1}
1095     }
1096   }
1097 }
1098 \seq_new:N \g__ctex_ltj_unknown_family_seq
1099 \msg_new:nnn { ctex } { family-unknown }
1100 {
1101   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1102   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1103 }
1104 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1105 {
1106   \str_case_x:nnF {#1}
1107   {
1108     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1109     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1110     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1111   }
1112   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1113   [...] \{...\}
1114 }
1115 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1116 {
1117   \str_case_x:nnF {#1}
1118   {
1119     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1120     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1121     \CJKttdefault { \token_to_str:N \CJKttdefault }
1122   }
1123   {#1}
1124 }

```

(End definition for __ctex_ltj_family_unknown_warning:n.)

\ctex_ltj_fontspec:nn

```

1125 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1126 {
1127   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1128   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1129   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1130   {
1131     \int_gincr:N \g__ctex_ltj_family_int
1132     \__ctex_ltj_fontspec:xnn
1133     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }

```

```

1134     {#1} {#2}
1135   }
1136 }
1137 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1138 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1139 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1140 {
1141   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1142   {
1143     \cs_if_free:cF
1144     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1145     {
1146       \cs_gset_eq:cc
1147       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1148       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1149       \cs_gset_eq:cc
1150       { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1151       { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1152     }
1153     \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1154   }
1155   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1156   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1157   \ctex_ltj_switch_family:n {#1}
1158 }
1159 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1160 \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`

```

1161 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1162 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1163 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1164 {
1165   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1166   {#1} \l__ctex_ltj_tmp_tl
1167   {
1168     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1169     {#1} \l__ctex_ltj_font_options_clist
1170     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1171     \bool_set_true:N \l__ctex_ltj_add_alternate_bool
1172     \ctex_ltj_fontspec:xx
1173     { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1174     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1175   }
1176   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1177 }
1178 \bool_new:N \l__ctex_ltj_add_alternate_bool
1179 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1180 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1181 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1182 {
1183   \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1184   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1185 }

```

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

`\setCJKfamilyfont`

`\newCJKfontfamily`

`\CJKfontspec`

`\addCJKfontfeatures`

```

1186 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
1187 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1188 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
1189 {
1190   \tl_set:Nx \l__ctex_ltj_tmp_tl

```

```

1191     { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1192     \cs_new_protected_nopar:Npx #2
1193     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1194     \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1195   }
1196   \NewDocumentCommand \CJKfontspec { 0 { } m }
1197   {
1198     \ctex_ltj_fontspec:xx {#1} {#2}
1199     \tex_ignorespaces:D
1200   }
1201   \NewDocumentCommand \addCJKfontfeatures { m }
1202   {
1203     \ctex_ltj_add_font_features:x {#1}
1204     \tex_ignorespaces:D
1205   }
1206   \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

(End definition for \setCJKfamilyfont and others.)

```

\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1207 \NewDocumentCommand \setCJKmainfont { 0 { } m }
1208 {
1209   \ctex_ltj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1210   \normalfont
1211 }
1212 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1213 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1214 {
1215   \ctex_ltj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1216   \normalfont
1217 }
1218 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1219 {
1220   \ctex_ltj_set_family:xxx { \CJKttdefault } {#1} {#2}
1221   \normalfont
1222 }
1223 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1224 { \ctex_ltj_set_family:xxx { \c__ctex_ltj_math_tl } {#1} {#2} }
1225 \NewDocumentCommand \defaultCJKfontfeatures { m }
1226 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1227 \clist_new:N \g__ctex_ltj_default_features_clist
1228 \@onlypreamble \setCJKmainfont
1229 \@onlypreamble \setCJKsansfont
1230 \@onlypreamble \setCJKmonofont
1231 \@onlypreamble \setCJKmathfont
1232 \@onlypreamble \setCJKromanfont
1233 \@onlypreamble \defaultCJKfontfeatures

```

(End definition for \setCJKmainfont and others.)

```

1234 \tl_if_exist:NF \CJKfamilydefault
1235 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1236 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1237 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1238 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1239 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1240 { \CJKfamily { \CJKrmdefault } }
1241 { }
1242 { \ctex_patch_failure:N \rmfamily }
1243 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1244 { \CJKfamily { \CJKsfdefault } }
1245 { }
1246 { \ctex_patch_failure:N \sffamily }
1247 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1248 { \CJKfamily { \CJKttdefault } }
1249 { }

```

```

1250 { \ctex_patch_failure:N \ttfamily }
1251 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1252 { \CJKfamily { \CJKfamilydefault } }
1253 { \cs_set_eq:NN \reset@font \normalfont }
1254 { \ctex_patch_failure:N \normalfont }

```

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

```

1255 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1256 {
1257   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1258   {
1259     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmpa_tl
1260     {
1261       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1262       { \use:n }
1263       {
1264         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmpa_tl
1265         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1266         { \use:n }
1267       }
1268       {
1269         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1270         {
1271           \prop_map_break:n
1272           { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1273         }
1274       }
1275     }
1276     \normalfont
1277     \ctex_ltj_update_mathfont:
1278   }
1279 }

```

(End definition for \ctex_ltj_ensure_default_family:.)

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

```

1280 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1281 {
1282   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1283   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1284   {
1285     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1286     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1287   }
1288 }
1289 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1290 {
1291   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1292   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1293   { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1294   \cs_if_free:cTF
1295   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1296   {
1297     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1298     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1299   }
1300   {
1301     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1302     { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1303   }
1304 }

```

(End definition for \ctex_ltj_update_mathfont:.)

15.4.3.6 替代字体的设置

AlternateFont 设置替代字体的选项。

CharRange

```

1305 \keys_define:nn { ctex_ltj / fontspec }
1306   {
1307     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1308     AlternateFont .value_required:n = true ,
1309     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1310     CharRange .value_required:n = true
1311   }

```

(End definition for *AlternateFont* and *CharRange*. These functions are documented on page 24.)

`\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。

```

1312 \group_begin:
1313   \char_set_catcode_other:N \|
1314   \cs_set:Npn \l__ctex_ltj_tmp:w #1
1315     {
1316       \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1317         {
1318           \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1319             {
1320               \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1321               \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1322               \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1323               \seq_set_filter:NnN \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1324                 { ! \tl_if_blank_p:n { ##### } }
1325               \seq_concat:NNN \l__ctex_ltj_alternate_seq
1326                 \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1327             }
1328         }
1329     }
1330   \char_set_catcode_active:N \|
1331   \l__ctex_ltj_tmp:w { || }
1332 \group_end:
1333 \seq_new:N \l__ctex_ltj_tmp_seq
1334 \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。

```

1335 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1336   {
1337     \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1338       {
1339         \__ctex_ltj_check_family:n {#1}
1340         \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1341           { \ctex_ltj_save_alternate_seq:cn { \l__ctex_ltj_alternate_cs:n {#1} } {#2} }
1342         #3
1343       }
1344     { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1345   }

```

(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`

```

1346 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1347   {
1348     \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1349       { \ctex_ltj_save_alternate_seq:Nnnwn #1 {#2} ##1 { } \q_stop }
1350   }

```



```

1351 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1352 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1353 { m m m +0{ } m u{ \q_stop } }
1354 {
1355   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1356   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1357   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1358   \tl_if_blank:nTF {#5}
1359     { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1360     {
1361       \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1362       \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1363     }
1364   \use:x
1365   {
1366     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1367     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1368     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1369     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1370   }
1371 }
1372 \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。

```

1373 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1374 {
1375   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1376   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1377   \ctex_ltj_set_alternate_family:coonn
1378   { \__ctex_ltj_alternate_cs:n {#1} }
1379   { \l__ctex_ltj_char_range_clist }
1380   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1381 }
1382 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1383 {
1384   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1385   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1386   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1387 }
1388 \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。

```

1389 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1390 {
1391   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1392   \cs_gset_protected_nopar:Npx #1
1393   { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1394 }

```

(End definition for \ctex_ltj_save_alternate_family:Nnnn.)

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

```

1395 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1396 {
1397   \group_begin:
1398   \__ctex_ltj_change_encoding:
1399   \cs_set_eq:NN \CJKfamily \use_none:n
1400   \ctex_ltj_swap_cs:NN

```

```

1401     \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1402     \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1403     \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1404     \group_end:
1405   }
1406   \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。

```

1407 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1408 {
1409   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1410   \cs_set_eq:NN #1 #2
1411   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1412   \cs_undefine:N \__ctex_ltj_tmp:w
1413 }

```

(End definition for \ctex_ltj_swap_cs:NN.)

LTJFONTUID 在 `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。

__ctex_ltj_update_family_uid:N

```

1414 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1415 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1416 {
1417   \int_gincr:N \g__ctex_ltj_family_int
1418   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1419 }
1420 \int_new:N \g__ctex_ltj_family_int

```

(End definition for `LTJFONTUID` and __ctex_ltj_update_family_uid:N.)

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

```

1421 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1422 {
1423   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1424   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1425   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1426   { \l_fontspec_family_tl } {#3} {#4}
1427 }

```

(End definition for \ctex_ltj_declare_alternate_shape:nnnnnn.)

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

```

1428 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1429 {
1430   \clist_map_inline:Nn #1
1431   {
1432     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1433     {
1434       \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1435       \l__ctex_ltj_char_range_tl
1436     }
1437     { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1438   }
1439   \__ctex_ltj_save_alternate_shape:cnn
1440   { \__ctex_ltj_alternate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }

```

```

1441     { luatexja.jfont.clear_alt_font_latex }
1442     { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1443 }

```

(End definition for `\ctex_ltj_set_alternate_shape:Nnnnnnn`.)

`\ctex_ltj_set_alternate_shape:nnn` 我们使用 `->` 而不是像 LuaTeX-ja 一样使用 `-` 作为区间的分隔符。LuaTeX-ja 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。

```

1444 \NewDocumentCommand \ctex_ltj_set_alternate_shape:nnn
1445 { m m > { \SplitArgument { \c_one } { -> } } m }
1446 { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
1447 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
1448 {
1449   \ctex_ltj_set_alternate_shape:n
1450   {
1451     \IfNoValueTF {#4}
1452     { \int_eval:n {#3} , \int_eval:n {#3} , }
1453     {
1454       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1455       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1456     }
1457     '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1458     '\lua_escape_x:n { \CJK@encoding/#1 }'
1459   }
1460 }
1461 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1462 {
1463   \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1464   \__ctex_ltj_save_alternate_shape:cnn
1465   { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1466   { luatexja.jfont.set_alt_font_latex } {#1}
1467 }

```

(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。

```

1468 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1469 {
1470   \tl_map_inline:Nn #3
1471   {
1472     \ctex_ltj_set_alternate_shape:n
1473     {
1474       ##1 ,
1475       '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1476       '\lua_escape_x:n { \CJK@encoding/#1 }'
1477     }
1478   }
1479 }

```

(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。

```

1480 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:nnn #1#2#3
1481 {
1482   \group_begin:
1483   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1484   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1485   \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1486   \cs_gset_protected_nopar:Npx #1

```

```

1487     { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1488   \group_end:
1489 }
1490 \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`

```

1491 \keys_define:nn { ctex }
1492 {
1493   clearalternatefont .code:n =
1494     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1495   resetalternatefont .code:n =
1496     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1497   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1498   resetalternatefont .default:n = \l_ctex_ltj_family_tl
1499 }
1500 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1501 {
1502   \group_begin:
1503     \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1504     {
1505       \cs_if_exist_use:cT { \_ctex_ltj_alternate_cs:n { clear / #1 } }
1506       {
1507         \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1508           {#1} { \l__ctex_ltj_base_family_tl }
1509         \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1510         \selectfont
1511       }
1512     }
1513     { \_ctex_ltj_family_unknown_warning:n {#1} }
1514   \group_end:
1515 }
1516 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1517 {
1518   \group_begin:
1519     \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1520     {
1521       \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1522       \use:c { \_ctex_ltj_alternate_cs:n { reset / #1 } }
1523       \selectfont
1524     }
1525   \group_end:
1526 }
1527 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1528 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalternatefont` and `resetalternatefont`. These functions are documented on page 25.)

`declarecharrange` 预先声明字符范围。

```

1529 \keys_define:nn { ctex }
1530 {
1531   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1532   declarecharrange .value_required:n = true
1533 }
1534 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1535 { \clist_map_inline:nn {#1} { \_ctex_ltj_declare_char_range:nn ##1 } }
1536 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1537 \cs_new_protected_nopar:Npn \_ctex_ltj_declare_char_range:nn #1#2
1538 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

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

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

`\g__ctex_ltj_char_range_prop`

```

1539 \cs_new_protected_nopar:Npn \ctex_ljt_declare_char_range:nn #1#2
1540 {
1541   \tl_clear:N \l__ctex_ljt_char_range_tl
1542   \clist_map_function:nN {#2} \ctex_ljt_save_char_range:n
1543   \prop_gput:Nno \g__ctex_ljt_char_range_prop {#1} { \l__ctex_ljt_char_range_tl }
1544   \ctex_ljt_def_char_range_key:n {#1}
1545   \tl_clear:N \l__ctex_ljt_char_range_tl
1546 }
1547 \tl_new:N \l__ctex_ljt_char_range_tl
1548 \prop_new:N \g__ctex_ljt_char_range_prop

(End definition for \ctex_ljt_declare_char_range:nn and \g__ctex_ljt_char_range_prop.)

```

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

```

1549 \NewDocumentCommand \ctex_ljt_save_char_range:n
1550 { > { \SplitArgument { \c_one } { -> } } m }
1551 { \ctex_ljt_save_char_range:nn #1 }
1552 \cs_new_protected_nopar:Npn \ctex_ljt_save_char_range:nn #1#2
1553 {
1554   \tl_put_right:Nx \l__ctex_ljt_char_range_tl
1555   { {
1556     \IfNoValueTF {#2}
1557     { \int_eval:n {#1} , \int_eval:n {#1} }
1558     {
1559       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1560       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1561     }
1562   } }
1563 }

(End definition for \ctex_ljt_save_char_range:n.)

```

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

```

1564 \cs_new_protected_nopar:Npn \ctex_ljt_def_char_range_key:n #1
1565 {
1566   \keys_if_exist:nnF { ctex_ljt / fontspec } {#1}
1567   {
1568     \keys_define:nn { ctex_ljt / fontspec }
1569     { #1 .code:n = \ctex_ljt_char_range_key:nn {#1} { ##1 } }
1570   }
1571 }

(End definition for \ctex_ljt_def_char_range_key:n.)

```

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

```

1572 \cs_new_protected:Npn \ctex_ljt_char_range_key:nn #1#2
1573 {
1574   \tl_if_blank:nTF {#2}
1575   { \tl_set:Nn \l__ctex_ljt_char_range_clist {#1} }
1576   {
1577     \clist_if_empty:NT \l__ctex_ljt_char_range_clist
1578     {
1579       \tl_set:Nn \l__ctex_ljt_tmp_tl { {#1} }
1580       \__ctex_ljt_char_range_parse_feature:w #2 \q_stop
1581     }
1582   }
1583 }

(End definition for \ctex_ljt_char_range_key:nn.)

```

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

```

1584 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1585 { +o o u { \q_stop } }
1586 {
1587   \exp_args:Nnf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1588   {
1589     \IfNoValueTF {#1} { {#3} }
1590     {
1591       \IfNoValueTF {#2}
1592       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ {#1} ] {#3} } }
1593       { [ {#1} ] { [ #2 ] } }
1594     }
1595   }
1596   \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1597 }

```

(End definition for __ctex_ltj_char_range_parse_feature:w)

15.4.3.7 其它设置

`\ctex_ltj_update_xkanjiskip:` `\ltjsetparameter` 对 `xkanjiskip` 是即时赋值。`\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`, 最后的设置会影响全段。

```

1598 \cs_new_protected_nopar:Npn \ctex_ltj_update_xkanjiskip:
1599 {
1600   \skip_if_eq:nnT
1601   { \ltjgetparameter { xkanjiskip } } { \l__ctex_ltj_xkanjiskip_skip }
1602   {
1603     \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }
1604     \ltjsetxkanjiskip \l__ctex_ltj_xkanjiskip_skip
1605   }
1606 }

```

注意, 此时还没有实际设置字体, 所以 `\zw` 是 `0pt`。

```

1607 \skip_new:N \l__ctex_ltj_xkanjiskip_skip
1608 \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }

```

(End definition for `\ctex_ltj_update_xkanjiskip:` and `\l__ctex_ltj_xkanjiskip_skip`.)

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

```

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

1609 \AtBeginDocument
1610 {
1611   \ctex_appto_cmd:NnnTF \verbatim@font
1612   { \char_set_catcode_letter:n { 64 } }
1613   { \CTEX@verbatim@font@hook }
1614   { }
1615   { \ctex_patch_failure:N \verbatim@font }
1616 }

```

```

1617 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1618 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1619 <@@=>

\@@italiccorr LATEX 的倾斜校正也要重新定义。
1620 \cs_set_eq:NN \@@italiccorr \/

(End definition for \@@italiccorr.)
1621 <@@=ctex>
1622 </luatex>

```

15.4.4 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。

```

1623 <*pdfTeX|luatex>
1624 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1625 {
1626   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1627   {
1628     \group_begin:
1629     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1630     \tl_gset:Nx \CJKfamilydefault
1631     {
1632       \str_case:onF { \familydefault }
1633       {
1634         { \rmdefault } { \exp_not:N \CJKrmdefault }
1635         { \sfdefault } { \exp_not:N \CJKsfdefault }
1636         { \ttdefault } { \exp_not:N \CJKttdefault }
1637       }
1638       { \CJKfamilydefault }
1639     }
1640     \group_end:
1641   }

```

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

```

1642 <*luatex>
1643   \ctex_ltj_ensure_default_family:
1644 </luatex>
1645 }
1646 \ctex_at_end_preamble:n { \ctex_update_default_family: }

(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。

```

1647 \tl_new:N \l__ctex_family_default_init_tl
1648 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1649 \tl_set:Nx \l__ctex_family_default_init_tl
1650 {
1651   \exp_not:N \__ctex_family_default_wrap:n
1652   { \exp_not:o { \CJKfamilydefault } }
1653 }
1654 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1655 </pdfTeX|luatex>

(End definition for \l__ctex_family_default_init_tl.)

```

15.4.5 操作系统的判断

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

```

1656 <(*luatex)
1657 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1658 {
1659   \tl_gset:Nx \g__ctex_fontset_tl
1660   {
1661     \lua_now_x:n
1662     {
1663       if ~ os.name == 'windows' then ~
1664         tex.sprint ( 'windows' )
1665       elseif ~ os.name == 'macosx' then ~
1666         tex.sprint ( 'mac' )
1667       else ~
1668         tex.sprint ( 'fandol' )
1669       end
1670     }
1671   }
1672 }
1673 </(*luatex)

```

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!

```

1674 <(*xetex|pdfTeX)
1675 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1676 {
1677   \file_if_exist:nTF { /dev/null }
1678   {
1679     \file_if_exist:nTF { nul: }
1680     {
1681       \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1682       { \ctex_if_macosx:TF { mac } { windows } }
1683       { \ctex_if_macosx:TF { mac } { fandol } }
1684     }
1685     { \ctex_if_macosx:TF { mac } { fandol } }
1686   }
1687   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1688 }
1689 \str_const:Nx \c__ctex_upper_case_file_str
1690 { \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。

```

1691 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1692 {
1693   \file_if_exist:nTF { \c__ctex_macosx_file_str }
1694   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1695   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1696 }
1697 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1698 </(*xetex|pdfTeX)

```

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

15.4.6 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_hyersetup: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_hyersetup: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。

```

1699 <*class|ctex>
1700 \@ifpackageloaded { hyperref }
1701 {
1702   \cs_new_protected_nopar:Npn \ctex_hyersetup:n #1
1703     { \hyersetup {#1} }
1704 }
1705 {
1706   \cs_new_protected_nopar:Npn \ctex_hyersetup:n #1
1707     { \PassOptionsToPackage {#1} { hyperref } }
1708 }
1709 </class|ctex>

```

(End definition for `\ctex_hyersetup: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。

```

1710 <*pdfTeX>
1711 \ctex_hyersetup:n { driverfallback = dvipdfmx }
1712 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1713 {
1714   \ctex_hyersetup:n { CJKbookmarks = true }
1715   \sys_if_output_pdf:TF
1716     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1717     {
1718       \ctex_at_end_package:nn { hyperref }
1719       {
1720         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1721         {
1722           \AtBeginShipoutFirst
1723             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1724         }
1725         { \RequirePackage { xCJK2uni } }
1726       }
1727     }
1728 }
1729 { \ctex_hyersetup:n { pdfencoding = unicode } }
1730 </pdfTeX>

```

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

```

\ifHy@unicode
  \HyPsd@ConvertToUnicode#1%
  \ifx\HyPsd@pdfencoding\HyPsd@pdfencoding@auto
    \ltx@ifundefined{StringEncodingConvertTest}{%
    }{%
      \EdefUnescapeString\HyPsd@temp#1%
      \ifxetex
        \let\HyPsd@UnescapedString\HyPsd@temp
        \StringEncodingConvertTest\HyPsd@temp\HyPsd@temp
          {utf16be}{ascii-print}{%
          \EdefEscapeString\HyPsd@temp\HyPsd@temp
          \global\let#1\HyPsd@temp
          \HyPsd@EscapeTeX#1%
          \Hy@unicodefalse
          }{%

```

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

```

\HyPsd@ToBigChars#1%
}%

```

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

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

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

```

1731 (*xetex|luatex)
1732 \ctex_hypersetup:n { pdfencoding = unicode }
1733 </xetex|luatex>

1734 (*pdfetex|xetex|luatex)

```

15.4.7 CJKfntef、xeCJKfntef 相关设置

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

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

```

1735 (*pdfetex)
1736 \RequirePackage { CJKfntef }
1737 \normalem
1738 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1739 { \tl_clear:c { CJK#1color } }
1740 </pdfetex>
1741 (*xetex)
1742 \RequirePackage { xeCJKfntef }
1743 \ifpackagealater { xeCJKfntef } { 2014/11/04 }
1744 {
1745   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1746   { \xeCJKsetup { #1 / format = { } } }
1747 }
1748 {
1749   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1750   { \tl_clear:c { CJK#1color } }
1751 }
1752 </xetex>
1753 (*luatex)
1754 \msg_new:nnn { ctex } { fntef-not-available }
1755 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1756 \msg_warning:nn { ctex } { fntef-not-available }
1757 </luatex>
1758 \clist_map_inline:nn
1759 { underdot , underline , underdblline , underwave , sout , xout }
1760 (*pdfetex|xetex)
1761 {
1762   \__ctex_clear_fntef_color:n {#1}
1763   \cs_new_protected_nopar:cpx { CTEX#1 }
1764   {
1765     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }

```

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

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

```

1766         { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1767         \exp_not:c { CJK#1 }
1768     }
1769 }
1770 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1771 {
1772     \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1773     { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1774     \CJKfilltwosides
1775 }
1776 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1777 </pdfTeX>
1778 <*LaTeX>
1779 { \cs_new_eq:cN { CTEX#1 } \use:n }
1780 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1781 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1782 </LaTeX>
1783 <*pdfTeX>
1784 \clist_map_inline:nn
1785 {
1786     underdotbasesep , underdotsep , underlinebasesep ,
1787     underlinesep , underdbl_linesep , underdbl_linesep ,
1788     underwavebasesep , underwavesep , southeight ,
1789     underdotcolor , underwavecolor , underlinecolor ,
1790     underdbl_linecolor , soutcolor , xoutcolor
1791 }
1792 {
1793     \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1794     \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1795 }
1796 </pdfTeX>

```

15.4.8 \ccwd 的更新

\ctex_update_ccwd:

```

\ccwd
1797 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1798 <*pdfTeX>
1799 {
1800     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1801     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1802 }
1803 </pdfTeX>
1804 <*LaTeX>
1805 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1806 </LaTeX>
1807 \dim_new:N \ccwd

```

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

\ctex_update_ccglue: 更新字间距。

```

1808 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1809 <*pdfTeX>
1810 {
1811     \cs_set_protected_nopar:Npn \CJKglue
1812     { \skip_horizontal:N \l__ctex_ccglue_skip }
1813 }
1814 </pdfTeX>
1815 <*XeTeX>
1816 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1817 </XeTeX>
1818 <*LaTeX>
1819 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1820 </LaTeX>
1821 \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
1822 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1823 {
1824   <pdfTeX>
1825   \if_meaning:w \CJKglue \_ctex_ccglue:
1826   \prg_return_false: \else: \prg_return_true: \fi:
1827 </pdfTeX>
1828 <luaTeX>
1829   \skip_if_eq:nnTF { \l_&ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1830   { \prg_return_false: } { \prg_return_true: }
1831 </luaTeX>
1832 }
```

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 `@@`。

```
1833 <pdfTeX>
1834 \ctex_at_end:n { \cs_new_eq:NN \_ctex_ccglue: \CJKglue }
1835 </pdfTeX>
```

(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 `1 em`, 因为这时的 `1 em` 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。

```
1836 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1837 <pdfTeX> { \dim_set:Nn \ccwd { \f@size \p@ } }
1838 <luaTeX> { \dim_set:Nn \ccwd { \zw } }
```

(End definition for `\ctex_update_em_unit:`.)

15.4.9 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`。 `everysel` 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。

```
1839 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1840 {
1841   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1842   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1843 }
1844 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1845 \cs_new_eq:Nc \_ctex_save_selectfont: { selectfont ~ }
1846 \ctex_preto_cmd:NnnTF \selectfont { \ExplSyntaxOff }
1847 { \CTEX@selectfont@hook }
1848 {
1849   \tl_put_left:Nn \@EverySelectfont@Init
1850   { \cs_set_eq:cN { selectfont ~ } \_ctex_save_selectfont: }
1851 }
1852 { \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。

```

1853 <*pdfTeX>
1854 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1855 </pdfTeX>
1856 <*xetex|luatex>
1857 \EverySelectfont { \CTEX@selectfont@hook }
1858 </xetex|luatex>

```

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。

```

1859 <*luatex>
1860 \ctex_add_to_selectfont:n
1861 {
1862   \ctex_ltj_select_font:
1863   \ctex_ltj_select_alternate_font:
1864   \ctex_ltj_update_xkanjiskip:
1865 }
1866 \tl_set:Nn \CJK@family { song } \selectfont
1867 \tl_clear:N \CJK@family
1868 </luatex>

```

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

```

1869 \keys_define:nn { ctex }
1870 {
1871 <*pdfTeX|xetex>
1872   space .choice: ,
1873   space / true .code:n =
1874 <pdfTeX>   { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
1875 <xetex>    { \xeCJKsetup { CJKspace = true } } ,
1876   space / auto .code:n =
1877 <pdfTeX>   { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
1878 <xetex>    { \xeCJKsetup { CJKspace = false } } ,
1879   space / false .code:n =
1880 <pdfTeX>   { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
1881 <xetex>    { \xeCJKsetup { CJKspace = false } } ,
1882   space .default:n = { true } ,
1883   space .initial:n = { auto }
1884 </pdfTeX|xetex>
1885 <*luatex>
1886   space .code:n =
1887   { \msg_warning:nn { ctex } { invalid-option } }
1888 </luatex>
1889 }

```

(End definition for space. This function is documented on page 27.)

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

```

1890 \keys_define:nn { ctex }
1891 {
1892   punct .code:n =
1893   {
1894     \tl_set:Nx \l__ctex_punct_tl { #1 }
1895 <pdfTeX>   \punctstyle { \l__ctex_punct_tl }
1896 <xetex>    \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
1897 <luatex>   \ctex_mono_jfm:o { \l__ctex_punct_tl }
1898   } ,
1899   punct .default:n = { quanjiao } ,
1900 }

```

(End definition for punct. This function is documented on page 27.)

X_YLaTeX 和 LuaLaTeX 总是使用 UTF8 编码。

```

1901 <*xetex|luatex>
1902 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
1903 </xetex|luatex>

1904 </pdfxetex|xetex|luatex>
1905 <*class|ctex>

```

15.4.10 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

1906 \ctex_file_input:n { \c__ctex_engine_file_str }

```

15.5 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。

```

1907 \keys_define:nn { ctex }
1908 {
1909   autoindent .choice: ,
1910   autoindent .default:n = { true } ,
1911   autoindent / true .code:n =
1912   {
1913     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
1914     \ctex_select_size:
1915   } ,
1916   autoindent / false .code:n =
1917   { \tl_clear:N \l__ctex_autoindent_tl } ,
1918   autoindent / unknown .code:n =
1919   {
1920     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
1921     \ctex_select_size:
1922   }
1923 }

```

(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。

```

1924 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
1925 \cs_new_protected_nopar:Npn \ctex_select_size:
1926 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

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

`\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。

```

1927 \cs_new_protected_nopar:Npn \ctex_update_size:
1928 {
1929   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
1930   {
1931     \ctex_update_stretch:
1932     \ctex_update_parindent:
1933   }
1934   { \ctex_update_ziju: }
1935 }
1936 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
1937 \tl_new:N \l__ctex_ziju_tl
1938 \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。

```

1939 \ctex_add_to_selectfont:n
1940 { \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`。

```

1941 \keys_define:nn { ctex }
1942 {
1943   linestretch .code:n =
1944   {
1945     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
1946     \ctex_select_size:
1947   } ,
1948   linestretch .value_required:n = true
1949 }
1950 \tl_new:N \l__ctex_line_stretch_tl
1951 \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。 ϵ -TeX 的除法是四舍五入，而我们这里应该用截断。由于没有可展性的要求，直接用原语 `\tex_divide:D` 要比 `\int_div_truncate:nn` 快一些。下面的算法还兼顾到了 `\linewidth` 不为汉字字宽的整数倍的情况。若用户禁用 `linestretch` 并且修改过 `\CJKglue`，则只更新 `\ccwd`，否则设置伸展量为 0.08 倍 `\baselineskip`。注意 `everysel` 的钩子位于 `\size@update` 之前，`\baselineskip` 还未更新，不能直接使用它。

```

1952 \cs_new_protected_nopar:Npn \ctex_update_stretch:
1953 {
1954   \ctex_update_em_unit:
1955   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
1956   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
1957   {
1958     \ctex_if_ccglue_touched:TF
1959     { \ctex_update_ccwd: }
1960     {
1961       \dim_set:Nn \l__ctex_tmp_dim
1962       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
1963       \skip_set:Nn \l__ctex_ccglue_skip
1964       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
1965       \ctex_update_ccglue:
1966     }
1967   }
1968   {
1969     \int_set:Nn \l__ctex_tmp_int
1970     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
1971     \tex_divide:D \l__ctex_tmp_int \ccwd
1972     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
1973     {
1974       \skip_set:Nn \l__ctex_ccglue_skip
1975       {
1976         \c_zero_dim plus \dim_eval:n
1977         {
1978           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
1979           \l__ctex_tmp_int
1980         }
1981       }
1982     }
1983     { \skip_zero:N \l__ctex_ccglue_skip }
1984     \ctex_update_ccglue:
1985   }
1986 }

```

(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。

```

1987 \cs_new_protected_nopar:Npn \ctex_update_parindent:
1988 {
1989   \tl_if_empty:NF \l__ctex_autoindent_tl
1990   {
1991     \dim_compare:nNnF \parindent = \c_zero_dim
1992     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
1993   }
1994 }

```

(End definition for `\ctex_update_parindent:`)

`\ziju` 若参数为 0, 则恢复正常间距。

```

1995 \NewDocumentCommand \ziju { m }
1996 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
1997 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
1998 {
1999   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2000   \ctex_select_size:
2001 }

```

(End definition for `\ziju`. This function is documented on page 21.)

`\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。

```

2002 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2003 {
2004   \ctex_update_em_unit:
2005   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2006   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2007   \dim_compare:nNnTF \ccwd > \c_zero_dim

```

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

```

2008   {
2009     \dim_set:Nn \l__ctex_tmp_dim
2010     { \linewidth - \ccwd + \l__ctex_ziju_dim }
2011     \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2012     \tex_divide:D \l__ctex_tmp_int \ccwd
2013     \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。

```

2014     \dim_compare:nNnF \parindent = \c_zero_dim
2015     {
2016       \int_compare:nNnF \l__ctex_tmp_int < \c_three
2017       { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2018     }
2019     \skip_set:Nn \l__ctex_ccglue_skip
2020     {
2021       \l__ctex_ziju_dim
2022       plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2023       minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2024       { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2025     }
2026   }
2027   { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2028   \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 2ccwd 。

```
2029 \ctex_update_parindent:
2030 }
2031 \dim_new:N \l__ctex_ziju_dim
```

(End definition for `\ctex_update_ziju:`)

`\CTEXindent`
`\CTEXnoindent` 过时命令。

```
2032 \NewDocumentCommand \CTEXindent { }
2033 {
2034   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2035   { \parindent is~ set~ to~ 2\ccwd. }
2036   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2037 }
2038 \NewDocumentCommand \CTEXnoindent { }
2039 {
2040   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2041   { \parindent is~ set~ to~ Opt. }
2042   \dim_zero:N \parindent
2043 }
```

(End definition for `\CTEXindent` and `\CTEXnoindent`. These functions are documented on page 28.)

15.6 中文数字与日期

```
2044 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2045 \RequirePackage { zhnumber }
```

`\chinese`

```
2046 \cs_new_eq:NN \chinese \zhnum
2047 \cs_new_eq:NN \Chinese \chinese
2048 \cs_new_eq:NN \CTEXcounter \use_none:n
```

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

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

```
2049 \ctex_at_end_package:nn { enumitem }
2050 {
2051   \cs_if_free:NF \AddEnumerateCounter
2052   {
2053     \AddEnumerateCounter * { \zhnum } { \zhnum_int:n } { 1 }
2054     \AddEnumerateCounter * { \chinese } { \zhnum_int:n } { 1 }
2055   }
2056 }
```

`\CTEXnumber`
`\CTEXdigits`

```
2057 \NewDocumentCommand \CTEXnumber { m m }
2058 { \protected@edef #1 { \zhnumber {#2} } }
2059 \NewDocumentCommand \CTEXdigits { m m }
2060 { \protected@edef #1 { \zhdigits {#2} } }
```

(End definition for `\CTEXnumber` and `\CTEXdigits`. These functions are documented on page 23.)

`today`

```
2061 \cs_set_eq:NN \CTEX@todayold \today
2062 \keys_define:nn { ctex }
2063 {
2064   today .choice: ,
```

```

2065 today / old .code:n =
2066 { \cs_set_eq:NN \today \CTEX@todayold } ,
2067 today / small .code:n =
2068 {
2069   \cs_set_eq:NN \today \zhtoday
2070   \zhnumsetup { time = Arabic }
2071 } ,
2072 today / big .code:n =
2073 {
2074   \cs_set_eq:NN \today \zhtoday
2075   \zhnumsetup { time = Chinese }
2076 } ,
2077 today / unknown .code:n =
2078 { \msg_error:nxx { ctex } { today-undef } {#1} }
2079 }
2080 \msg_new:nnnn { ctex } { today-undef }
2081 { Today~format~`#1'~is~undefined. }
2082 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

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

15.7 其它中文标题定义

```

2083 \keys_define:nn { ctex }
2084 {
2085   contentsname .tl_set:N = \contentsname ,
2086   listfigurename .tl_set:N = \listfigurename ,
2087   listtablename .tl_set:N = \listtablename ,
2088   figurename .tl_set:N = \figurename ,
2089   tablename .tl_set:N = \tablename ,
2090   abstractname .tl_set:N = \abstractname ,
2091   indexname .tl_set:N = \indexname ,
2092   appendixname .tl_set:N = \appendixname ,
2093   <article> bibname .tl_set:N = \refname
2094   <book|report> bibname .tl_set:N = \bibname
2095 }
2096 <*ctex>
2097 \msg_new:nnn { ctex } { ctexbibname }
2098 {
2099   Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2100   The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2101 }
2102 \tl_if_exist:NTF \bibname
2103 { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2104 {
2105   \tl_if_exist:NTF \refname
2106   { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2107   {
2108     \msg_warning:nn { ctex } { ctexbibname }
2109     \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2110   }
2111 }
2112 </ctex>
2113 </class|ctex>

```

15.8 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。

```

2114 <*class|heading>

```

15.8.1 定义标题格式选项

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

```
2115 \seq_new:N \c__ctex_section_headings_seq
2116 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2117 { section , subsection , subsubsection , paragraph , subparagraph }
```

(End definition for `\c__ctex_section_headings_seq`)

`\c__ctex_headings_seq`

```
2118 \seq_new:N \c__ctex_headings_seq
2119 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
2120 <book|report>\seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2121 \seq_gput_left:Nn \c__ctex_headings_seq { part }
```

(End definition for `\c__ctex_headings_seq`)

`__ctex_initial_heading:n`

```
2122 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2123 {
2124   \tl_new:c { CTEX@pre#1 }
2125   \tl_new:c { CTEX@post#1 }
2126   \tl_const:cx { CTEXthe#1 }
2127   {
2128     \exp_not:c { CTEX@pre#1 }
2129     \exp_not:c { CTEX@the#1 }
2130     \exp_not:c { CTEX@post#1 }
2131   }
2132   \tl_const:cx { CTEX@#1name }
2133   {
2134     \group_begin:
2135     \exp_not:c { CTEX@#1@nameformat }
2136     {
2137       \exp_not:c { CTEX@pre#1 }
2138       \exp_not:N \tl_if_empty:NTF
2139       \exp_not:c { CTEX@#1@numberformat }
2140       { \exp_not:c { CTEX@the#1 } }
2141       {
2142         \group_begin:
2143         \exp_not:c { CTEX@#1@numberformat }
2144         \exp_not:c { CTEX@the#1 }
2145         \group_end:
2146       }
2147       \exp_not:c { CTEX@post#1 }
2148     }
2149   }
2150 }
2151 }
```

(End definition for `__ctex_initial_heading:n`)

`__ctex_def_heading_keys:n`

```
2152 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2153 {
2154   \tl_put_right:Nx \l__ctex_tmp_tl
2155   {
2156     #1 .meta:nn = { ctex / #1 } { ####1 } ,
2157     #1 / name .code:n =
2158     { \ctex_assign_heading_name:nn {#1} { ####1 } } ,
2159     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2160     #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2161     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
```

```

2162     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2163     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2164     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2165     #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2166     #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
2167     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2168     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2169     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2170     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2171     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2172     #1 / format+ .code:n =
2173     { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2174     #1 / nameformat+ .code:n =
2175     { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2176     #1 / numberformat+ .code:n =
2177     { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2178     #1 / aftername+ .code:n =
2179     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2180     #1 / titleformat+ .code:n =
2181     { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2182     #1 / aftertitle+ .code:n =
2183     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2184     #1 / numbering .initial:n = true ,
2185     #1 / beforeskip .initial:n = \c_zero_skip ,
2186     #1 / afterskip .initial:n = \c_zero_skip ,
2187     #1 / indent .initial:n = \c_zero_dim ,
2188     #1 / beforeskip .value_required:n = true ,
2189     #1 / afterskip .value_required:n = true ,
2190     #1 / indent .value_required:n = true ,
2191   }
2192 }

```

(End definition for `_ctex_def_heading_keys:n`.)

`ctex_assign_heading_name:nn`
`_ctex_assign_heading_name:nnn`

`name` 的值是一个至多两个元素的逗号分隔列表。由于 L^AT_EX₃ 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。

```

2193 \NewDocumentCommand \ctex_assign_heading_name:nn
2194 { m > { \SplitArgument { \c_one } { , } } +m }
2195 { \_ctex_assign_heading_name:nnn {#1} #2 }
2196 \cs_new_protected:Npn \_ctex_assign_heading_name:nnn #1#2#3
2197 {
2198   \tl_set:cn { CTEX@pre#1 } {#2}
2199   \IfNoValueTF {#3}
2200     { \tl_clear:c { CTEX@post#1 } }
2201     { \tl_set:cn { CTEX@post#1 } {#3} }
2202 }

```

(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`

```

2203 <*book|report>
2204 \tl_set:Nn \l__ctex_tmp_tl
2205 {
2206   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2207   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2208 }
2209 </book|report>

```

(End definition for `part/pagestyle` and `chapter/pagestyle`. These functions are documented on page 20.)

定义标题键值选项。

```

2210 \seq_map_inline:Nn \c__ctex_headings_seq
2211 {
2212   \__ctex_initial_heading:n {#1}
2213   \__ctex_def_heading_keys:n {#1}
2214 }
2215 \use:x { \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } } }
2216 \tl_clear:N \l__ctex_tmp_tl

```

15.8.2 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。

```

2217 \cs_new_protected_nopar:Npn \CTEX@makeheadinganchor { }
2218 \ctex_at_end_package:nn { hyperref }
2219 { \cs_gset_protected_nopar:Npn \CTEX@makeheadinganchor { \phantomsection } }

```

(End definition for `\CTEX@makeheadinganchor`.)

15.8.2.1 part 的标题

```
2220 <@=>
```

`\part`

```

2221 <*article>
2222 \renewcommand\part{%
2223   \if@noskipsec \leavevmode \fi
2224   \par
2225   % \addvspace{4ex}%
2226   \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
2227   \ifodd \CTEX@part@afterindent
2228     \@afterindenttrue
2229   \else
2230     \@afterindentfalse
2231   \fi
2232   \secdef\@part\@spart}
2233 </article>
2234 <*book|report>
2235 \renewcommand\part{%
2236   \if@openright
2237     \cleardoublepage
2238   \else
2239     \clearpage
2240   \fi
2241   % \thispagestyle{plain}%
2242   \thispagestyle{\CTEX@part@pagestyle}%
2243   \if@twocolumn
2244     \onecolumn
2245     \@tempwattrue
2246   \else
2247     \@tempwafalse
2248   \fi
2249   % \null\vfil
2250   \null\vskip \glueexpr \CTEX@part@beforeskip \relax
2251   \secdef\@part\@spart}
2252 </book|report>

```

(End definition for `\part`.)

`\@part`

```

2253 <*article>
2254 \def\@part[#1]#2{%
2255   \ifnum \c@secnumdepth >\m@ne
2256     \ifodd \CTEX@part@numbering

```

```

2257     \refstepcounter{part}%
2258 %     \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2259     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2260     \else
2261         \CTEX@makeheadinganchor
2262         \addcontentsline{toc}{part}{#1}%
2263     \fi
2264 \else
2265     \addcontentsline{toc}{part}{#1}%
2266 \fi
2267 {\interlinepenalty \@M
2268 % \normalfont \parindent \z@ \raggedright
2269 \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2270 \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2271 %     \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2272     \CTEX@partname \CTEX@part@aftername
2273 \fi \fi
2274 % \huge\bfseries #2%
2275 \CTEX@part@titleformat{#2}%
2276 \markboth{}{}%
2277 \CTEX@part@aftertitle}%
2278 \nobreak
2279 % \vskip 3ex
2280 \vskip \glueexpr \CTEX@part@afterskip \relax
2281 \afterheading}
2282 </article>
2283 <*book|report>
2284 \def\@part[#1]#2{%
2285     \ifnum \c@secnumdepth >-2\relax
2286         \ifodd \CTEX@part@numbering
2287             \refstepcounter{part}%
2288 %             \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2289             \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2290         \else
2291             \CTEX@makeheadinganchor
2292             \addcontentsline{toc}{part}{#1}%
2293         \fi
2294     \else
2295         \addcontentsline{toc}{part}{#1}%
2296     \fi
2297     \markboth{}{}%
2298     {\interlinepenalty \@M
2299 %     \normalfont \centering
2300 \normalfont \CTEX@part@format
2301 \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2302 %     \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2303     \CTEX@partname \CTEX@part@aftername
2304 \fi \fi
2305 % \Huge\bfseries #2\par}%
2306 \CTEX@part@titleformat{#2}%
2307 \CTEX@part@aftertitle}%
2308 \@endpart}
2309 </book|report>

```

(End definition for \@part.)

\@spart

```

2310 <*article>
2311 \def\@spart#1{%
2312     {\interlinepenalty \@M
2313 %     \normalfont \parindent \z@ \raggedright
2314 \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2315 \CTEX@part@format
2316 %     \huge \bfseries #1\par}%
2317 \CTEX@part@titleformat{#1}%
2318 \CTEX@part@aftertitle}%

```

```

2319     \nobreak
2320 %     \vskip 3ex
2321     \vskip \glueexpr \CTEX@part@afterskip \relax
2322     \@afterheading}
2323 </article>
2324 <*book|report>
2325 \def\@spart#1{%
2326     {\interlinepenalty \@M
2327 %     \normalfont \centering
2328     \normalfont \CTEX@part@format
2329 %     \Huge \bfseries #1\par}%
2330     \CTEX@part@titleformat{#1}%
2331     \CTEX@part@aftertitle}%
2332     \@endpart}
2333 </book|report>

```

(End definition for \@spart.)

\@endpart

```

2334 <*book|report>
2335 \def\@endpart{%
2336 %     \vfil
2337     \vskip \glueexpr \CTEX@part@afterskip \relax
2338     \newpage
2339     \if@twoside
2340     \if@openright
2341     \null
2342     \thispagestyle{empty}%
2343     \newpage
2344     \fi
2345     \fi
2346     \if@tempwa
2347     \twocolumn
2348     \fi}
2349 </book|report>

```

(End definition for \@endpart.)

15.8.2.2 chapter 的标题

```

2350 <*book|report>

```

\chapter

```

2351 \renewcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
2352 %     \thispagestyle{plain}%
2353     \thispagestyle{\CTEX@chapter@pagestyle}%
2354     \global\@topnum\z@
2355 %     \@afterindentfalse
2356     \ifodd \CTEX@chapter@afterindent
2357     \afterindenttrue
2358     \else
2359     \afterindentfalse
2360     \fi
2361     \secdef\@chapter\@schapter}

```

(End definition for \chapter.)

\@chapter

```

2362 \def\@chapter[#1]#2{%
2363     \ifnum \c@secnumdepth >\m@ne
2364 <book>     \if@mainmatter
2365     \ifodd \CTEX@chapter@numbering
2366     \refstepcounter{chapter}%

```

```

2367 % \typeout{\@chapapp\space\thechapter.}%
2368 \typeout{\CTEXthechapter}%
2369 \addcontentsline{toc}{chapter}
2370 % {\protect\numberline{\thechapter}#1}%
2371 {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2372 \else
2373 \CTEX@makeheadinganchor
2374 \addcontentsline{toc}{chapter}{#1}%
2375 \fi
2376 <book> \else
2377 <book> \addcontentsline{toc}{chapter}{#1}%
2378 <book> \fi
2379 \else
2380 \addcontentsline{toc}{chapter}{#1}%
2381 \fi
2382 \chaptermark{#1}%
2383 \addtocontents{lof}{\protect\addvspace{10\p}}%
2384 \addtocontents{lot}{\protect\addvspace{10\p}}%
2385 \if@twocolumn
2386 \topnewpage[\@makechapterhead{#2}]%
2387 \else
2388 \makechapterhead{#2}%
2389 \@afterheading
2390 \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2391 \def\@makechapterhead#1{%
2392 % \vspace*{50\p}%
2393 \vspace*{\glueexpr\CTEX@chapter@beforeskip\relax}%
2394 % {\normalfont \parindent \z@ \raggedright
2395 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2396 \CTEX@chapter@format
2397 \ifnum \c@secnumdepth >\m@ne
2398 <book> \if@mainmatter
2399 \ifodd \CTEX@chapter@numbering
2400 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2401 \CTEX@chaptername \CTEX@chapter@aftername
2402 \fi
2403 <book> \fi
2404 \fi
2405 \interlinepenalty\@M
2406 % \Huge \bfseries #1\par\nobreak
2407 \CTEX@chapter@titleformat{#1}%
2408 \CTEX@chapter@aftertitle
2409 \nobreak
2410 % \vskip 40\p@
2411 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2412 }}

```

(End definition for \@makechapterhead.)

\@makeschapterhead

```

2413 \def\@makeschapterhead#1{%
2414 % \vspace*{50\p}%
2415 \vspace*{\glueexpr\CTEX@chapter@beforeskip\relax}%
2416 % {\normalfont \parindent \z@ \raggedright
2417 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2418 \CTEX@chapter@format
2419 \interlinepenalty\@M
2420 % \Huge \bfseries #1\par\nobreak
2421 \CTEX@chapter@titleformat{#1}
2422 \CTEX@chapter@aftertitle
2423 \nobreak

```



```

2424 % \vskip 40\p@
2425 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2426 }

```

(End definition for \@makeschapterhead.)

```
2427 </book|report>
```

15.8.2.3 section 类的标题

\@startsection

```

2428 \def\@startsection#1#2#3#4#5#6{%
2429 \if@noskipsec \leavevmode \fi
2430 \par
2431 % \@tempskipa #4\relax
2432 % \@afterindenttrue
2433 % \ifdim \@tempskipa <\z@
2434 % \@tempskipa -\@tempskipa \@afterindentfalse
2435 % \fi
2436 \CTEX@update@sectionformat@n{#1}%
2437 \ifodd \CTEX@afterindent
2438 \@afterindenttrue
2439 \else
2440 \@afterindentfalse
2441 \fi
2442 \if@nobreak
2443 \everypar{}%
2444 \else
2445 % \addpenalty\@secpenalty\addvspace\@tempskipa
2446 \addpenalty\@secpenalty\addvspace{\glueexpr #4\relax}%
2447 \fi
2448 \@ifstar
2449 {\@ssect{#3}{#4}{#5}{#6}}%
2450 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}

```

(End definition for \@startsection.)

\@secntformat

```

2451 \def\@secntformat#1{%
2452 % \csname the#1\endcsname\quad}%
2453 \csname CTEX@#1name\endcsname \csname CTEX@#1@aftername\endcsname}

```

(End definition for \@secntformat.)

\@sect

```

2454 \def\@sect#1#2#3#4#5#6[#7]#8{%
2455 \ifnum #2>\c@secnumdepth
2456 \let\@svsec\@empty
2457 \else
2458 \ifodd \csname CTEX@#1@numbering\endcsname
2459 \refstepcounter{#1}%
2460 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2461 \else
2462 \CTEX@makeheadinganchor
2463 \let\@svsec\@empty
2464 \fi
2465 \fi
2466 % \@tempskipa #5\relax
2467 % \ifdim \@tempskipa>\z@
2468 \unless \ifodd \CTEX@runin
2469 \begingroup
2470 #6{%
2471 \@hangfrom{\hskip\glueexpr #3\relax\@svsec}%

```

```

2472 % \interlinepenalty \@M #8\@par}%
2473 \interlinepenalty \@M
2474 \csname CTEX@#1@titleformat\endcsname{#8}%
2475 \csname CTEX@#1@aftertitle\endcsname}%
2476 \endgroup
2477 \csname #1mark\endcsname{#7}%
2478 \addcontentsline{toc}{#1}{%
2479 \ifnum #2>\c@secnumdepth \else
2480 \ifodd \csname CTEX@#1@numbering\endcsname
2481 % \protect\numberline{\csname the#1\endcsname}%
2482 \protect\numberline{\csname CTEXthe#1\endcsname}%
2483 \fi
2484 \fi
2485 #7}%
2486 \else
2487 \def\@svsechd{%
2488 #6{\hskip\glueexpr #3\relax
2489 % \@svsec #8}%
2490 \@svsec
2491 \csname CTEX@#1@titleformat\endcsname{#8}%
2492 \csname CTEX@#1@aftertitle\endcsname}%
2493 \csname #1mark\endcsname{#7}%
2494 \addcontentsline{toc}{#1}{%
2495 \ifnum #2>\c@secnumdepth \else
2496 \ifodd \csname CTEX@#1@numbering\endcsname
2497 % \protect\numberline{\csname the#1\endcsname}%
2498 \protect\numberline{\csname CTEXthe#1\endcsname}%
2499 \fi
2500 \fi
2501 #7}}%
2502 \fi
2503 \@xsect{#5}}

```

(End definition for \@sect.)

\@ssect

```

2504 \def\@ssect#1#2#3#4#5{%
2505 % \@tempskipa #3\relax
2506 % \ifdim \@tempskipa>\z@
2507 \unless \ifodd \CTEX@runin
2508 \begingroup
2509 #4{%
2510 \@hangfrom{\hskip\glueexpr #1\relax}%
2511 % \interlinepenalty \@M #5\@par}%
2512 \interlinepenalty \@M
2513 \CTEX@titleformat@n{#5}%
2514 \CTEX@aftertitle}%
2515 \endgroup
2516 \else
2517 % \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2518 \def\@svsechd{#4{\hskip\glueexpr #1\relax
2519 \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2520 \fi
2521 \@xsect{#3}}

```

(End definition for \@ssect.)

\@xsect

```

2522 \def\@xsect#1{%
2523 % \@tempskipa #1\relax
2524 % \ifdim \@tempskipa>\z@
2525 \unless \ifodd \CTEX@runin
2526 \par \nobreak
2527 % \vskip \@tempskipa
2528 \vskip\glueexpr #1\relax

```

```

2529 \afterheading
2530 \else
2531 \nobreakfalse
2532 \global\@noskipsectrue
2533 \everypar{%
2534 \if@noskipsec
2535 \global\@noskipsecfalse
2536 {\setbox\z@\lastbox}%
2537 \clubpenalty\@M
2538 \begingroup \@svsechd \endgroup
2539 \unskip
2540 % \@tempskipa #1\relax
2541 % \hskip -\@tempskipa
2542 \hskip\glueexpr #1\relax
2543 \else
2544 \clubpenalty \@clubpenalty
2545 \everypar{%
2546 \fi}%
2547 \fi
2548 \ignorespaces}

```

(End definition for \@xsect.)

```
2549 <@@=ctex>
```

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

```

2550 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2551 {
2552 \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2553 \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2554 \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2555 \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2556 }
2557 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2558 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2559 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2560 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

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

```

2561 \int_zero:N \l__ctex_tmp_int
2562 \seq_map_inline:Nn \c__ctex_section_headings_seq
2563 {
2564 \int_incr:N \l__ctex_tmp_int
2565 \cs_gset_protected_nopar:cpx {#1}
2566 {
2567 \exp_not:N \@startsection {#1}
2568 { \int_use:N \l__ctex_tmp_int }
2569 { \exp_not:c { CTEX@#1@indent } }
2570 { \exp_not:c { CTEX@#1@beforeskip } }
2571 { \exp_not:c { CTEX@#1@afterskip } }
2572 { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2573 }
2574 }

```

15.8.2.4 附录标题

appendix/name

appendix/number

appendix/numbering

```

2575 \keys_define:nn { ctex }
2576 {
2577 appendix .meta:nn = { ctex / appendix } {#1} ,
2578 appendix / name .code:n =
2579 { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2580 appendix / number .tl_set:N = \CTEX@appendix@number ,

```

```

2581     appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2582     appendix / numbering .initial:n = true
2583   }
2584   \tl_new:N \CTEX@preappendix
2585   \tl_new:N \CTEX@postappendix

```

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

`\appendix`

```

2586 \cs_new_eq:NN \CTEX@save@appendix \appendix
2587 \cs_gset_protected_nopar:Npn \appendix
2588   {
2589     \CTEX@save@appendix
2590     (*article)
2591     \gdef \CTEX@presection { \CTEX@preappendix }
2592     \gdef \CTEX@thesection { \CTEX@appendix@number }
2593     \gdef \CTEX@postsection { \CTEX@postappendix }
2594     \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2595     (/article)
2596     (*book|report)
2597     \gdef \CTEX@prechapter { \CTEX@preappendix }
2598     \gdef \CTEX@thechapter { \CTEX@appendix@number }
2599     \gdef \CTEX@postchapter { \CTEX@postappendix }
2600     \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2601     (/book|report)
2602   }

```

(End definition for `\appendix`.)

15.8.2.5 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 `\ttl@<section>` 的宏中备用, 它的内容是

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

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

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

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

```

2603 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
2604   {
2605     \@ifpackagewith { titlesec } { explicit }
2606     {
2607       \cs_set_eq:NN \__ctex_titlesec_format:Nn
2608         \__ctex_titlesec_format_explicit:Nn
2609     }
2610     { }
2611     \clist_map_inline:mn
2612     { indentafter , noindentafter , indentfirst , nonindentfirst }

```

```

2613     {
2614     \@ifpackagewith { titlesec } { ##1 }
2615     {
2616     \clist_map_break:n
2617     { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
2618     }
2619     { }
2620     }
2621     \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
2622     }
2623     \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
2624     {
2625     \__ctex_titlesec_format:n {#1}
2626     \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
2627     }
2628     \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
2629     {
2630     \cs_if_free:cF { ttlf@#1 }
2631     { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
2632     }
2633     \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
2634     {
2635     \tl_set:Nx #1
2636     {
2637     \bool_if:cTF { CTEX@#2@runin }
2638     { \exp_not:N \ttlh@runin }
2639     { \exp_not:N \ttlh@hang }
2640     \tl_tail:N #1
2641     }
2642     }
2643     \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
2644     {
2645     \cs_set_nopar:Npx #1 ##1
2646     {
2647     \bool_if:cTF { CTEX@#2@runin }
2648     { \exp_not:N \ttlh@runin }
2649     { \exp_not:N \ttlh@hang }
2650     \exp_args:No \tl_tail:n { #1 { } }
2651     }
2652     }
2653     \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
2654     { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnn #1 {#2} } }
2655     \cs_new:Npn \__ctex_titlesec_spacing:nnnnn #1#2#3#4#5#6
2656     {
2657     \exp_not:n { {#1} {#2} {#3} {#4} }
2658     { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
2659     }

```

(End definition for `\ctex_titlesec_hook:.`)

```

2660 \@ifpackageloaded { titlesec }
2661 { }
2662 {
2663 \ctex_at_end_package:nn { titlesec }
2664 {
2665 \@ifpackagewith { titlesec } { loadonly }
2666 { }
2667 { \ctex_titlesec_hook: }
2668 }
2669 }

```

15.8.2.6 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。

```

2670 \group_begin:
2671 \char_set_catcode_other:N \#
2672 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
2673 {
2674   \ctex_patch_cmd:Nnn \ttl@settopmark
2675   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
2676   {
2677     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
2678     \CTEX@titlepslabel@set {#1}
2679   }
2680   \ctex_patch_cmd:Nnn \ttl@setsubmark
2681   { \protect \@namedef { the#1 } { } }
2682   {
2683     \protect \@namedef { the#1 } { }
2684     \CTEX@titlepslabel@clear {#1}
2685   }
2686   \ctex_patch_cmd:Nnn \ttl@setsubmark
2687   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
2688   {
2689     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
2690     \CTEX@titlepslabel@set {#2}
2691   }
2692 }
2693 \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。

```

2694 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
2695 {
2696   \cs_if_free:cF { CTEXthe#1 }
2697   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
2698 }
2699 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
2700 {
2701   \cs_if_free:cF { CTEXthe#1 }
2702   { \protect \@namedef { CTEXthe#1 } { } }
2703 }

```

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

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

```

2704 \ctex_at_end_package:nn { titlesec }
2705 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
2706 \ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }

```

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

```

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

```

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

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

15.8.3 目录标签的宽度

`\numberline`

```

2707 \cs_new_protected:Npn \CTEX@toc@width@n #1
2708 {
2709   \hbox_set:Nn \l__ctex_tmp_box {#1}
2710   \dim_set:Nn \@tempdima
2711     {
2712       \dim_max:nn { \@tempdima }
2713       { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2714     }
2715 }
2716 \group_begin:
2717 \char_set_catcode_other:N \#
2718 \use:n
2719 {
2720   \group_end:
2721   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
2722     { \CTEX@toc@width@n {#1} }
2723     { }
2724     { \ctex_patch_failure:N \numberline }
2725   \ifpackageloaded { tocloft }
2726     { }
2727     {
2728       \ctex_at_end_package:nn { tocloft }
2729         {
2730           \ctex_preto_cmd:NnnTF \numberline
2731             { \char_set_catcode_letter:n { 64 } }
2732             { \CTEX@toc@width@n {#1} }
2733             { }
2734             { \ctex_patch_failure:N \numberline }
2735         }
2736     }
2737 }

```

(End definition for \numberline.)

15.8.4 页眉信息的修改

`\ps@headings`

```

2738 <*article>
2739 \ctex_patch_cmd:Nnn \ps@headings
2740 { \thesection \quad }
2741 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2742 \if@twoside
2743   \ctex_patch_cmd:Nnn \ps@headings
2744     { \thesubsection \quad }
2745     { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2746 \fi:
2747 </article>
2748 <*book|report>
2749 \ctex_patch_cmd:Nnn \ps@headings
2750 { \@chapapp\ \thechapter.\ }
2751 { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2752 \if@twoside
2753   \ctex_patch_cmd:Nnn \ps@headings
2754     { \thesection.\ }
2755     { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2756 \fi:
2757 </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。

```

2758 \ctex_at_end_package:nn { fancyhdr }
2759 {
2760   (*article)
2761   \ctex_patch_cmd:Nnn \ps@fancy
2762     { \thesection \hskip 1em \relax }
2763     { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2764   \ctex_patch_cmd:Nnn \ps@fancy
2765     { \thesubsection \hskip 1em \relax }
2766     { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2767   (/article)
2768   (*book|report)
2769   \ctex_patch_cmd:Nnn \ps@fancy
2770     { \@chapapp\ \thechapter.\ }
2771   (book)   { \if@mainmatter \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi \fi }
2772   (report) { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2773   \ctex_patch_cmd:Nnn \ps@fancy
2774     { \thesection.\ }
2775     { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2776   (/book|report)
2777 }

```

(End definition for `\ps@fancy`.)

15.8.5 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。

```

2778 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
2779 {
2780   \protected@edef \@currentlabel
2781     {
2782       \cs_if_exist:cTF { CTEX@the#1 }
2783         { \exp_args:cc { p@#1 } { CTEX@the#1 } }
2784         { \exp_not:o { \@currentlabel } }
2785     }
2786 }

```

(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 `CTEX` 之后被载入,我们之前的修改将会被覆盖。

```

2787 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
2788 {
2789   \seq_map_inline:Nn \c__ctex_headings_seq
2790     { \ctex_fix_varioref_label:n { ##1 } }
2791 }

```

(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>`。`LATEX` 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。

```

2792 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1

```



```

2793 {
2794   \tl_if_empty:cT { p@#1 }
2795   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
2796 }

```

(End definition for `_ctex_fix_varioref_label:n`)

如果 `varioref` 已经被载入,则使用它来设置。

```

2797 \@ifpackageloaded { varioref }
2798 { \ctex_varioref_hook: }
2799 {
2800   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
2801   \RenewDocumentCommand \refstepcounter { m }
2802   {
2803     \CTEX@save@refstepcounter {#1}
2804     \CTEX@setcurrentlabel@n {#1}
2805   }
2806   \ctex_at_end_package:nm { varioref } { \ctex_varioref_hook: }
2807 }

```

15.8.6 载入 `(scheme)` 文件

```

2808 \ctex_scheme_input:o { \l__ctex_scheme_tl }
2809 </class|heading>

```

15.8.7 标题格式的 `scheme` 定义

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

```

2810 <*scheme&(article|book|report)>
2811 \keys_set:nm { ctex / part }
2812 {
2813   aftertitle = \par ,
2814   <*plain>
2815     name      = \partname \space ,
2816     number    = \thepart ,
2817   </plain>
2818   <*chinese>
2819     number    = \chinese { part } ,
2820   </chinese>
2821   <*article>
2822     beforekip = 4ex ,
2823     afterskip = 3ex ,
2824   <*plain>
2825     format    = \raggedright ,
2826     nameformat = \Large \bfseries ,
2827     aftername = \par \nobreak ,
2828     titleformat = \huge \bfseries ,
2829     afterindent = false
2830   </plain>
2831   <*chinese>
2832     format    = \Large \bfseries \centering ,
2833     aftername = \quad ,
2834     afterindent = true
2835   </chinese>
2836   </article>
2837   <*book|report>
2838     aftername = \par \vskip 20 \p@ ,
2839     beforekip = 0pt \@plus 1fil ,
2840     afterskip = 0pt \@plus 1fil ,
2841     pagestyle = plain ,
2842   <*plain>
2843     format    = \centering ,

```

```

2844     nameformat = \huge \bfseries ,
2845     titleformat = \Huge \bfseries
2846   \plain
2847   (*chinese)
2848     format      = \huge \bfseries \centering
2849   \chinese
2850   \book|report
2851   }

2852   (*book|report)
2853   \keys_set:nn { ctex / chapter }
2854   {
2855     pagestyle = plain ,
2856     aftertitle = \par ,
2857     beforekip = 50 \p@ ,
2858     afterskip = 40 \p@ ,
2859   \plain
2860     name      = \chaptername \space ,
2861     number    = \thechapter ,
2862     format    = \raggedright ,
2863     nameformat = \huge \bfseries ,
2864     aftername = \par \nobreak \vskip 20 \p@ ,
2865     titleformat = \Huge \bfseries ,
2866     afterindent = false
2867   \plain
2868   (*chinese)
2869     number    = \chinese { chapter } ,
2870     format    = \huge \bfseries \centering ,
2871     aftername = \quad ,
2872     afterindent = true
2873   \chinese
2874   }
2875   \book|report

2876   (@@=)

2877   \keys_set:nn { ctex / section }
2878   {
2879     number      = \thesection ,
2880     aftername    = \quad ,
2881     aftertitle   = \@@par ,
2882     beforekip    = 3.5ex \@plus 1ex \@minus .2ex ,
2883     afterskip    = 2.3ex \@plus .2ex ,
2884     runin        = false ,
2885   \plain
2886     format      = \Large \bfseries ,
2887     afterindent = false
2888   \plain
2889   (*chinese)
2890     format      = \Large \bfseries \centering ,
2891     afterindent = true
2892   \chinese
2893   }

2894   \keys_set:nn { ctex / subsection }
2895   {
2896     number      = \thesubsection ,
2897     format      = \large \bfseries ,
2898     aftername    = \quad ,
2899     aftertitle   = \@@par ,
2900     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
2901     afterskip    = 1.5ex \@plus .2ex ,
2902     runin        = false ,
2903   \plain
2904     afterindent = false
2905   \plain
2906   (*chinese)
2907     afterindent = true

```

```

2908 </chinese>
2909 }

2910 \keys_set:nn { ctex / subsubsection }
2911 {
2912     number      = \thesubsubsection ,
2913     format      = \normalsize \bfseries ,
2914     aftername   = \quad ,
2915     aftertitle  = \@@par ,
2916     beforekip   = 3.25ex \@plus 1ex \@minus .2ex ,
2917     afterskip   = 1.5ex \@plus .2ex ,
2918     runin       = false ,
2919     <!--*plain-->
2920     afterindent = false
2921 </plain-->
2922     <!--*chinese-->
2923     afterindent = true
2924 </chinese-->
2925 }

2926 \keys_set:nn { ctex / paragraph }
2927 {
2928     number      = \theparagraph ,
2929     format      = \normalsize \bfseries ,
2930     aftername   = \quad ,
2931     beforekip   = 3.25ex \@plus 1ex \@minus .2ex ,
2932     <!--*plain-->
2933     afterindent = false
2934 </plain-->
2935     <!--*chinese-->
2936     afterindent = true
2937 </chinese-->
2938 }

2939 \keys_set:nn { ctex / subparagraph }
2940 {
2941     number      = \thesubparagraph ,
2942     format      = \normalsize \bfseries ,
2943     aftername   = \quad ,
2944     beforekip   = 3.25ex \@plus 1ex \@minus .2ex ,
2945     <!--*plain-->
2946     afterindent = false
2947 </plain-->
2948     <!--*chinese-->
2949     afterindent = true
2950 </chinese-->
2951 }

```

处理 sub3section 与 sub4section 的格式。

```

2952 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2953 {
2954     \keys_set:nn { ctex / paragraph }
2955     {
2956         aftertitle = \@@par ,
2957         afterskip  = 1ex \@plus .2ex ,
2958         runin      = false
2959     }
2960 }
2961 {
2962     \keys_set:nn { ctex / paragraph }
2963     {
2964         afterskip  = 1em ,
2965         runin      = true
2966     }
2967 }
2968 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
2969 {

```

```

2970 \keys_set:nn { ctex / subparagraph }
2971 {
2972   aftertitle = \@@par ,
2973   afterskip  = 1ex \@plus .2ex ,
2974   runin      = false
2975 }
2976 }
2977 {
2978 \keys_set:nn { ctex / subparagraph }
2979 {
2980   afterskip = 1em ,
2981   runin     = true
2982 }
2983 }
2984 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2985 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
2986 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
2987 <@=@=ctex>

```

处理附录的格式。

```

2988 \keys_set:nn { ctex / appendix }
2989 <*article>
2990 { number = \@Alph \c@section }
2991 </article>
2992 <*book|report>
2993 {
2994   name = \appendixname \space ,
2995   number = \@Alph \c@chapter
2996 }
2997 </book|report>
2998 </scheme&(article|book|report)>

```

15.8.8 ctex.sty 的 heading 选项

```
2999 <*ctex|ctexheading>
```

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

```

3000 \clist_map_inline:nn { article , book , report }
3001 {
3002   \ifclassloaded {#1}
3003   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3004   { }
3005 }

```

(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`。

```

3006 \msg_new:nnn { ctex } { not-standard-class }
3007 {
3008   None~of~the~standard~document~classes~was~loaded.\\
3009   Heading~`#1'~is~selected.\\
3010   ctex~may~not~work~as~expected.
3011 }
3012 <ctex>\bool_if:NTF \l__ctex_heading_bool
3013 <ctexheading>\use:n
3014 {
3015   \tl_if_exist:NTF \c__ctex_std_class_tl
3016   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3017   {
3018     \cs_if_exist:NTF \chapter
3019     {

```

```

3020         \cs_if_exist:NF \if@mainmatter
3021         { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3022         \tl_const:Nn \c__ctex_class_tl { book }
3023     }
3024     { \tl_const:Nn \c__ctex_class_tl { article } }
3025     \msg_warning:nxx { ctex } { not-standard-class } { \c__ctex_class_tl }
3026 }
3027 \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3028 }
3029 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3030 </ctex|ctexheading>

```

15.8.9 标题配置文件

```

3031 <^UTF8|GBK>
3032 \keys_set:nn { ctex }
3033 {
3034     contentsname = 目录 ,
3035     listfigurename = 插图 ,
3036     listtablename = 表格 ,
3037     figurename = 图 ,
3038     tablename = 表 ,
3039     abstractname = 摘要 ,
3040     indexname = 索引 ,
3041     bibname = 参考文献 ,
3042     appendixname = 附录
3043 }
3044 \keys_if_exist:nnT { ctex / part } { name }
3045 {
3046     \keys_set:nn { ctex / part } { name = { 第 , 部分 } }
3047     \keys_if_exist:nnT { ctex / chapter } { name }
3048     { \keys_set:nn { ctex / chapter } { name = { 第 , 章 } } }
3049 }
3050 </UTF8|GBK>

```

15.9 chinese 方案的其他设置

```
3051 <^scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
3052 <article|book|report> \pagestyle { headings }
```

汉化标题名字。

```

3053 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3054 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3055 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

日期格式。

```
3056 \keys_set:nn { ctex } { today = small }
```

若用户未设置宏包选项 zihao, 则设置 \normalsize 为五号字。

```

3057 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3058 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }

```

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。

```

3059 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3060 { \keys_set:nn { ctex / option } { linespread = 1.3 } }

```

若用户未设置宏包选项 autoindent, 则自动调整首行缩进。

```

3061 \ctex_if_autoindent_touched:F
3062 { \keys_set:nn { ctex } { autoindent = true } }

```

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。

```

3063 (*generic)
3064 \tl_if_exist:NT \c__ctex_std_class_tl
3065   { \RequirePackage { indentfirst } }
3066 (/generic)
3067 (/scheme&chinese)

```

15.10 中文字号

```

3068 (*class|ctex|ctexsize)

```

`\zihao`

```

3069 \NewDocumentCommand \zihao { m }
3070   { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

(End definition for `\zihao`. This function is documented on page 21.)

`\ctex_zihao:n`

```

3071 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3072   {
3073     \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3074     { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3075     { \msg_error:nnn { ctex } { fontsize } {#1} }
3076   }
3077 \msg_new:nnnn { ctex } { fontsize }
3078   { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao. }
3079   {
3080     The~old~font~size~is~used~if~you~continue.\\
3081     The~available~font~sizes~are~listed~as~follow.\\
3082     \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3083   }

```

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

15.10.1 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。

```

3084 \prop_new:N \c__ctex_font_size_prop
3085 \seq_new:N \c__ctex_font_size_seq
3086 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3087   {
3088     \use:x
3089     {
3090       \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3091       {
3092         { \dim_to_decimal:n {#2} }
3093         { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3094       }
3095     }
3096     \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3097   }
3098 \clist_map_inline:nn
3099   {
3100     { 8 } { 5 bp } ,
3101     { 7 } { 5.5 bp } ,
3102     { -6 } { 6.5 bp } ,
3103     { 6 } { 7.5 bp } ,

```

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

```

3104     { -5 } { 9 bp } ,
3105     { 5 } { 10.5 bp } ,
3106     { -4 } { 12 bp } ,
3107     { 4 } { 14 bp } ,
3108     { -3 } { 15 bp } ,
3109     { 3 } { 16 bp } ,
3110     { -2 } { 18 bp } ,
3111     { 2 } { 22 bp } ,
3112     { -1 } { 24 bp } ,
3113     { 1 } { 26 bp } ,
3114     { -0 } { 36 bp } ,
3115     { 0 } { 42 bp }
3116   }
3117   { \c__ctex_save_font_size:nn #1 }

```

(End definition for `\c__ctex_font_size_prop`.)

`\ctex_declare_math_sizes:nmm`

```

3118 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nmmn #1#2#3#4
3119   {
3120     \c__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3121     \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3122   }

```

(End definition for `\ctex_declare_math_sizes:nmmn`.)

`__ctex_get_font_sizes:Nn`

```

3123 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3124   {
3125     \tl_clear:N #1
3126     \tl_map_inline:nn {#2}
3127     {
3128       \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3129       { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3130       { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3131     }
3132   }

```

(End definition for `__ctex_get_font_sizes:Nn`.)

```

3133 \clist_map_inline:nn
3134   {
3135     { 8 } { 8 } { 5pt } { 5pt } ,
3136     { 7 } { 7 } { 5pt } { 5pt } ,
3137     { -6 } { -6 } { 5pt } { 5pt } ,
3138     { 6 } { 6 } { 5pt } { 5pt } ,
3139     { -5 } { -5 } { 6pt } { 5pt } ,
3140     { 5 } { 5 } { 7pt } { 5pt } ,
3141     { -4 } { -4 } { 8pt } { 6pt } ,
3142     { 4 } { 4 } { 5 } { 6 } ,
3143     { -3 } { -3 } { -4 } { -5 } ,
3144     { 3 } { 3 } { 4 } { 5 } ,
3145     { -2 } { -2 } { -3 } { -4 } ,
3146     { 2 } { 2 } { 3 } { 4 } ,
3147     { -1 } { -1 } { -2 } { -3 } ,
3148     { 1 } { 1 } { 2 } { 3 } ,
3149     { -0 } { -0 } { -1 } { -2 } ,
3150     { 0 } { 0 } { 1 } { 2 }
3151   }
3152   { \ctex_declare_math_sizes:nmmn #1 }

```

15.10.2 修改默认字号大小

`\ctex_set_font_size:Nnn`

```

3153 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3154 {
3155     \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3156     { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3157     { \msg_error:nnn { ctex } { fontsize } {#2} }
3158 }
3159 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3160 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

(End definition for \ctex_set_font_size:Nnn.)

```

3161 \if_case:w \g__ctex_font_size_flag
3162 \ctex_file_input:n { ctex-c5size.clo }
3163 \or:
3164 \ctex_file_input:n { ctex-cs4size.clo }
3165 \fi:
3166 </class|ctex|ctexsize>
3167 <*c5size>
3168 \ctex_set_font_size:Nnn \normalsize { 5 }
3169 {
3170     \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3171     \abovedisplayshortskip \z@ \@plus3\p@
3172     \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3173     \belowdisplayskip \abovedisplayskip
3174     \let\@listi\@listI
3175 }
3176 \ctex_set_font_size:Nnn \small { -5 }
3177 {
3178     \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3179     \abovedisplayshortskip \z@ \@plus2\p@
3180     \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3181     \def\@listi{\leftmargin\leftmarginI
3182         \topsep 4\p@ \@plus2\p@ \@minus2\p@
3183         \parsep 2\p@ \@plus\p@ \@minus\p@
3184         \itemsep \parsep}
3185     \belowdisplayskip \abovedisplayskip
3186 }
3187 \ctex_set_font_size:Nnn \footnotesize { 6 }
3188 {
3189     \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3190     \abovedisplayshortskip \z@ \@plus\p@
3191     \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3192     \def\@listi{\leftmargin\leftmarginI
3193         \topsep 3\p@ \@plus\p@ \@minus\p@
3194         \parsep 2\p@ \@plus\p@ \@minus\p@
3195         \itemsep \parsep}
3196     \belowdisplayskip \abovedisplayskip
3197 }
3198 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3199 \ctex_set_font_size:Nnn \tiny { 7 } { }
3200 \ctex_set_font_size:Nnn \large { -4 } { }
3201 \ctex_set_font_size:Nnn \Large { -3 } { }
3202 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3203 \ctex_set_font_size:Nnn \huge { 2 } { }
3204 \ctex_set_font_size:Nnn \Huge { 1 } { }
3205 </c5size>
3206 <*cs4size>
3207 \ctex_set_font_size:Nnn \normalsize { -4 }
3208 {
3209     \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3210     \abovedisplayshortskip \z@ \@plus3\p@
3211     \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3212     \belowdisplayskip \abovedisplayskip
3213     \let\@listi\@listI
3214 }
3215 \ctex_set_font_size:Nnn \small { 5 }

```



```

3216 {
3217   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3218   \abovedisplaysshortskip \z@ \@plus3\p@
3219   \belowdisplaysshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3220   \def\@listi{\leftmargin\leftmargini
3221             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3222             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3223             \itemsep \parsep}
3224   \belowdisplayskip \abovedisplayskip
3225 }
3226 \ctex_set_font_size:Nnn \footnotesize { -5 }
3227 {
3228   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3229   \abovedisplaysshortskip \z@ \@plus3\p@
3230   \belowdisplaysshortskip 6\p@ \@plus3\p@ \@minus3\p@
3231   \def\@listi{\leftmargin\leftmargini
3232             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3233             \parsep 3\p@ \@plus2\p@ \@minus\p@
3234             \itemsep \parsep}
3235   \belowdisplayskip \abovedisplayskip
3236 }
3237 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3238 \ctex_set_font_size:Nnn \tiny { -6 } { }
3239 \ctex_set_font_size:Nnn \large { -3 } { }
3240 \ctex_set_font_size:Nnn \Large { -2 } { }
3241 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3242 \ctex_set_font_size:Nnn \huge { -1 } { }
3243 \ctex_set_font_size:Nnn \Huge { 1 } { }
3244 </cs4size>
3245 <ctexsize>\normalsize
3246 <*class|ctex>

```

15.11 更新行距

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

```

3247 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3248 {
3249   \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。

```

3250   \group_begin: \footnotesize \exp_args:NNNo \group_end:
3251   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3252 }

```

(End definition for `\footnotesep`.)

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

```

3253 \normalsize

```

15.12 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。

```

3254 \NewDocumentCommand \CTeX { }
3255 { \ifincsname CTeX \else: C \TeX \fi: }
3256 \ctex_at_end_package:nn { hyperref }
3257 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

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

`captiondelimiter` 过时选项。

```

3258 \keys_define:nn { ctex }
3259 {
3260   captiondelimiter .code:n =
3261   {
3262     \msg_warning:nnn { ctex } { deprecated-option }
3263     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3264   }
3265 }

```

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

```
3266 </class|ctex>
```

15.12.1 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`

```

3267 <!--scheme!generic&chinese)
3268 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3269 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3270 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3271 </scheme!generic&chinese)

```

(End definition for `\verse` and `\quotation`.)

```
3272 <!--class|ctex>
```

15.13 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。

```

3273 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3274 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3275 \msg_new:nnn { ctex } { fontset-unavailable }
3276 { 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。

```

3277 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3278 {
3279   \tl_if_empty:NTF \g__ctex_fontset_tl
3280   {
3281     \ctex_detected_platform:
3282     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3283   }
3284   {
3285     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3286     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3287     {
3288       \use:x
3289       {
3290         \ctex_detected_platform:
3291         \msg_error:nxxx { ctex } { fontset-not-found }

```

```

3292             { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3293         }
3294     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3295 }
3296 }
3297 }
3298 \@onlypreamble \ctex_load_fontset:
3299 \msg_new:nnnn { ctex } { fontset-not-found }
3300 {
3301     CTex~fontset~`#1'~could~not~be~found.\\
3302     Fontset~`#2'~will~be~used~instead.
3303 }
3304 { 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。

```

3305 \keys_define:nn { ctex }
3306 {
3307     fontset .code:n =
3308     {
3309         \ctex_if_preamble:TF
3310         {
3311             \str_if_eq_x:nnTF {#1} { none }
3312             { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3313             {
3314                 \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3315                 {
3316                     \tl_gset:Nx \g__ctex_fontset_tl {#1}
3317                     \ctex_load_fontset:
3318                 }
3319                 {
3320                     \msg_error:nxxx { ctex } { fontset-loaded }
3321                     { \g__ctex_fontset_tl } {#1}
3322                 }
3323             }
3324         }
3325         { \msg_error:nn { ctex } { fontset-only-preamble } }
3326     }
3327 }
3328 \msg_new:nnnn { ctex } { fontset-loaded }
3329 {
3330     CTex~fontset~`#1'~has~been~loaded.
3331     \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
3332 }
3333 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3334 \msg_new:nnn { ctex } { fontset-only-preamble }
3335 {
3336     The~ `fontset'~ option~ can~ be~ used~ only~ in~ preamble.
3337 }

```

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

载入中文字库。

```

3338 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3339 { \ctex_load_fontset: }

```

15.14 宏包配置文件

15.14.1 ctex.cfg

```

3340 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3341 </class>ctex

```

```

3342 <*config>
3343 %%
3344 </config>

```

15.14.2 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。

```

3345 <*ctexopts>
3346 %%
3347 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3348 </ctexopts>

```

15.15 字体定义文件

15.15.1 传统定义方式

```

3349 <*c19|c70>
3350 %%
3351 %% Chinese characters
3352 %%
3353 <c19>%% character set: GBK (extension of GB 2312)
3354 <c70>%% character set: Unicode
3355 %% font encoding: Unicode
3356 %%
3357 </c19|c70>

3358 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3359 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3360 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3361 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3362 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3363 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

3364 <*rm>
3365 <*c19>
3366 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3367 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3368 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3369 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3370 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3371 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3372 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3373 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3374 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3375 </c19>
3376 <*c70>
3377 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3378 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3379 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3380 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3381 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3382 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3383 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3384 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3385 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3386 </c70>
3387 </rm>

3388 <*sf>
3389 <*c19>
3390 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3391 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3392 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3393 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3394 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3395 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}

```

```

3396 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3397 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3398 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3399 </c19>
3400 < *c70>
3401 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
3402 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
3403 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
3404 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
3405 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3406 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3407 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
3408 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
3409 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
3410 </c70>
3411 </sf>
3412 < *tt>
3413 < *c19>
3414 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
3415 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
3416 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
3417 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
3418 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3419 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3420 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
3421 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
3422 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
3423 </c19>
3424 < *c70>
3425 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
3426 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
3427 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
3428 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
3429 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
3430 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
3431 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
3432 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
3433 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
3434 </c70>
3435 </tt>
3436 < *fontset>

```

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

ctex 包利用 C:\boot.ini 文件的存在性来判断是否使用 Windows XP 及以前的版本, 分别载入新旧字体设置。

```

3437 < *windows>
3438 \file_if_exist:nTF { C:/boot.ini }
3439 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3440 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3441 </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。

```

3442 < *windowsnew|windowsold>
3443 \sys_if_engine_pdfTeX:TF
3444 {
3445   \ctex_zhmap_case:nnn
3446   {
3447     \ctex_punct_set:n { windows }
3448     \setCJKmainfont
3449     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3450 < *windowsold>
3451     \setCJKsansfont { simhei.ttf }

```

```

3452         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3453 </windowsold>

```

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。

```

3454 <(*windowsnew)
3455     \file_if_exist:nTF { C:/Windows/Fonts/msyh.ttc }
3456     {
3457         \setCJKsansfont [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3458         \setCJKfamilyfont { zhyahei }
3459         [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3460     }
3461     {
3462         \setCJKsansfont [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3463         \setCJKfamilyfont { zhyahei }
3464         [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3465     }
3466     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3467     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3468 </windowsnew)
3469     \setCJKmonofont { simfang.ttf }
3470     \setCJKfamilyfont { zhkai } { simkai.ttf }
3471     \setCJKfamilyfont { zhfs } { simfang.ttf }
3472     \setCJKfamilyfont { zhsong } { simsun.ttc }
3473     \setCJKfamilyfont { zhhei } { simhei.ttf }
3474     \setCJKfamilyfont { zhli } { simli.ttf }
3475     \setCJKfamilyfont { zhyou } { simyou.ttf }
3476     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3477     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3478     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3479     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3480 }
3481 {
3482     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3483     \ctex_punct_set:n { windows }
3484     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3485     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3486     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3487 }
3488 {
3489     \tl_set:Nn \CJKrmdefault { rm }
3490     \tl_set:Nn \CJKsfdefault { sf }
3491     \tl_set:Nn \CJKttdefault { tt }
3492 }
3493 }
3494 {
3495 <(*windowsold)
3496     \setCJKmainfont
3497     [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3498     \setCJKsansfont { SimHei }
3499     \setCJKmonofont { FangSong_GB2312 }
3500     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3501     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3502 </windowsold)
3503 <(*windowsnew)
3504     \setCJKmainfont
3505     [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3506     \setCJKsansfont
3507     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3508     \setCJKmonofont { FangSong }
3509     \setCJKfamilyfont { zhkai } { KaiTi }
3510     \setCJKfamilyfont { zhfs } { FangSong }
3511 </windowsnew)
3512     \setCJKfamilyfont { zhsong } { SimSun }
3513     \setCJKfamilyfont { zhhei } { SimHei }
3514     \setCJKfamilyfont { zhli } { LiSu }
3515     \setCJKfamilyfont { zhyou } { YouYuan }
3516 <(*windowsnew)

```

```

3517     \setCJKfamilyfont { zhyahei }
3518     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3519 </windowsnew>
3520 }
3521 </windowsnew|windowsold>

```

15.15.3 ctex-fontset-adobe.def

```
3522 (*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。

```

3523 \sys_if_engine_pdftex:TF
3524 {
3525     \sys_if_output_pdf:TF
3526     { \ctex_fontset_error:n { adobe } }
3527     {
3528         \ctex_zhmap_case:nnn
3529         {
3530             \setCJKmainfont
3531             [
3532                 cmap = UniGB-UTF16-H ,
3533                 BoldFont = AdobeHeitiStd-Regular.otf ,
3534                 ItalicFont = AdobeKaitiStd-Regular.otf
3535             ] { AdobeSongStd-Light.otf }
3536             \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3537             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3538             \setCJKfamilyfont { zhsong }
3539             [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3540             \setCJKfamilyfont { zhhei }
3541             [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3542             \setCJKfamilyfont { zhkai }
3543             [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3544             \setCJKfamilyfont { zhfs }
3545             [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3546             \ctex_punct_set:n { adobe }
3547             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3548             \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3549             \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3550             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3551             \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3552         }
3553         {
3554             \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3555             \ctex_punct_set:n { adobe }
3556             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3557             \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3558             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3559         }
3560         { \ctex_fontset_error:n { adobe } }
3561     }
3562 }
3563 {
3564     \setCJKmainfont
3565     [
3566         BoldFont = AdobeHeitiStd-Regular ,
3567         ItalicFont = AdobeKaitiStd-Regular
3568     ] { AdobeSongStd-Light }
3569     \setCJKsansfont { AdobeHeitiStd-Regular}
3570     \setCJKmonofont { AdobeFangsongStd-Regular}
3571     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3572     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3573     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3574     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3575 }

```

```
3576 </adobe>
3577 <*fandol>
```

15.15.4 ctex-fontset-fandol.def

```
3578 \sys_if_engine_pdftex:TF
3579 {
3580   \sys_if_output_pdf:TF
3581   { \ctex_fontset_error:n { fandol } }
3582   {
3583     \ctex_zhmap_case:nnn
3584     {
3585       \setCJKmainfont
3586       [
3587         cmap = UniGB-UTF16-H ,
3588         BoldFont = FandolSong-Bold.otf ,
3589         ItalicFont = FandolKai-Regular.otf
3590       ] { FandolSong-Regular.otf }
3591       \setCJKsansfont
3592       [
3593         cmap = UniGB-UTF16-H ,
3594         BoldFont = FandolHei-Bold.otf
3595       ] { FandolHei-Regular.otf }
3596       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3597       \setCJKfamilyfont { zhsong }
3598       [
3599         cmap = UniGB-UTF16-H ,
3600         BoldFont = FandolSong-Bold.otf
3601       ] { FandolSong-Regular.otf }
3602       \setCJKfamilyfont { zhhei }
3603       [
3604         cmap = UniGB-UTF16-H ,
3605         BoldFont = FandolHei-Bold.otf
3606       ] { FandolHei-Regular.otf }
3607       \setCJKfamilyfont { zhfs }
3608       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3609       \setCJKfamilyfont { zhkai }
3610       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
3611       \ctex_punct_set:n { fandol }
3612       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3613       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3614       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3615       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3616       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3617       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
3618     }
3619     {
3620       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
3621       \ctex_punct_set:n { fandol }
3622       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3623       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3624       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3625     }
3626     { \ctex_fontset_error:n { fandol } }
3627   }
3628 }
3629 {
3630   \setCJKmainfont
3631   [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
3632   { FandolSong-Regular }
3633   \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3634   \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
3635   \setCJKfamilyfont { zhsong }
3636   [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
3637   \setCJKfamilyfont { zhhei }
3638   [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3639   \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
```



```

3640     \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
3641   }
3642 </fandol>
3643 <*mac>

```

15.15.5 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...

```

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

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。

```

3644 \sys_if_engine_pdftex:TF
3645   { \ctex_fontset_error:n { mac } }
3646   {
3647     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
3648     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
3649     \setCJKmonofont { STFangsong }
3650     \setCJKfamilyfont { zhsong } { STSong }
3651     \setCJKfamilyfont { zhhei } { STHeiti }
3652     \setCJKfamilyfont { zhfs } { STFangsong }
3653     \setCJKfamilyfont { zhkai } { STKaiti }
3654   }
3655 </mac>
3656 <*founder>

```

15.15.6 ctex-fontset-founder.def

```

3657 \sys_if_engine_pdftex:TF
3658   {
3659     \ctex_zhmap_case:nnn
3660     {
3661       \setCJKmainfont
3662         [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
3663       \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
3664       \setCJKmonofont { FZFSK.TTF }
3665       \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
3666       \setCJKfamilyfont { zhhei } { FZHTK.TTF }
3667       \setCJKfamilyfont { zhkai } { FZKTK.TTF }
3668       \setCJKfamilyfont { zhfs } { FZFSK.TTF }
3669       \setCJKfamilyfont { zhli } { FZLSK.TTF }
3670       \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
3671       \ctex_punct_set:n { founder }
3672       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3673       \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
3674       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3675       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3676       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3677       \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
3678       \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
3679     }
3680     {
3681       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
3682       \ctex_punct_set:n { founder }
3683       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3684       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3685       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3686     }
3687     { \ctex_fontset_error:n { founder } }
3688   }
3689   {
3690     \setCJKmainfont
3691       [ 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），XeTeX 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 luaotload 不支持汉字名称，故使用 Postscript 名字，速度不受影响。

```

3692     \setCJKsansfont [ BoldFont = FZHei-B01 ]
3693     { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }

```

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

```

3694     \setCJKmonofont { FZFangSong-Z02 }
3695     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
3696     \setCJKfamilyfont { zhhei } { FZHei-B01 }
3697     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
3698     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
3699     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
3700     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
3701 }
3702 </founder>
3703 <!*ubuntu>

```

15.15.7 ctex-fontset-ubuntu.def

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

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

需要注意的是 uming.ttc 和 ukai.ttc 看起来像有四种字形的样子,但其实只有“令”和“骨”这区区两个字有新字形,其余都取旧字形²⁴。

```

3704 \sys_if_engine_pdftex:TF
3705 {
3706     \ctex_zhmap_case:nnn
3707     {
3708         \setCJKmainfont
3709         [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
3710         \setCJKsansfont { wqy-zenhei.ttc }
3711         \setCJKmonofont { uming.ttc }
3712         \setCJKfamilyfont { zhsong } { uming.ttc }
3713         \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
3714         \setCJKfamilyfont { zhkai } { ukai.ttc }
3715         \ctex_punct_set:n { ubuntu }
3716         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3717         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3718         \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
3719         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3720         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3721     }
3722     {
3723         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
3724         \ctex_punct_set:n { ubuntu }
3725         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3726         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3727         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3728     }
3729     { \ctex_fontset_error:n { ubuntu } }
3730 }
3731 {
3732     \setCJKmainfont
3733     [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
3734     \setCJKsansfont { WenQuanYi~Zen~Hei }
3735     \setCJKmonofont { AR~PL~UMing~CN }
3736     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
3737     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
3738     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
3739 }
3740 </ubuntu>

```

15.15.8 中文字体命令

```
<!*windows>
```

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

```

3742 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
3743 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
3744 (!ubuntu)\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
3745 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
3746 (*windowsnew|windowsold|founder)
3747 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
3748 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
3749 (<windowsnew|windowsold|founder)
3750 (<windowsnew)\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
3751 (</!windows)
3752 (</fontset)
3753 (<*zhmap)

```

15.15.9 zhmetrics 的字体映射

确认 `\catcode`, 没有重复载入检查。

```

3754 \begingroup\catcode61\catcode48\catcode32=10\relax%
3755 \catcode 35=6 % #
3756 \catcode123=1 % {
3757 \catcode125=2 % }
3758 \toks0{\endlinechar=\the\endlinechar\relax}%
3759 \toks2{\endlinechar=13 }%
3760 \def\x#1 #2 {%
3761 \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
3762 \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
3763 \x 13 5 % carriage return
3764 \x 32 10 % space
3765 \x 35 6 % #
3766 \x 40 12 % (
3767 \x 41 12 % )
3768 \x 45 12 % -
3769 \x 46 12 % .
3770 \x 47 12 % /
3771 \x 58 12 % :
3772 \x 60 12 % <
3773 \x 61 12 % =
3774 \x 64 11 % @
3775 \x 91 12 % [
3776 \x 93 12 % ]
3777 \x 123 1 % {
3778 \x 125 2 % }
3779 \edef\x#1{\endgroup%
3780 \edef\noexpand#1{%
3781 \the\toks0 %
3782 \let\noexpand\noexpand\noexpand#1%
3783 \noexpand\noexpand\noexpand\undefined%
3784 \noexpand\noexpand\noexpand\endinput}%
3785 \the\toks2}%
3786 \expandafter\x\csname ctex@zhmap@endinput\endcsname
3787 \begingroup\expandafter\endgroup
3788 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
3789 \expandafter\ifx\csname ifctexpdf\endcsname\relax
3790 \expandafter\ifx\csname pdfoutput\endcsname\relax
3791 iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
3792 \else ifctexpdf\fi
3793 \endcsname

```

提供非 \LaTeX 格式下的 `\ProvidesFile`。

```

3794 \begingroup
3795 \expandafter\ifx\csname ProvidesFile\endcsname\relax
3796 \long\def\x#1\ProvidesFile#2[#3]{%
3797 #1%
3798 \immediate\write-1{File: #2 #3}%
3799 \expandafter\xdef\csname ver@#2\endcsname{#3}}

```

```

3800 \expandafter\x%
3801 \fi
3802 \endgroup

```

15.15.9.1 zhwindowsfonts.tex

```

3803 (*windows)
3804 \ProvidesFile{zhwindowsfonts.tex}%
3805 [2016/02/02 v2.3 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
3806
3807 \ifzhmappdf
3808 \pdfmapline{=gbk@UGBK@ <simSun.ttc}
3809 \pdfmapline{=gbksong@UGBK@ <simSun.ttc}
3810 \pdfmapline{=gbkkai@UGBK@ <simKai.ttf}
3811 \pdfmapline{=gbkhei@UGBK@ <simHei.ttf}
3812 \pdfmapline{=gbkfs@UGBK@ <simFang.ttf}
3813 \pdfmapline{=gbkli@UGBK@ <simLi.ttf}
3814 \pdfmapline{=gbkyou@UGBK@ <simYou.ttf}
3815
3816 \pdfmapline{=cyberb@Unicode@ <simSun.ttc}
3817 \pdfmapline{=unisong@Unicode@ <simSun.ttc}
3818 \pdfmapline{=unikai@Unicode@ <simKai.ttf}
3819 \pdfmapline{=unihei@Unicode@ <simHei.ttf}
3820 \pdfmapline{=unifs@Unicode@ <simFang.ttf}
3821 \pdfmapline{=unili@Unicode@ <simLi.ttf}
3822 \pdfmapline{=uniyou@Unicode@ <simYou.ttf}
3823
3824 \pdfmapline{=gbksongsl@UGBK@ <simSun.ttc}
3825 \pdfmapline{=gbkkaisl@UGBK@ <simKai.ttf}
3826 \pdfmapline{=gbkheisl@UGBK@ <simHei.ttf}
3827 \pdfmapline{=gbkfssl@UGBK@ <simFang.ttf}
3828 \pdfmapline{=gbklisl@UGBK@ <simLi.ttf}
3829 \pdfmapline{=gbkyousl@UGBK@ <simYou.ttf}
3830
3831 \pdfmapline{=unisongsl@Unicode@ <simSun.ttc}
3832 \pdfmapline{=unikaisl@Unicode@ <simKai.ttf}
3833 \pdfmapline{=uniheisl@Unicode@ <simHei.ttf}
3834 \pdfmapline{=unifssl@Unicode@ <simFang.ttf}
3835 \pdfmapline{=unilisl@Unicode@ <simLi.ttf}
3836 \pdfmapline{=uniyousl@Unicode@ <simYou.ttf}
3837
3838 \else
3839 \special{pdf:mapline gbk@UGBK@ unicode :0:simSun.ttc -v 50}
3840 \special{pdf:mapline gbksong@UGBK@ unicode :0:simSun.ttc -v 50}
3841 \special{pdf:mapline gbkkai@UGBK@ unicode simKai.ttf -v 70}
3842 \special{pdf:mapline gbkhei@UGBK@ unicode simHei.ttf -v 150}
3843 \special{pdf:mapline gbkfs@UGBK@ unicode simFang.ttf -v 50}
3844 \special{pdf:mapline gbkli@UGBK@ unicode simLi.ttf -v 150}
3845 \special{pdf:mapline gbkyou@UGBK@ unicode simYou.ttf -v 60}
3846
3847 \special{pdf:mapline cyberb@Unicode@ unicode :0:simSun.ttc -v 50}
3848 \special{pdf:mapline unisong@Unicode@ unicode :0:simSun.ttc -v 50}
3849 \special{pdf:mapline unikai@Unicode@ unicode simKai.ttf -v 70}
3850 \special{pdf:mapline unihei@Unicode@ unicode simHei.ttf -v 150}
3851 \special{pdf:mapline unifs@Unicode@ unicode simFang.ttf -v 50}
3852 \special{pdf:mapline unili@Unicode@ unicode simLi.ttf -v 150}
3853 \special{pdf:mapline uniyou@Unicode@ unicode simYou.ttf -v 60}
3854
3855 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simSun.ttc -s .167 -v 50}
3856 \special{pdf:mapline gbkkaisl@UGBK@ unicode simKai.ttf -s .167 -v 70}
3857 \special{pdf:mapline gbkheisl@UGBK@ unicode simHei.ttf -s .167 -v 150}
3858 \special{pdf:mapline gbkfssl@UGBK@ unicode simFang.ttf -s .167 -v 50}
3859 \special{pdf:mapline gbklisl@UGBK@ unicode simLi.ttf -s .167 -v 150}
3860 \special{pdf:mapline gbkyousl@UGBK@ unicode simYou.ttf -s .167 -v 60}
3861
3862 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simSun.ttc -s .167 -v 50}
3863 \special{pdf:mapline unikaisl@Unicode@ unicode simKai.ttf -s .167 -v 70}

```

```

3864 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
3865 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
3866 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
3867 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
3868
3869 </windows>

```

15.15.9.2 zhadobefonts.tex

```

3870 (*adobe)
3871 \ProvidesFile{zhadobefonts.tex}%
3872 [2016/02/02 v2.3 Adobe font map loader for DVIPDFMx (CTEX)]
3873
3874 \ifzhmappdf
3875 %% pdfTeX does not support OTF fonts
3876
3877 \else
3878 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3879 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3880 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3881 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3882 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3883
3884 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3885 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3886 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3887 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3888 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3889
3890 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3891 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3892 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3893 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3894
3895 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3896 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3897 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3898 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3899
3900 </adobe>

```

15.15.9.3 zhfandolfonts.tex

```

3901 (*fandol)
3902 \ProvidesFile{zhfandolfonts.tex}%
3903 [2016/02/02 v2.3 Fandol font map loader for DVIPDFMx (CTEX)]
3904
3905 \ifzhmappdf
3906 %% pdfTeX does not support OTF fonts
3907
3908 \else
3909 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3910 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3911 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
3912 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
3913 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
3914
3915 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3916 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3917 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
3918 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
3919 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
3920
3921 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3922 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3923 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3924 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3925

```



```

3926 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3927 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3928 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3929 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3930
3931 (/fandol)

```

15.15.9.4 zhfounderfonts.tex

```

3932 (*founder)
3933 \ProvidesFile{zhfounderfonts.tex}%
3934 [2016/02/02 v2.3 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
3935
3936 \ifzhmappdf
3937 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
3938 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
3939 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
3940 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
3941 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
3942 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
3943 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
3944
3945 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
3946 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
3947 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
3948 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
3949 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
3950 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
3951 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
3952
3953 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
3954 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
3955 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
3956 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
3957 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
3958 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
3959
3960 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
3961 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
3962 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
3963 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
3964 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
3965 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
3966
3967 \else
3968 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
3969 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
3970 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
3971 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
3972 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
3973 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
3974 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
3975
3976 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
3977 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
3978 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
3979 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
3980 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
3981 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
3982 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
3983
3984 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
3985 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
3986 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
3987 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
3988 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
3989 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
3990

```

```

3991 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
3992 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
3993 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
3994 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
3995 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
3996 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
3997
3998 </founder>

```

15.15.9.5 zhubuntufonts.tex

```

3999 <*ubuntu>
4000 \ProvidesFile{zhubuntufonts.tex}%
4001 [2016/02/02 v2.3 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4002
4003 \ifzhmappdf
4004 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4005 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4006 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4007 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4008 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4009 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4010
4011 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4012 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4013 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4014 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4015 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4016 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4017
4018 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4019 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4020 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4021 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4022 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4023
4024 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4025 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4026 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4027 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4028 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4029
4030 \else
4031 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4032 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4033 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4034 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4035 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4036
4037 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4038 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4039 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4040 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4041 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4042
4043 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4044 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4045 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4046 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4047
4048 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4049 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4050 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4051 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4052
4053 </ubuntu>
4054 \fi

```



```

4055
4056 \ctex@zhmap@endinput
4057 </zhmap>

```

15.15.10 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。

```

4058 (*spa)
4059 (*macro)
4060 \input expl3-generic %
4061 \ExplSyntaxOn
4062 \sys_if_engine_xetex:F
4063 {
4064   \msg_new:nn { ctex } { xetex }
4065   { XeTeX~is~required~to~compile~this~document! }
4066   \msg_fatal:nn { ctex } { xetex }
4067 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

4068 \seq_new:N \c__ctex_punct_seq
4069 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4070 {
4071   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4072   "3008 , "300A , "3016 , "3010 ,
4073   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4074   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4075   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4076 }

```

`\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。

```

4077 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4078 {
4079   \iow_open:Nn \g__ctex_spa_iow {#1}
4080   \clist_map_inline:nn {#2}
4081   { \__ctex_write_family:nn ##1 }
4082   \iow_close:N \g__ctex_spa_iow
4083 }
4084 \iow_new:N \g__ctex_spa_iow
4085 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

(End definition for \ctex_make_spa:nn)

4086 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4087 {
4088   \group_begin:
4089   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4090   \l__ctex_punct_font
4091   \clist_clear:N \l__ctex_punct_bounds_clist
4092   \seq_map_inline:Nn \c__ctex_punct_seq
4093   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4094   \iow_now:Nx \g__ctex_spa_iow
4095   {
4096     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4097   { \l__ctex_punct_bounds_clist , , , }
4098 }

```

```

4099   \group_end:
4100 }
4101 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4102 {
4103   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4104   {
4105     \__ctex_calc_bounds:nn { \c_one } {#1} ,
4106     \__ctex_calc_bounds:nn { \c_three } {#1}
4107   }
4108 }
4109 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1 em 的比值的一百倍。

```

4110 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4111 {
4112   \fp_eval:n
4113   {
4114     round
4115     (
4116       \dim_to_decimal_in_unit:nn
4117       { 100 \xetex_glyphbounds:D #1 ~ #2 }
4118       { 1 em }
4119     )
4120   }
4121 }
4122 \ExplSyntaxOff
4123 </macro>

```

下面是 CTeX 定义的一些字体。

```

4124 <*make>
4125 \input ctexspamacro %
4126
4127 \MAKESPA {ctexpunct.spa}
4128 {
4129   {adobezhsong}      {AdobeSongStd-Light} ,
4130   {adobezhhei}      {AdobeHeitiStd-Regular} ,
4131   {adobezhkai}      {AdobeKaitiStd-Regular} ,
4132   {adobezhfs}       {AdobeFangsongStd-Regular} ,
4133   {fandolzhsong}    {FandolSong} ,
4134   {fandolzhsongb}   {FandolSong-Bold} ,
4135   {fandolzhhei}     {FandolHei} ,
4136   {fandolzhheib}    {FandolHei-Bold} ,
4137   {fandolzhkai}     {FandolKai} ,
4138   {fandolzhfs}      {FandolFang} ,
4139   {founderzhsong}   {FZShuSong-Z01} ,
4140   {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
4141   {founderzhhei}    {FZHei-B01} ,
4142   {founderzhheil}   {FZXiHeiI-Z08} ,
4143   {founderzhkai}    {FZKai-Z03} ,
4144   {founderzhfs}     {FZFangSong-Z02} ,
4145   {founderzhli}     {FZLiShu-S01} ,
4146   {founderzhyou}    {FZXiYuan-M01} ,
4147   {founderzhyoub}   {FZZhunYuan-M02} ,
4148   {buntuzhsong}     {AR PL UMinG CN} ,
4149   {buntuzhhei}      {WenQuanYi Zen Hei} ,
4150   {buntuzhkai}      {AR PL UKai CN} ,
4151   {windowszhsong}   {SimSun} ,
4152   {windowszhhei}    {SimHei} ,
4153   {windowszhkai}    {KaiTi} ,
4154   {windowszhfs}     {FangSong} ,
4155   {windowszhli}     {LiSu} ,
4156   {windowszhyou}    {YouYuan} ,
4157   {windowszhyahei}  {Microsoft YaHei} ,
4158   {windowszhyaheib} {Microsoft YaHei Bold}
4159 }
4160

```

```

4161 \primitive\end
4162 </make>
4163 </spa>

```

15.16 ctexcap 宏包

```
4164 (*ctexcap)
```

ctexcap 是过时宏包。

```

4165 \clist_new:N \l__ctex_ctexcap_options_clist
4166 \clist_set:Nx \l__ctex_ctexcap_options_clist
4167 { \exp_not:v { opt@ \@currname . \@currentext } , heading }
4168 \msg_new:nnn { ctexcap } { deprecated }
4169 {
4170   Package~`ctexcap'~is~deprecated.\\
4171   Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
4172   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4173 }
4174 \msg_warning:nnx { ctexcap } { deprecated }
4175 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

4176 \PassOptionsToPackage { heading = true } { ctexcap }
4177 \RequirePackageWithOptions { ctex }
4178 </ctexcap>

```

15.17 ctexhook 宏包

```
4179 (*ctexhook)
```

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4180 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4181 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
4182 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4183 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
4184 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4185 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
4186 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4187 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4188 \cs_set_nopar:Npx \document
4189 {
4190   \CTEX@document@left@hook
4191   \exp_not:o { \document }
4192   \CTEX@document@right@hook
4193 }
4194 \tl_new:N \g__ctex_end_preamble_hook_tl
4195 \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。

```

4196 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4197 {
4198   \@ifpackageloaded {#1}
4199     {#2}
4200     { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
4201 }

```

(End definition for \ctex_at_end_package:nn.)

```

\ctex_gadd_hook:Nn 给钩子附加内容。
\ctex_gadd_hook:cn
4202 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4203 {
4204   \tl_if_exist:NF #1 { \tl_new:N #1 }
4205   \tl_gput_right:Nn #1 {#2}
4206 }
4207 \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
4208 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
4209 {
4210   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
4211   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
4212 }
4213 \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。

```

4214 \tl_put_left:Nn \@popfilename
4215 {
4216   \cs_if_eq:NNT \@currentx \@pkgextension
4217   { \ctex_package_end_hook:o { \@currname } }
4218 }
4219 </ctexhook>

```

15.18 ctexpatch 宏包

```

4220 <*ctexpatch>
4221 \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。

```

4222 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
4223 {
4224   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4225   { once } {#2} { \use_i:nn } { \use_ii:nn }
4226 }

```

(End definition for \ctex_patch_cmd_once:NnnnTF.)

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

```

4227 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
4228 {
4229   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4230   { all } {#2} { \use_i:nn } { \use_ii:nn }
4231 }

```

(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。

```

4232 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
4233 {
4234   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
4235   { once }
4236   {
4237     \ExplSyntaxOff
4238     \char_set_catcode_letter:n { 64 }

```

```

4239     }
4240     { }
4241     { \ctex_patch_failure:N #1 }
4242   }
4243   \cs_new_protected:Npn \ctex_patch_failure:N #1
4244     { \msg_warning:nxx { ctex } { patch-failure } { \token_to_str:N #1 } }
4245   \msg_new:nnn { ctex } { patch-failure }
4246     { 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。

```

4247   \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
4248     {
4249       \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
4250       { left } {#2} { \use_i:nn } { \use_ii:nn }
4251     }

```

(End definition for \ctex_preto_cmd:NnnTF.)

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```

4252   \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
4253     {
4254       \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
4255       { right } {#2} { \use_i:nn } { \use_ii:nn }
4256     }

```

(End definition for \ctex_appto_cmd:NnnTF.)

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

```

4257   \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
4258     {
4259       \tl_set:Nn \__ctex_patch_true:w {#5}
4260       \tl_set:Nn \__ctex_patch_false:w {#6}
4261       \group_begin:
4262         \char_set_catcode_other:n { 35 }
4263         \ctex_parse_name:NN #1 #2 {#3} {#4}
4264     }
4265   \tl_new:N \__ctex_patch_true:w
4266   \tl_new:N \__ctex_patch_false:w

```

(End definition for \ctex_patch_boot:NNnnTF.)

\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_\u          % \DeclareRobustCommand\xaa[1]{...}
2 \protect_\xab_\u          % \DeclareRobustCommand\xab[1][]{...}
3 \@protected@testopt_\xac_\u\ \xac_\u{} % \newcommand\xac[1][]{...}
4 \@testopt_\xad_\u{}       % \newrobustcmd\xad[1][]{...}
5 \x@protect_\1\protect_\1_\u % \DeclareRobustCommand\1[1]{...}
6 \x@protect_\2\protect_\2_\u % \DeclareRobustCommand\2[1][]{...}
7 \@protected@testopt_\3\3_\u{} % \newcommand\3[1][]{...}
8 \@testopt_\4_\u{}         % \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。

```

4267 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
4268 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
4269 \group_begin:
4270 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
4271 {
4272   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
4273   {
4274     \bool_if:nTF
4275     {
4276       \cs_if_exist_p:c { ##3 ~ } ||
4277       \cs_if_exist_p:c { #1##3 }
4278     }
4279     {
4280       \group_begin:
4281       \use:x
4282       {
4283         \group_end:
4284         \__ctex_parse_name:nNNNnN
4285         { \token_get_replacement_spec:N ##2 }
4286         \exp_not:N ##2
4287         \exp_not:c { ##3 ~ }
4288         \exp_not:c { #1##3 }
4289         } {##3} ##1
4290       }
4291       { ##1##2 }
4292     }
4293   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
4294   {
4295     \exp_args:Nc ##6
4296     {
4297       \str_case:nnTF {##1}
4298       {
4299         { \protect ##3 } { }
4300         { \x@protect ##2 \protect ##3 } { }
4301       }
4302       {
4303         \str_if_eq_x:nnTF
4304         { \exp_not:n { #1@protected@ ##3 #1##3 } }
4305         {
4306           \exp_last_unbraced:Nf \__ctex_parse_name:w
4307           \token_get_replacement_spec:N ##3 #2 ~ #1 \q_stop
4308         }
4309         { #1##5 ~ } { ##5 ~ }
4310       }
4311       {
4312         \str_case:onTF { \__ctex_parse_name:w ##1 #2 ~ #1 \q_stop }
4313         {
4314           { #1@protected@ ##2 ##4 } { }
4315           { #1@ ##4 } { }
4316         }
4317         { #1##5 } {##5}
4318       }
4319     }
4320   }
4321   \cs_new:Npn \__ctex_parse_name:w ##1 #2 ~ ##2 #1 ##3 \q_stop { ##1##2 }
4322 }
4323 \use:x
4324 {
4325   \__ctex_tmp:w
4326   { \c_backslash_str }
4327   { \c_left_brace_str }
4328   { \tl_to_str:n { testopt } }
4329 }
4330 \group_end:
4331 \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`

```
4332 \str_new:N \l__ctex_prefix_str
4333 \str_new:N \l__ctex_parameter_str
4334 \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
4335 \group_begin:
4336 \cs_set_protected:Npn \__ctex_tmp:w #1
4337 {
4338   \prg_new_protected_conditional:Npnn
4339   \ctex_get_macro_meaning:N ##1 { TF }
4340   {
4341     \exp_after:wN \__ctex_get_macro_meaning:w
4342     \token_to_meaning:N ##1 \q_mark #1 -> \q_no_value \q_mark \q_stop
4343   }
4344   \cs_new_protected:Npn \__ctex_get_macro_meaning:w
4345   ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
4346   {
4347     \tl_set:Nn \l__ctex_replacement_tl {##3}
4348     \quark_if_no_value:NTF \l__ctex_replacement_tl
4349     { \prg_return_false: }
4350     {
4351       \str_set:Nn \l__ctex_prefix_str {##1}
4352       \str_set:Nn \l__ctex_parameter_str {##2}
4353       \str_set:Nn \l__ctex_replacement_str {##3}
4354       \prg_return_true:
4355     }
4356   }
4357 }
4358 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
4359 \group_end:
```

(End definition for `\ctex_get_macro_meaning:NTF`.)

`\ctex_if_rescanable:NnTF` 检查宏是否可以重建。

```
4360 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
4361 {
4362   \ctex_get_macro_meaning:NTF #1
4363   {
4364     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
4365     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
4366   }
4367   {#4}
4368 }
4369 \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。

```
4370 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
4371 {
4372   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
4373   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
4374   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
4375   \use:x
4376   {
4377     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
4378     \exp_not:o { \l__ctex_parameter_tl }
4379     { \exp_not:o { \l__ctex_replacement_tl } }
4380   }
4381 }
```

```

4382 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
4383 {
4384   \str_if_empty:NTF #2
4385     { \tl_clear:N #1 }
4386     { \tl_set_rescan:Nnx #1 {#3} {#2} }
4387 }
4388 \tl_new:N \l__ctex_prefix_tl
4389 \tl_new:N \l__ctex_parameter_tl
4390 \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。

```

4391 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
4392 {
4393   \group_end:
4394   \ctex_if_rescanable:NnTF #1 {#3}
4395     {
4396       \use:x
4397       {
4398         \__ctex_patch_replace:nnnTF {#2}
4399         { \tl_to_str:n {#4} }
4400         { \tl_to_str:n {#5} }
4401       }
4402       {
4403         \__ctex_patch_rebuild:Nn #1 {#3}
4404         \__ctex_patch_true:w
4405       }
4406       { \__ctex_patch_false:w }
4407     }
4408     { \__ctex_patch_false:w }
4409 }

```

(End definition for __ctex_patch_cmd:Nnnnw.)

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

```

4410 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
4411 {
4412   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
4413     { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
4414 }

```

(End definition for __ctex_patch_replace:nnnTF.)

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

4415 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
4416 {
4417   \group_end:
4418   \ctex_get_macro_meaning:NTF #1
4419     {
4420       \str_if_empty:NTF \l__ctex_parameter_str
4421         { \__ctex_hookto_cmd_parameterless:Nnnw #1 {#2} {#4} }
4422         { \__ctex_hookto_cmd_parameter:Nnnnw #1 {#2} {#3} {#4} }
4423     }
4424     { \__ctex_patch_false:w }
4425 }

```

(End definition for __ctex_hookto_cmd:Nnnnw.)

__ctex_hookto_cmd_parameterless:Nnnw 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。

```

4426 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnw #1#2#3
4427 {

```



```

4428 \str_if_empty:NF \l__ctex_prefix_str
4429 { \tl_rescan:nx { } { \l__ctex_prefix_str } }
4430 \tex_edef:D #1
4431 {
4432   \use:c { __ctex_ #2 _hook_aux:nn }
4433   { \exp_not:o {#1} }
4434   { \exp_not:n {#3} }
4435 }
4436 \__ctex_patch_true:w
4437 }
4438 \cs_generate_variant:Nn \tl_rescan:nn { nx }
4439 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
4440 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

(End definition for __ctex_hookto_cmd_parameterless:Nnnw.)

__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。

```

4441 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
4442 {
4443   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
4444   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
4445   {
4446     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
4447     \__ctex_patch_rebuild:Nn #1 {#3}
4448     \__ctex_patch_true:w
4449   }
4450   { \__ctex_patch_false:w }
4451 }

```

(End definition for __ctex_hookto_cmd_parameter:Nnnnw.)

4452 </ctexpatch>

版本历史

v2.0

- General: c5size, cs4size 是过时选项。 35
- captiondelimitter 是过时选项。 102
- fancyhdr 成为过时选项,原选项功能总是打开。 38
- fnctef 成为过时选项,原选项功能总是打开。 38
- hyperref 成为过时选项,原选项功能总是打开。 38
- indent, noindent 是过时选项。 35
- nofonts, adobefonts, winfonts 是过时选项。 36
- nopunct 是过时选项。 37
- nospace 是过时选项。 37
- nozhmap 是过时选项。 37
- punct 选项可以设置标点格式。 37
- ctex 宏包新增 heading 选项。 37
- \CTEXindent, \CTEXnoindent 是过时命令。 77
- \CTEXsetup, \CTEXoptions 是过时命令。 14, 40
- \CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; CTEXfilltwosides 是过时环境。 70
- 标题设置新增 pagestyle 选项。 80
- 调整 \footnotesep 的大小,以适合行距的变化。 101
- 兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。 39
- 将标题汉化功能加入 ctex.sty。 78
- 将中文字号功能提取到可以独立使用的 ctexsize。 98
- 解决 etoolbox 与 breqn 关于 \end 的冲突。 102
- 默认关闭 CJKfnctef 或 xeCJKfnctef 的彩色设置。 70
- 删除 c19gbsn.fd 和 c19gkai.fd。 1
- 通过 LuaTeX-ja 宏包支持 Lua^AT_EX。 46
- 新增 autoindent 选项。 35
- 新增 fontset 选项。 36
- 新增 linespread 选项。 35
- 新增 linestretch 选项。 75
- 新增 scheme 选项,并将 cap 和 nocap 列为过时选项。 38
- 新增 zhmCJK 支持选项。 37
- 新增 zihao 选项。 35
- 新增统一设置接口 \ctexset。 40
- 应用 ^AT_EX3 重新整理代码。 1
- 中文字号不再采用近似值。 98
- 自动检测操作系统,载入对应的字体配置。 68
- \CJK@surr: 解决与 \nouppercase 的冲突。 43

v2.0.1

- General: 修复 10pt、11pt 等选项无效的问题。 39

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。 98

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。 79
- nameformat 可以接受章节名字为参数。 79
- 不依赖 ifpdf 宏包。 33
- 不再设置 hyperref 宏包的 colorlinks 选项。 68
- 给 pdf^AT_EX 下的非 UTF8 编码 CJK 字体族加上 CMap。 40
- 将章节标题设置功能提取到可以独立使用的宏包 ctex-heading 中。 1
- 新的标题格式选项 aftertitle。 79
- 修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。 93
- \ctex_if_macosx:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。 68
- hyperref: 补充定义 \hypersetup。 38

v2.2

- General: beforekip 和 afterskip 选项的符号不再有特殊意义。 79
- beforekip、afterskip 和 indent 选项支持表达式。 81
- 不再依赖 etoolbox 宏包。 1
- 非 ctexart 类的 part/beforekip 和 part/afterskip 选项有意义。 81
- 给 enumitem 宏包注册 \chinese 和 \zhnum。 77
- 将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。 119
- 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。 93
- 删去 etoolbox 与 breqn 的兼容补丁。 102
- 新的标题格式选项 afterindent。 79
- 新的标题格式选项 numbering。 79
- 新的标题格式选项 runin。 79
- 新增子宏包 ctexpatch 实现给宏打补丁功能。 120

v2.3

- General: .value_required: 和 .value_forbidden: 已过时。 35
- 代码实现避免使用 \lowercase 技巧 (Joseph Wright)。 33
- 更新 LuaTeX-ja 支持 (20150922.0)。 46
- 更新 unicode-math 宏包补丁。 47
- 兼容 titleps 宏包。 90
- 修复 nameformat 作用域问题。 79
- 与 ^AT_EX3 (2015/12/20) 同步。 31

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	Page
_	2750, 2754, 2770, 2774
\#	471, 2671, 2717
.../afterindent	18
.../aftername	17
.../aftername+	17

.../afterskip 19

.../aftertitle 18

.../aftertitle+ 18

.../beforeskip 19

.../format 15

.../format+ 15

.../indent 19

.../name 14

.../nameformat 16

.../nameformat+ 16

.../number 15

.../numberformat 16

.../numberformat+ 16

.../titleformat 17

.../titleformat+ 17

\/ 1620

\\ 15, 33,
34, 98, 141, 143, 145, 201, 211, 658, 1101, 1183, 2099,
3008, 3009, 3080, 3081, 3301, 3331, 4170, 4171, 4172, 4246

\{ 1112, 1113, 4172

\} 1112, 1113, 4172

10pt 8, 39

11pt 8, 39

12pt 8, 39

A

\abovedisplayshortskip . 3171, 3179, 3190, 3210, 3218, 3229

\abovedisplayskip 3170, 3173,
3178, 3185, 3189, 3196, 3209, 3212, 3217, 3224, 3228, 3235

\abstractname 2090

abstractname 12

\addCJKfontfeature 1183, 1206

\addCJKfontfeatures 57, 1201, 1206

\addcontentsline 2258, 2259, 2262, 2265,
2288, 2289, 2292, 2295, 2369, 2374, 2377, 2380, 2478, 2494

\AddEnumerateCounter 2051, 2053, 2054

\addpenalty 2445, 2446

\addtocontents 2383, 2384

\addvspace 2225, 2226, 2383, 2384, 2445, 2446

adobe-fonts 27

AlternateFont 24, 60

\appendix 88, 2586, 2587

appendix/name 20, 87

appendix/number 21, 87

appendix/numbering 20, 87

\appendixname 2092, 2994

appendixname 12

\AtBeginDocument 561, 1609

\AtBeginDvi 497

\AtBeginShipoutFirst 499, 1722

\AtBeginUTFCommand 702

\AtEndOfClass 131

\AtEndOfPackage 132

\AtEndUTFCommand 708

autoindent 10, 35, 74

B

backslash commands:

\c_backslash_str 4326

\baselinestretch 1962

\begingroup 2469, 2508, 2538, 3754, 3787, 3794

\belowdisplayshortskip . 3172, 3180, 3191, 3211, 3219, 3230

\belowdisplayskip 3173, 3185, 3196, 3212, 3224, 3235

\bfdefault 964, 970, 1295, 1302

\bfseries 2271, 2274, 2302, 2305, 2316,
2329, 2400, 2406, 2420, 2826, 2828, 2832, 2844, 2845,
2848, 2863, 2865, 2870, 2886, 2890, 2897, 2913, 2929, 2942

\bibname 2094, 2099, 2102, 2103

bibname 12

bool commands:

\bool_gset_false:N 261, 266

\bool_gset_true:N 256

\bool_if:cTF 2637, 2647, 2658

\bool_if:NT 1141

\bool_if:NTF 68, 3012

\bool_if:nTF 4274

\bool_new:N 127, 1178

\bool_set_false:N 1153

\bool_set_true:N 1171

box commands:

\box_ht:N 3251

\box_new:N 59

\box_wd:N 1801, 2713

C

c5size 27

cap 27

captiondelimiter 28, 102

\catcode 3754, 3755, 3756, 3757, 3761, 3762

CCT 27

CCTfont 27

\ccwd 23, 71, 111, 184, 1801, 1805, 1807,
1837, 1838, 1913, 1951, 1970, 1971, 1978, 2005, 2006,
2007, 2010, 2012, 2013, 2024, 2035, 2036, 3268, 3269, 3270

\centering 2299, 2327, 2832, 2843, 2848, 2870, 2890

\chapter 83, 2351, 3018

chapter/beforeskip 29

chapter/numbering 14

chapter/pagestyle 20, 80

\chaptermark 2382

\chaptername 2860

char commands:

\char_generate:nn 728

\char_set_catcode:nn 71, 534, 535

\char_set_catcode_active:N 1330

\char_set_catcode_letter:n 65, 1612, 2731, 4238

\char_set_catcode_math_superscript:n 529

\char_set_catcode_other:N . 471, 897, 1313, 2671, 2717

\char_set_catcode_other:n 528, 4262

\char_value_catcode:n 71, 534, 535

CharRange 24, 60

\Chinese 28, 2047

- \chinese 23, 77, 2046, 2047, 2054, 2819, 2869
- \CJK 1766
- \CJKbold
 - 3373, 3374, 3384, 3385, 3391, 3392, 3394, 3395, 3397,
 - 3398, 3402, 3403, 3405, 3406, 3408, 3409, 3415, 3416,
 - 3418, 3419, 3421, 3422, 3426, 3427, 3429, 3430, 3432, 3433
- \CJKfamily . 55, 508, 512, 516, 520, 1038, 1060, 1240, 1244,
 - 1248, 1252, 1399, 3742, 3743, 3744, 3745, 3747, 3748, 3750
- \CJKfamilydefault . 502, 503, 520, 575, 1234, 1235, 1252,
 - 1259, 1261, 1265, 1272, 1285, 1626, 1630, 1638, 1652, 1654
- \CJKfilltwosides 1774
- \CJKfontspec 57, 1196
- \CJKglue 1800, 1811, 1825, 1834
- \CJKhook 44, 590
- \CJKnormal 3366, 3367, 3368, 3369, 3370, 3371, 3372,
 - 3377, 3378, 3379, 3380, 3381, 3382, 3383, 3390, 3393,
 - 3396, 3401, 3404, 3407, 3414, 3417, 3420, 3425, 3428, 3431
- \CJKpunctmapfamily
 - . 623, 624, 625, 626, 627, 628, 632, 633, 634, 635, 636, 637
- \CJKrmdefault . 494, 503, 504, 508, 1108, 1119, 1209, 1235,
 - 1236, 1240, 1261, 1264, 1265, 1634, 3476, 3478, 3479,
 - 3484, 3485, 3486, 3489, 3547, 3550, 3551, 3556, 3557,
 - 3558, 3612, 3615, 3616, 3622, 3623, 3624, 3672, 3675,
 - 3676, 3683, 3684, 3685, 3716, 3719, 3720, 3725, 3726, 3727
- \CJKsfdefault
 - ... 495, 505, 512, 1109, 1120, 1215, 1237, 1244, 1635,
 - 3452, 3466, 3467, 3490, 3548, 3613, 3617, 3673, 3677, 3717
- \CJKtilde 576
- \CJKttdefault 496, 506, 516, 1110, 1121,
 - 1220, 1238, 1248, 1636, 3477, 3491, 3549, 3614, 3674, 3718
- clearalternatefont 25, 64
- \cleardoublepage 2237, 2351
- \clearpage 2239, 2351
- clist commands:
 - \clist_clear:N 985, 4091
 - \clist_concat:NNN 1007
 - \clist_const:Nn 603
 - \clist_gput_right:Nn 376, 398, 400
 - \clist_gput_right:No 388
 - \clist_gset:Nn 1226
 - \clist_if_empty:NT 1318, 1577
 - \clist_if_empty:NTF 1337
 - \clist_map_break:n 2616, 3003
 - \clist_map_function:nN 1528, 1542
 - \clist_map_function:xN 1494, 1496
 - \clist_map_inline:Nn 593, 1430
 - \clist_map_inline:nn
 - . 361, 615, 1535, 1758, 1784, 2611, 3000, 3098, 3133, 4080
 - \clist_new:N 134, 1004, 1227, 1372, 4109, 4165
 - \clist_put_left:Nx 1008
 - \clist_put_right:Nn 1170
 - \clist_put_right:Nx 1418, 4103
 - \clist_set:Nn 1355, 1356
 - \clist_set:No 989
 - \clist_set:Nx 4166
 - \clist_use:Nn 4175
- \clubpenalty 2537, 2544
- colon commands:
 - \c_colon_str 922, 923, 924
- \contentsname 2085
- contentsname 11
- cs commands:
 - \cs:w 2795
 - \cs_end: 2795
 - \cs_generate_variant:Nn
 - 91, 447, 459, 745, 949, 1072, 1086, 1159, 1179,
 - 1180, 1351, 1388, 1490, 1528, 1536, 4207, 4213, 4331, 4438
 - \cs_gset_eq:cc 1146, 1149
 - \cs_gset_eq:NN 157, 158,
 - 159, 257, 262, 267, 304, 306, 371, 375, 479, 792, 796, 3058
 - \cs_gset_nopar:Npx 793
 - \cs_gset_protected_nopar:cpx 1034, 2565
 - \cs_gset_protected_nopar:Npn 733, 2219, 2587
 - \cs_gset_protected_nopar:Npx 831, 1392, 1486
 - \cs_if_eq:cNF 933
 - \cs_if_eq:NNT 4216
 - \cs_if_eq:NNTF 4365, 4444
 - \cs_if_exist:cF 787, 829, 867
 - \cs_if_exist:cTF 2782
 - \cs_if_exist:NF 352, 456, 799, 1391, 1483, 1854, 3020, 4221
 - \cs_if_exist:NTF 3018
 - \cs_if_exist_p:c 4276, 4277
 - \cs_if_exist_use:c 1056, 1057
 - \cs_if_exist_use:cF 4, 770
 - \cs_if_exist_use:cT 1505, 4210
 - \cs_if_exist_use:cTF 1078
 - \cs_if_free:cF 441, 595, 610, 1143, 2630, 2696, 2701
 - \cs_if_free:cTF 1294
 - \cs_if_free:NF 451, 1940, 2051, 2705
 - \cs_if_free:NTF 1926
 - \cs_new:Npn
 - . 106, 113, 120, 888, 894, 898, 981, 1059, 2655, 4321, 4439
 - \cs_new_eq:cc 1793
 - \cs_new_eq:cN 6, 7, 8, 9, 20, 23, 1779
 - \cs_new_eq:Nc 1845
 - \cs_new_eq:NN 92, 102, 103, 104, 129, 130,
 - 148, 353, 448, 580, 641, 808, 833, 909, 940, 1206, 1212,
 - 1648, 1780, 1781, 1834, 1844, 2046, 2047, 2048, 2557,
 - 2558, 2559, 2560, 2586, 2800, 3016, 3021, 4085, 4369, 4440
 - \cs_new_nopar:Npn 848, 1051, 1104, 1115, 2694, 2699, 4110
 - \cs_new_protected:Npn 60,
 - 726, 731, 982, 1001, 1316, 1407, 1421, 1572, 1839,
 - 2196, 2707, 3153, 3159, 4086, 4180, 4182, 4196, 4202,
 - 4222, 4227, 4232, 4243, 4247, 4252, 4257, 4267, 4272,
 - 4293, 4344, 4360, 4370, 4382, 4391, 4410, 4415, 4426, 4441
 - \cs_new_protected_nopar:cpx 1763
 - \cs_new_protected_nopar:Npn
 - 75, 77, 131, 132, 439, 449, 454, 460, 492, 524,
 - 541, 582, 588, 591, 608, 613, 621, 630, 639, 710, 768,
 - 777, 790, 795, 797, 809, 823, 850, 865, 878, 902, 911,
 - 928, 942, 959, 1005, 1013, 1032, 1052, 1062, 1087, 1125,
 - 1137, 1139, 1161, 1163, 1255, 1280, 1289, 1335, 1346,

- 1373, 1382, 1389, 1395, 1415, 1428, 1447, 1461, 1468,
1480, 1500, 1516, 1534, 1537, 1539, 1552, 1564, 1598,
1617, 1624, 1657, 1675, 1691, 1702, 1706, 1738, 1745,
1749, 1770, 1776, 1797, 1808, 1836, 1925, 1927, 1952,
1987, 1997, 2002, 2122, 2152, 2217, 2550, 2603, 2623,
2628, 2633, 2643, 2653, 2672, 2778, 2787, 2792, 3071,
3086, 3118, 3123, 3273, 3277, 4077, 4101, 4184, 4186, 4208
- `\cs_new_protected_nopar:Npx` 464, 1192
- `\cs_set:Npn` 1314
- `\cs_set_eq:cc` 597, 611
- `\cs_set_eq:cN` 1850
- `\cs_set_eq:Nc` 2552, 2553, 2554, 2555
- `\cs_set_eq:NN` 105, 185, 190, 195,
521, 539, 559, 585, 589, 748, 751, 802, 803, 910, 941,
979, 1038, 1253, 1391, 1399, 1409, 1410, 1411, 1483,
1484, 1485, 1620, 1629, 2061, 2066, 2069, 2074, 2607, 2617
- `\cs_set_nopar:cpx` 1794
- `\cs_set_nopar:Npx` 2645, 4188
- `\cs_set_protected:Npn` 713, 4270, 4336
- `\cs_set_protected_nopar:Npn` 584, 752, 1811, 3160
- `\cs_set_protected_nopar:Npx` 1841
- `\cs_to_str:N` 52, 1191, 4268
- `\cs_undefine:c` 1017, 1018, 1022, 1023, 4211
- `\cs_undefine:N` 1412
- `cs4size` 27
- `\csname` 2452, 2453,
2458, 2474, 2475, 2477, 2480, 2481, 2482, 2491, 2492,
2493, 2496, 2497, 2498, 3786, 3788, 3789, 3790, 3795, 3799
- `\CTeX` 23, 101, 3254, 3257
- ctx commands:
- `\ctex_add_cmap:cn` 452
- `\ctex_add_cmap:n` 41, 444, 449
- `\ctex_add_cmap:Nn` 454, 459
- `\ctex_add_to_selectfont:n` 72, 1839, 1860, 1939
- `\ctex_after_end_preamble:n` 33, 105, 119, 4182
- `\g_ctex_after_end_preamble_hook_tl` 4183, 4187, 4195
- `\ctex_appto_cmd:NnnTF` 121, 475, 1611, 4252
- `\ctex_assign_heading_name:nn` ... 80, 2158, 2193, 2579
- `_ctex_assign_heading_name:nnn` 80, 2195, 2196
- `\ctex_at_end:n` 34, 131, 132, 289, 350, 1834, 3340
- `\ctex_at_end_package:nn`
..... 119, 120, 478, 498, 746, 757, 1716, 1718,
2049, 2218, 2663, 2704, 2706, 2728, 2758, 2806, 3256, 4196
- `\ctex_at_end_preamble:n` 119, 355, 572, 1646, 4180
- `\ctex_auto_ignorespaces:` 44, 580, 589, 1877
- `\l_ctex_autoindent_tl`
.... 34, 128, 184, 189, 194, 1913, 1917, 1920, 1989, 1992
- `_ctex_calc_bounds:nn` 4105, 4106, 4110
- `_ctex_ccglue:` 1825, 1834
- `\l_ctex_ccglue_skip`
1812, 1816, 1819, 1821, 1829, 1963, 1974, 1983, 2019, 2027
- `\ctex_CJK_input:n` 42, 524, 539
- `\c_ctex_class_tl`
.... 82, 84, 403, 408, 413, 3016, 3022, 3024, 3025, 3027
- `_ctex_clear_fntef_color:n` ... 1738, 1745, 1749, 1762
- `\c_ctex_cmap_encoding_seq` 40, 436, 437, 443
- `\l_ctex_ctexcap_options_clist` 4165, 4166, 4175
- `\ctex_declare_math_sizes:nnnn` 99, 3118, 3152
- `_ctex_def_heading_keys:n` 79, 2152, 2213
- `\ctex_default_pt:n` 33, 109, 113
- `_ctex_default_pt:w` 115, 120
- `\ctex_detected_platform:` ... 68, 1657, 1675, 3281, 3290
- `\l_ctex_encoding_tl` 33, 124,
125, 217, 218, 485, 488, 567, 575, 1712, 1902, 2044, 3053
- `\g_ctex_end_preamble_hook_tl` 4181, 4185, 4194
- `\c_ctex_engine_file_str` 31, 48, 1906
- `\ctex_family_cmap:nn` 40, 439, 447
- `\ctex_family_cmap:xn` 448
- `\l_ctex_family_default_init_tl`
..... 67, 1626, 1647, 1649, 1654
- `_ctex_family_default_wrap:n` 1629, 1648, 1651
- `\ctex_file_input:n` 32, 75, 391, 642, 1906, 3027,
3054, 3055, 3162, 3164, 3282, 3286, 3294, 3340, 3439, 3440
- `\ctex_file_wrapper:nnn` 32, 60, 76, 79, 526
- `_ctex_fix_varioref_label:n` 92
- `\ctex_fix_varioref_label:n` 2790, 2792
- `\g_ctex_font_size_flag`
... 34, 148, 157, 158, 159, 371, 375, 397, 3057, 3058, 3161
- `\c_ctex_font_size_prop` 98, 3073, 3084, 3090, 3128, 3155
- `\c_ctex_font_size_seq` 3082, 3085, 3096
- `\l_ctex_font_size_tl` 3073, 3074, 3120, 3121, 3155, 3156
- `\ctex_fontset_error:n`
.... 102, 3273, 3526, 3560, 3581, 3626, 3645, 3687, 3729
- `\g_ctex_fontset_tl` ... 221, 1659, 1687, 1694, 1695,
3279, 3282, 3285, 3286, 3292, 3294, 3314, 3316, 3321, 3338
- `\ctex_gadd_hook:cn` 120, 4200
- `\ctex_gadd_hook:Nn` 120, 590, 4202, 4207
- `_ctex_get_font_sizes:Nn` 99, 3120, 3123
- `\ctex_get_macro_meaning:N` 4339
- `\ctex_get_macro_meaning:NTF` 123, 4362, 4418
- `_ctex_get_macro_meaning:w` 123, 4341, 4344
- `\l_ctex_heading_bool` 300, 3012
- `\c_ctex_headings_seq`
..... 79, 2118, 2119, 2120, 2121, 2210, 2789
- `_ctex_hookto_cmd:Nnnnw` 124, 4249, 4254, 4415
- `_ctex_hookto_cmd_parameter:Nnnnw` .. 125, 4422, 4441
- `_ctex_hookto_cmd_parameterless:Nnnw` 124, 4421, 4426
- `\ctex_hypersetup:n`
..... 69, 69, 353, 1702, 1706, 1711, 1714, 1729, 1732
- `\ctex_if_autoindent_touched:F`
..... 34, 129, 185, 190, 195, 3061
- `\ctex_if_ccglue_touched:` 1822
- `\ctex_if_ccglue_touched:TF` 72, 1958
- `\ctex_if_ccglue_touched_p:` 72
- `\ctex_if_macosx:TF` 68, 1682, 1683, 1685, 1691
- `\ctex_if_preamble:TF` 33, 104, 105, 3309
- `\ctex_if_rescanable:NnTF` 123, 4360, 4394
- `\ctex_ignorespaces_case:N` ... 44, 582, 1874, 1877, 1880
- `_ctex_initial_heading:n` 79, 2122, 2212
- `_ctex_left_hook_aux:nn` 4439
- `\l_ctex_line_spread_fp` ... 101, 177, 3059, 3247, 3249
- `\l_ctex_line_stretch_tl` 1945, 1950, 1951, 1955

`\ctex_load_fontset:` [102](#), [3277](#), [3298](#), [3317](#), [3339](#)
`\ctex_load_zhmap:n` [501](#)
`\ctex_load_zhmap:nmmn`
 [42](#), [492](#), [3482](#), [3554](#), [3620](#), [3681](#), [3723](#)
`\l_ctex_ltj_add_alternate_bool` [1141](#), [1153](#), [1171](#), [1178](#)
`\ctex_ltj_add_font_features:n` [57](#), [1161](#), [1179](#)
`\ctex_ltj_add_font_features:nn` [57](#), [1163](#), [1180](#)
`\ctex_ltj_add_font_features:x` [1203](#)
`\ctex_ltj_add_font_features:xn` [1162](#)
`__ctex_ltj_alternate_cs:n`
 [1018](#), [1021](#), [1022](#), [1023](#), [1056](#), [1057](#), [1059](#), [1144](#),
 [1147](#), [1148](#), [1150](#), [1151](#), [1341](#), [1378](#), [1440](#), [1465](#), [1505](#), [1522](#)
`\l_ctex_ltj_alternate_family_tl` [1403](#), [1406](#)
`\l_ctex_ltj_alternate_options_clist`
 [1356](#), [1357](#), [1368](#), [1372](#)
`\l_ctex_ltj_alternate_seq`
 [986](#), [1325](#), [1326](#), [1334](#), [1340](#), [1348](#), [1596](#)
`\l_ctex_ltj_base_CJKfamily_tl`
 [987](#), [1003](#), [1054](#), [1440](#), [1465](#)
`\l_ctex_ltj_base_family_tl` [63](#),
 [1019](#), [1049](#), [1055](#), [1384](#), [1425](#), [1484](#), [1503](#), [1508](#), [1509](#), [1521](#)
`__ctex_ltj_change_encoding:` [53](#), [959](#), [1037](#), [1398](#)
`\l_ctex_ltj_char_range_clist` [985](#),
 [1309](#), [1318](#), [1337](#), [1355](#), [1367](#), [1379](#), [1402](#), [1424](#), [1575](#), [1577](#)
`\ctex_ltj_char_range_key:nn` [65](#), [1569](#), [1572](#)
`__ctex_ltj_char_range_parse_feature:w`
 [65](#), [1580](#), [1584](#)
`\g_ctex_ltj_char_range_prop` ... [64](#), [1432](#), [1543](#), [1548](#)
`\l_ctex_ltj_char_range_tl`
 [1432](#), [1435](#), [1541](#), [1543](#), [1545](#), [1547](#), [1554](#)
`__ctex_ltj_check_family:n` [54](#), [1013](#), [1339](#)
`\ctex_ltj_clear_alternate_font:n` [1494](#), [1500](#)
`\l_ctex_ltj_current_font_tl`
 [49](#), [770](#), [774](#), [775](#), [787](#), [829](#), [834](#)
`\l_ctex_ltj_current_shape_tl` [852](#), [857](#), [862](#), [863](#)
`\ctex_ltj_declare_alternate_shape:nmmmm`
 [62](#), [1401](#), [1421](#), [1423](#)
`\ctex_ltj_declare_char_range:n` [1534](#), [1536](#)
`__ctex_ltj_declare_char_range:nn` [1535](#), [1537](#)
`\ctex_ltj_declare_char_range:nn` [64](#), [1538](#), [1539](#)
`\ctex_ltj_declare_char_range:x` [1531](#)
`\ctex_ltj_def_char_range_key:n` [65](#), [1544](#), [1564](#)
`\g_ctex_ltj_default_features_clist` [1007](#), [1226](#), [1227](#)
`\ctex_ltj_ensure_default_family:` ... [59](#), [1255](#), [1643](#)
`\ctex_ltj_extract_font:` [49](#), [802](#), [809](#)
`__ctex_ltj_family_csname:n` ... [1017](#), [1034](#), [1051](#), [1078](#)
`\g_ctex_ltj_family_font_name_prop`
 [54](#), [992](#), [1011](#), [1015](#), [1089](#), [1165](#), [1257](#), [1269](#)
`\g_ctex_ltj_family_font_options_prop`
 [54](#), [993](#), [1012](#), [1168](#)
`\ctex_ltj_family_if_exist:nTF` [55](#)
`\ctex_ltj_family_if_exist:xN` [1073](#)
`\ctex_ltj_family_if_exist:xNF` [1259](#)
`\ctex_ltj_family_if_exist:xNT` [1285](#)
`\ctex_ltj_family_if_exist:xNTF` [1064](#), [1264](#), [1282](#), [1503](#)
`\g_ctex_ltj_family_int` . [1131](#), [1133](#), [1417](#), [1418](#), [1420](#)
`\g_ctex_ltj_family_name_prop`
 [54](#), [1010](#), [1019](#), [1041](#), [1075](#), [1384](#)
`\l_ctex_ltj_family_tl`
 [1066](#), [1071](#), [1128](#), [1129](#), [1144](#), [1148](#), [1151](#), [1162](#), [1497](#), [1498](#)
`__ctex_ltj_family_unknown_warning:n`
 [56](#), [1069](#), [1087](#), [1513](#)
`\l_ctex_ltj_font_options_clist` [989](#), [994](#),
 [995](#), [996](#), [1004](#), [1040](#), [1169](#), [1170](#), [1173](#), [1375](#), [1376](#), [1380](#)
`\ctex_ltj_fontspec:nn` [56](#), [1125](#), [1138](#)
`__ctex_ltj_fontspec:nnn` [1139](#), [1159](#)
`__ctex_ltj_fontspec:xnn` [1132](#)
`\ctex_ltj_fontspec:xx` [1137](#), [1172](#), [1198](#)
`\g_ctex_ltj_fontspec_family_tl`
 [1039](#), [1042](#), [1043](#), [1044](#), [1050](#), [1055](#), [1080](#)
`\g_ctex_ltj_fontspec_prop` [1127](#), [1155](#), [1160](#)
`\ctex_ltj_get_and_define_fonts:nN` [52](#), [902](#), [910](#)
`\ctex_ltj_get_and_define_fonts_al:nN` [906](#), [909](#)
`\ctex_ltj_get_and_define_fonts_ja:nN` [905](#), [911](#)
`__ctex_ltj_gset_family_cs:nn` [55](#), [997](#), [1032](#)
`\ctex_ltj_if_alternate_shape_exist:n` [843](#)
`\ctex_ltj_if_alternate_shape_exist:nF` [825](#)
`\ctex_ltj_if_alternate_shape_exist:nT` [812](#), [852](#)
`\ctex_ltj_if_alternate_shape_exist:nTF` [50](#)
`\ctex_ltj_if_jfont:nTF` [51](#), [53](#), [888](#), [899](#), [904](#)
`\ctex_ltj_if_jfont_math:NTF` [52](#), [894](#), [935](#)
`__ctex_ltj_if_jfont_math:w` [895](#), [898](#)
`\l_ctex_ltj_jfm_tl` ... [53](#), [945](#), [946](#), [948](#), [963](#), [965](#), [1008](#)
`\c_ctex_ltj_math_fam_int` [971](#), [972](#)
`\c_ctex_ltj_math_family_tl`
 [1291](#), [1293](#), [1295](#), [1298](#), [1302](#)
`\ctex_ltj_math_group_hook:` [52](#), [752](#), [934](#), [940](#)
`\c_ctex_ltj_math_tl`
 [966](#), [967](#), [969](#), [971](#), [1224](#), [1282](#), [1292](#), [1297](#), [1301](#)
`__ctex_ltj_msg_def_family_map:n` [1102](#), [1104](#)
`__ctex_ltj_msg_family_map:n` [1031](#), [1101](#), [1115](#)
`__ctex_ltj_noembed_wrap:n` [979](#), [981](#)
`__ctex_ltj_patch_external_font:w` [50](#), [815](#), [848](#)
`\ctex_ltj_pickup_font:`
 [49](#), [783](#), [797](#), [808](#), [872](#), [914](#), [916](#), [918](#)
`__ctex_ltj_pop_fontname:` [786](#), [795](#), [874](#)
`__ctex_ltj_push_fontname:n` [782](#), [790](#), [871](#)
`\ctex_ltj_reset_alternate_font:n` [1496](#), [1516](#)
`\g_ctex_ltj_reset_alternate_prop`
 [1024](#), [1507](#), [1519](#), [1527](#)
`\ctex_ltj_save_alternate_family:Nmmn`
 [61](#), [1366](#), [1386](#), [1389](#)
`\ctex_ltj_save_alternate_seq:cn` [1341](#)
`\ctex_ltj_save_alternate_seq:Nn` [60](#), [1346](#), [1351](#)
`\ctex_ltj_save_alternate_seq:Nmmwn` [60](#)
`\ctex_ltj_save_alternate_seq:Nmmwnw` [1349](#), [1352](#)
`__ctex_ltj_save_alternate_shape:cnn` ... [1439](#), [1464](#)
`__ctex_ltj_save_alternate_shape:Nnn` [63](#), [1480](#), [1490](#)
`\ctex_ltj_save_char_range:n` [65](#), [1542](#), [1549](#)
`\ctex_ltj_save_char_range:nn` [1551](#), [1552](#)
`__ctex_ltj_save_fontname:w` [792](#), [796](#)
`\ctex_ltj_select_alternate_font:` [51](#), [850](#), [1863](#)

<code>\ctex_ltj_select_font:</code>	48 , 768 , 1862	<code>\ctex_patch_cmd:Nnn</code> 120 , 2674 , 2680 , 2686 , 2739 , 2743 ,
<code>__ctex_ltj_select_font_aux:</code>	48 , 771 , 777 , 788	2749 , 2753 , 2761 , 2764 , 2769 , 2773 , 3268 , 3269 , 3270 , 4232
<code>\ctex_ltj_set_alternate_family:coonn</code>	1377	<code>__ctex_patch_cmd:Nnnnw</code> . 124 , 4224 , 4229 , 4234 , 4391
<code>__ctex_ltj_set_alternate_family:n</code>	1045 , 1052	<code>\ctex_patch_cmd_all:NnnnTF</code>
<code>\ctex_ltj_set_alternate_family:nn</code> ..	61 , 1344 , 1373	120 , 4227
<code>\ctex_ltj_set_alternate_family:nnF</code> ...	60 , 990 , 1335	<code>\ctex_patch_cmd_once:NnnnTF</code>
<code>\ctex_ltj_set_alternate_family:nnn</code>	120 , 4222
.....	61 , 1385 , 1393 , 1395	<code>\ctex_patch_failure:N</code>
<code>\ctex_ltj_set_alternate_family:Nnnnn</code> ...	1382 , 1388	481 , 510 , 514 , 518 , 522 ,
<code>\ctex_ltj_set_alternate_seq:n</code>	60 , 1307 , 1316	1242 , 1246 , 1250 , 1254 , 1615 , 1852 , 2724 , 2734 , 4241 , 4243
<code>\ctex_ltj_set_alternate_shape:n</code> ...	1449 , 1461 , 1472	<code>__ctex_patch_false:w</code> 4260 , 4266 , 4406 , 4408 , 4424 , 4450
<code>\ctex_ltj_set_alternate_shape:nnN</code> ..	63 , 1434 , 1468	<code>__ctex_patch_rebuild:Nn</code>
<code>\ctex_ltj_set_alternate_shape:nnn</code> ..	63 , 1437 , 1444
<code>\ctex_ltj_set_alternate_shape:nnnn</code>	1446 , 1447
<code>\ctex_ltj_set_alternate_shape:Nnnnnn</code> 62 , 1424 , 1428	123 , 4364 , 4370 , 4403 , 4443 , 4447
<code>\ctex_ltj_set_family:nnn</code>	54 , 982 , 1002 , 1156	<code>__ctex_patch_replace:nnnTF</code>
<code>\ctex_ltj_set_family:xxx</code>	124 , 4398 , 4410
.....	1001 , 1187 , 1194 , 1209 , 1215 , 1220 , 1224	<code>__ctex_patch_rescan:NNn</code>
<code>\ctex_ltj_subst_font:</code>	50 , 803 , 823	4372 , 4373 , 4374 , 4382
<code>\ctex_ltj_swap_cs:NN</code>	62 , 1400 , 1407	<code>__ctex_patch_true:w</code>
<code>\ctex_ltj_switch_family:n</code>	1062 , 1072 , 1157 , 1193	4259 , 4265 , 4404 , 4436 , 4448
<code>\ctex_ltj_switch_family:x</code>	1061 , 1129	<code>\ctex_plane_to_utfxvibe:Nn</code>
<code>__ctex_ltj_tmp:w</code>	1314 , 1331 , 1409 , 1411 , 1412	43 , 541 , 559
<code>\l__ctex_ltj_tmp_seq</code>	1322 , 1323 , 1326 , 1333	<code>\l__ctex_prefix_str</code> ..
<code>\l__ctex_ltj_tmp_tl</code>	988 , 989 , 1015 , 1026 , 1029 ,	123 , 4332 , 4351 , 4372 , 4428 , 4429
1166 , 1174 , 1190 , 1193 , 1194 , 1282 , 1283 , 1285 , 1286 ,	1320 , 1321 , 1322 , 1359 , 1361 , 1362 , 1369 , 1579 , 1587 , 1596	<code>\l__ctex_prefix_tl</code>
1320 , 1321 , 1322 , 1359 , 1361 , 1362 , 1369 , 1579 , 1587 , 1596	4372 , 4377 , 4388
<code>\l__ctex_ltj_tmpa_tl</code>	1259 , 1264	<code>\ctex_preto_cmd:NnnTF</code>
<code>__ctex_ltj_um_char:Nn</code>	721 , 726	121 , 507 , 511 ,
<code>__ctex_ltj_um_char_aux:Nn</code>	731 , 745	515 , 519 , 1239 , 1243 , 1247 , 1251 , 1846 , 2721 , 2730 , 4247
<code>__ctex_ltj_um_char_aux:Nx</code>	728	<code>\l__ctex_punct_bounds_clist</code> ...
<code>__ctex_ltj_um_define_math_chars:</code>	710 , 748	4091 , 4097 , 4103 , 4109
<code>\g__ctex_ltj_unknown_family_seq</code> ...	1091 , 1093 , 1098	<code>\c__ctex_punct_family_clist</code>
<code>__ctex_ltj_update_family_uid:N</code> .	62 , 995 , 1375 , 1415	593 , 603
<code>\ctex_ltj_update_mathfont:</code>	59 , 1277 , 1280	<code>\l__ctex_punct_font</code>
<code>\ctex_ltj_update_mathfont:n</code>	1283 , 1286 , 1289	4089 , 4090
<code>\ctex_ltj_update_xkanjiskip:</code>	66 , 1598 , 1864	<code>\ctex_punct_map_bfseries:nn</code>
<code>__ctex_ltj_use_global_options:N</code> 996 , 1005 , 1357 , 1376
<code>\ctex_ltj_use_math_group:Nn</code>	52 , 751 , 928 , 941
<code>\l__ctex_ltj_xkanjiskip_skip</code>	45 , 613 , 3467 , 3479 , 3485 , 3551 ,
.....	66 , 1601 , 1603 , 1604 , 1607 , 1608	3557 , 3616 , 3617 , 3623 , 3676 , 3677 , 3678 , 3684 , 3720 , 3726
<code>\c__ctex_macosx_file_str</code>	1693 , 1697	<code>\ctex_punct_map_family:nn</code> ...
<code>\ctex_make_spa:nn</code>	117 , 4077 , 4085	45 , 608 , 3452 , 3466 ,
<code>\ctex_mono_jfm:n</code>	53 , 942 , 949	3476 , 3477 , 3484 , 3547 , 3548 , 3549 , 3556 , 3612 , 3613 ,
<code>\ctex_mono_jfm:o</code>	950 , 1897	3614 , 3622 , 3672 , 3673 , 3674 , 3683 , 3716 , 3717 , 3718 , 3725
<code>\ctex_package_end_hook:n</code>	120 , 4208 , 4213	<code>\ctex_punct_map_itshape:nn</code>
<code>\ctex_package_end_hook:o</code>	120 , 4217	45 , 630 ,
<code>\l__ctex_parameter_str</code> ...	123 , 4333 , 4352 , 4373 , 4420	3478 , 3486 , 3550 , 3558 , 3615 , 3624 , 3675 , 3685 , 3719 , 3727
<code>\l__ctex_parameter_tl</code>	4373 , 4378 , 4389	<code>\ctex_punct_map_series:nnn</code>
<code>\ctex_parse_name:NN</code>	121 , 4263 , 4267	617 , 618 , 621
<code>\ctex_parse_name:NNn</code>	4272 , 4331	<code>\c__ctex_punct_seq</code>
<code>__ctex_parse_name:nNNnN</code>	4284 , 4293	4068 , 4069 , 4092
<code>\ctex_parse_name:NNx</code>	4268	<code>\ctex_punct_set:n</code>
<code>__ctex_parse_name:w</code>	4306 , 4312 , 4321	44 , 591 ,
<code>\ctex_patch_boot:NnnnTF</code>	3447 , 3483 , 3546 , 3555 , 3611 , 3621 , 3671 , 3682 , 3715 , 3724
.....	121 , 4224 , 4229 , 4234 , 4249 , 4254 , 4257	<code>\ctex_punct_space:nn</code>
		45 , 639 , 641
		<code>\l__ctex_punct_tl</code>
	
		276 , 563 , 564 , 650 , 950 , 1894 , 1895 , 1896 , 1897
		<code>__ctex_rebuild_cmd:w</code> ...
		4364 , 4365 , 4369 , 4443 , 4444
		<code>\l__ctex_replacement_str</code>
	
		123 , 4334 , 4353 , 4374 , 4412 , 4413 , 4446
		<code>\l__ctex_replacement_tl</code> .
		4347 , 4348 , 4374 , 4379 , 4390
		<code>__ctex_right_hook_aux:nn</code>
		4440
		<code>__ctex_save_bounds:n</code>
		4093 , 4101
		<code>__ctex_save_cmap:Nn</code>
		456 , 460
		<code>__ctex_save_font_size:nn</code>
		98 , 3086 , 3117
		<code>__ctex_save_selectfont:</code>
		1845 , 1850
		<code>\ctex_scheme_input:n</code>
		32 , 77 , 91
		<code>\ctex_scheme_input:o</code>
		2808 , 3029
		<code>\l__ctex_scheme_tl</code>
		309 , 2808 , 3029
		<code>\g__ctex_section_depth_flag</code>
	
		32 , 92 , 304 , 306 , 2952 , 2968 , 2984
		<code>\c__ctex_section_headings_seq</code>
	
		79 , 2115 , 2116 , 2119 , 2562 , 2621
		<code>\ctex_select_size:</code> ..
		1914 , 1921 , 1924 , 1925 , 1946 , 2000
		<code>\ctex_set_default_ccwd:Nn</code> ...
		33 , 106 , 194 , 1920 , 1945

- `\ctex_set_font_size:Nnn` 99, 3153, 3168, 3176, 3187, 3198, 3199, 3200, 3201, 3202, 3203, 3204, 3207, 3215, 3226, 3237, 3238, 3239, 3240, 3241, 3242, 3243
`_ctex_set_font_size:nnNn` 3156, 3159
`\ctex_set_ignorespaces:` 44, 584, 586, 588, 590
`\g_ctex_spa_iow` 4079, 4082, 4084, 4094
`\c_ctex_std_class_tl` 96, 3003, 3015, 3016, 3064
`\g_ctex_std_options_clist` 34, 134, 376, 388, 398, 400, 404, 409, 414
`\ctex_titleps_hook:` 90, 2672, 2705, 2706
`_ctex_titlesec_format:n` 2617, 2625, 2628
`_ctex_titlesec_format:Nn` 2607, 2631, 2633
`_ctex_titlesec_format_explicit:Nn` 2608, 2643
`\ctex_titlesec_hook:` 88, 2603, 2667
`_ctex_titlesec_hook:n` 2617, 2621, 2623
`_ctex_titlesec_spacing:Nn` 2626, 2653
`_ctex_titlesec_spacing:nnnnn` 2654, 2655
`_ctex_tmp:w` 4270, 4325, 4336, 4358
`\l_ctex_tmp_box` 32, 59, 1800, 1801, 2709, 2713
`\l_ctex_tmp_dim` 32, 58, 1955, 1956, 1961, 1964, 1970, 2009, 2011, 2013, 2022, 2024
`\l_ctex_tmp_int` 32, 57, 544, 546, 547, 549, 553, 555, 1969, 1971, 1972, 1978, 1979, 2011, 2012, 2013, 2016, 2017, 2022, 2024, 2561, 2564, 2568
`\l_ctex_tmp_tl` 32, 56, 360, 367, 382, 383, 462, 463, 543, 545, 2154, 2204, 2215, 2216, 3128, 3129
`\ctex_update_ccglue:` 71, 1808, 1965, 1984, 2028
`\ctex_update_ccwd:` 71, 1797, 1959, 2036
`\ctex_update_default_family:` 67, 1624, 1646
`\ctex_update_em_unit:` 72, 1836, 1954, 2004
`\ctex_update_parindent:` 76, 1932, 1987, 2029
`\ctex_update_size:` 74, 1926, 1927, 1940
`\ctex_update_stretch:` 75, 1931, 1952
`\ctex_update_ziju:` 76, 1934, 2002
`\c_ctex_upper_case_file_str` 1681, 1689
`\ctex_varioref_hook:` 92, 2787, 2798, 2806
`\c_ctex_version_tl` 3, 6, 7, 8, 9, 20, 23
`_ctex_write_family:nn` 4081, 4086
`\c_ctex_zero_tl` 1929, 1936, 1938
`\ctex_zhmap_case:nnn` 34, 130, 257, 262, 267, 3445, 3528, 3583, 3659, 3706
`\g_ctex_zhmCJK_bool` ... 34, 127, 256, 261, 266, 484, 566
`\ctex_zihao:n` 98, 3070, 3071
`\ctex_ziju:n` 1996, 1997
`\l_ctex_ziju_dim` 2005, 2006, 2010, 2021, 2023, 2027, 2031
`\l_ctex_ziju_tl` 1929, 1937, 1938, 1999, 2005
`\ctexbibname` 2100, 2109
`\CTEXcounter` 2048
`\CTEXdigits` 23, 77, 2059
`\CTEXfilltwosides` 28, 1770, 1780
`\CTEXindent` 28, 77, 2032, 2034
`\CTEXnoindent` 28, 77, 2038, 2040
`\CTEXnumber` 23, 77, 2057
`\CTEXoptions` 28, 40, 428, 430
`\ctexset` 5, 40, 419, 425, 431
`\CTEXsetfont` 28, 74, 1924
`\CTEXsetup` 28, 40, 422, 424
`\CTEXsout` 28
`\ctexspadef` 45, 641, 4096
`\CTEXthechapter` 15, 2368, 2371, 2751, 2771, 2772
`\CTEXtheparagraph` 15
`\CTEXthepart` 15, 2259, 2289
`\CTEXthesection` 15, 2741, 2755, 2763, 2775
`\CTEXthesubparagraph` 15
`\CTEXthesubsection` 15, 2745, 2766
`\CTEXthesubsubsection` 15
`\CTEXunderdblline` 28
`\CTEXunderdot` 28
`\CTEXunderline` 28
`\CTEXunderwave` 28
`\CTEXxout` 28
`\CurrentOption` 388
- D**
- `declarecharrange` 24, 64
`\DeclareFontEncoding` 952
`\DeclareFontFamily` 41, 475, 481, 961, 3358, 3359, 3360, 3361, 3362, 3363
`\DeclareFontShape` 962, 964, 3366, 3367, 3368, 3369, 3370, 3371, 3372, 3373, 3374, 3377, 3378, 3379, 3380, 3381, 3382, 3383, 3384, 3385, 3390, 3391, 3392, 3393, 3394, 3395, 3396, 3397, 3398, 3401, 3402, 3403, 3404, 3405, 3406, 3407, 3408, 3409, 3414, 3415, 3416, 3417, 3418, 3419, 3420, 3421, 3422, 3425, 3426, 3427, 3428, 3429, 3430, 3431, 3432, 3433
`\DeclareFontSubstitution` 955
`\DeclareMathSizes` 3121
`\DeclareSymbolFont` 967, 1292
`\def` 2254, 2284, 2311, 2325, 2335, 2362, 2391, 2413, 2428, 2451, 2454, 2487, 2504, 2517, 2518, 2522, 3181, 3192, 3220, 3231, 3760, 3796
`\defaultCJKfontfeatures` 58, 1225, 1233
dim commands:
`\dim_abs:n` 2023
`\dim_add:Nn` 2006
`\dim_compare:nNnF` 1991, 2014
`\dim_compare:nNnTF` 108, 1956, 2007
`\dim_eval:n` 1976, 2022
`\dim_max:nn` 2712
`\dim_min:nn` 2023
`\dim_new:N` 58, 1807, 2031
`\dim_set:Nn` 1801, 1837, 1838, 1955, 1961, 1992, 2005, 2009, 2036, 2710, 3251
`\dim_sub:Nn` 2013
`\dim_to_decimal:n` 33, 3092, 3093, 3130
`\dim_to_decimal_in_unit:nn` 4116
`\dim_use:N` 116, 3251
`\dim_zero:N` 2042
`\dimexpr` 2269, 2314, 2395, 2417
`\directlua` 688, 689
`\document` 4188, 4191

E

`\edef` 3779, 3780

`\else` 2229, 2238, 2246, 2260, 2264, 2290, 2294, 2351, 2358, 2372, 2376, 2379, 2387, 2439, 2444, 2457, 2461, 2479, 2486, 2495, 2516, 2530, 2543, 3791, 3792, 3838, 3877, 3908, 3967, 4030

else commands:

`\else:` 487, 846, 892, 1826, 3255

`\end` 4161

`\endCJKfilltwosides` 1776

`\endcsname` 2452, 2453, 2458, 2474, 2475, 2477, 2480, 2481, 2482, 2491, 2492, 2493, 2496, 2497, 2498, 3786, 3788, 3789, 3790, 3793, 3795, 3799

`\endCTEXfilltwosides` 1776, 1781

`\endgroup` 2476, 2515, 2538, 3779, 3787, 3802

`\endinput` 3784

`\endlinechar` 3758, 3759

etex commands:

`\etex_dimexpr:D` 116, 1970

`\etex_glueexpr:D` 1962

`\everypar` 2443, 2533, 2545

`\EverySelectfont` 1854, 1857

exp commands:

`\exp_after:wN` 115, 799, 815, 817, 892, 895, 2654, 3074, 3121, 3156, 4341

`\exp_args:cc` 2783

`\exp_args:Nc` 2626, 2631, 4295

`\exp_args:Nnc` 92

`\exp_args:NNf` 1587

`\exp_args:NNNo` 3250

`\exp_args:Nno` 2795

`\exp_args:No` 545, 1690, 2650, 4093, 4358

`\exp_args:Nx` 289, 572, 646, 1996, 3070, 3249

`\exp_last_unbraced:Nf` 4306

`\exp_not:c` 834, 1765, 1767, 1794, 2128, 2129, 2130, 2135, 2137, 2139, 2140, 2143, 2144, 2147, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2173, 2175, 2177, 2179, 2181, 2183, 2569, 2570, 2571, 2572, 4287, 4288

`\exp_not:N` 120, 121, 466, 574, 575, 576, 761, 762, 764, 870, 955, 1041, 1042, 1043, 1044, 1366, 1487, 1634, 1635, 1636, 1651, 1766, 2138, 2567, 2572, 2638, 2639, 2648, 2649, 3090, 3292, 4286, 4377

`\exp_not:n` 67, 1038, 1039, 1629, 2657, 4304, 4434

`\exp_not:o` 382, 763, 765, 1040, 1173, 1174, 1367, 1368, 1369, 1393, 1487, 1652, 1842, 2215, 2784, 4191, 4377, 4378, 4379, 4433

`\exp_not:v` 4167

`\expandafter` 3761, 3762, 3786, 3787, 3788, 3789, 3790, 3795, 3799, 3800

`\ExplSyntaxOff` 64, 70, 475, 507, 511, 515, 519, 670, 1239, 1243, 1247, 1251, 1846, 2721, 4122, 4237

`\ExplSyntaxOn` 69, 699, 4061

F

false commands:

`\c_false_bool` 2560

`\familydefault` 1632

`fancyhdr` 27, 38

`\fangsong` 3744

`\fi` 2223, 2231, 2240, 2248, 2263, 2266, 2273, 2293, 2296, 2304, 2344, 2345, 2348, 2351, 2360, 2375, 2378, 2381, 2390, 2402, 2403, 2404, 2429, 2435, 2441, 2447, 2464, 2465, 2483, 2484, 2499, 2500, 2502, 2520, 2546, 2547, 2741, 2745, 2751, 2755, 2763, 2766, 2771, 2772, 2775, 3791, 3792, 3801, 4054

fi commands:

`\fi:` 401, 523, 560, 578, 846, 892, 1826, 2746, 2756, 3165, 3255

`\figurename` 2088

`figurename` 11

file commands:

`\g_file_current_name_tl` 662, 1690

`\file_if_exist:nTF` 46, 1677, 1679, 1681, 1693, 3285, 3438, 3455

`\file_if_exist_input:nF` 84

`\file_input:n` 76, 85, 87, 497, 499, 532

five commands:

`\c_five` 3093

`fntef` 28, 38

`\font` 466, 3358, 3359, 3360, 3361, 3362, 3363

`fontset` 7, 36, 103

`\fontsize` 3074

fontspec commands:

`\g_fontspec_encoding_tl` 960

`\l_fontspec_family_tl` 1426

`_fontspec_namewrap:n` 54, 979

`\fontspec_set_family:Nnn` 1039, 1403

`\footnotesep` 101, 3251

`\footnotesize` 3187, 3226, 3250

four commands:

`\c_four` 306, 553, 555

fp commands:

`\fp_compare:nNnF` 3247

`\fp_compare:nNnT` 3059

`\fp_eval:n` 1999, 4112

`\fp_use:N` 1936, 3249

G

GBK 6, 36

`\gdef` 2591, 2592, 2593, 2594, 2597, 2598, 2599, 2600

`\global` 2354, 2532, 2535

`\globaljfont` 817

`\glueexpr` 2226, 2250, 2280, 2321, 2337, 2393, 2411, 2415, 2425, 2446, 2471, 2488, 2510, 2517, 2518, 2528, 2542

group commands:

`\group_begin:` . 470, 704, 712, 779, 801, 827, 869, 896, 984, 1036, 1312, 1397, 1482, 1502, 1518, 1628, 2134, 2142, 2670, 2716, 3250, 4088, 4185, 4261, 4269, 4280, 4335

`\group_end:` 474, 483, 708, 724, 784, 805, 840, 873, 900, 999,

1046, 1332, 1404, 1488, 1514, 1525, 1640, 2145, 2149,
2693, 2720, 3250, 4099, 4185, 4283, 4330, 4359, 4393, 4417
\group_insert_after:N 837, 838

H

hbox commands:

\hbox_set:Nn 1800, 2709
heading 9, 37
\heiti 3743
\hskip ... 2471, 2488, 2510, 2517, 2518, 2541, 2542, 2762, 2765
\hspace 2258, 2259, 2288, 2289, 2371
\Huge 2305, 2329, 2406, 2420, 2845, 2865, 3204, 3243
\huge 2274,
2302, 2316, 2400, 2828, 2844, 2848, 2863, 2870, 3203, 3242
hyperref 28, 38
\hypersetup 352, 353, 1703
\hyphenchar 3358, 3359, 3360, 3361, 3362, 3363

I

if commands:

\if_bool:N 484, 566
\if_case:w 397, 3161
\if_cs_exist:N 540
\if_false: 103
\if_meaning:w 1825
\if_true: 102
\ifctexpdf 33, 102, 103
\ifdim 2433, 2467, 2506, 2524
\ifincsname 3255
\IfNoValueF 426, 432
\IfNoValueTF 1191, 1451, 1556, 1589, 1591, 2199
\ifnum 2255, 2270, 2285, 2301, 2363, 2397, 2455, 2479, 2495, 3791
\ifodd 2227, 2256, 2270, 2286, 2301,
2356, 2365, 2399, 2437, 2458, 2468, 2480, 2496, 2507,
2525, 2741, 2745, 2751, 2755, 2763, 2766, 2771, 2772, 2775
\ifx 3789, 3790, 3795
\ifzhmappdf 3807, 3874, 3905, 3936, 4003
\ignorespaces 2548
\immediate 3798
indent 27, 35
\indexname 2091
indexname 12
\input 4060, 4125
int commands:
\int_compare:nNnF 2016, 3057
\int_compare:nNnTF 546, 1972, 2952, 2968, 2984
\int_const:Nn 971
\int_div_truncate:nn 75, 553
\int_eval:n 1452, 1454, 1455, 1557, 1559, 1560
\int_from_hex:n 545
\int_gincr:N 1131, 1417
\int_incr:N 2564
\int_mod:nn 555
\int_new:N 57, 1420
\int_set:Nn 536, 544, 1969, 2011
\int_set_eq:NN 530
\int_sub:Nn 549, 2017

\int_to_Hex:n 547, 552, 554
\int_use:N 467, 536, 884, 1133, 1418, 2568, 4093
\int_zero:N 2561
\interlinepenalty
2267, 2298, 2312, 2326, 2405, 2419, 2472, 2473, 2511, 2512

iow commands:

\iow_close:N 4082
\iow_indent:n 4172
\iow_new:N 4084
\iow_now:Nx 4094
\iow_open:Nn 4079
\itemsep 3184, 3195, 3223, 3234

J

\jfam 936, 972

K

\kaishu 3745
kernel commands:
\l__kernel_expl_bool 68
keys commands:
\l_keys_choice_tl 290
\keys_define:nn 151, 382, 385,
976, 1305, 1414, 1491, 1529, 1568, 1869, 1890, 1907,
1941, 2062, 2083, 2103, 2106, 2109, 2215, 2575, 3258, 3305
\keys_if_exist:nnF 1566
\keys_if_exist:nnT 3044, 3047
\l_keys_key_tl 137, 139, 141
\keys_set:nn 26, 167, 173, 229, 238, 247,
273, 283, 290, 297, 325, 331, 419, 426, 432, 2811, 2853,
2877, 2894, 2910, 2926, 2939, 2954, 2962, 2970, 2978,
2985, 2986, 2988, 3032, 3046, 3048, 3056, 3060, 3062, 3347
\keys_set_known:nnN 988

L

\labelformat 2795
\LARGE 3202, 3241
\Large 2271, 2826, 2832, 2886, 2890, 3201, 3240
\large 2897, 3200, 3239
\lastbox 2536
\leavevmode 2223, 2429
left commands:
\c_left_brace_str 4327
\leftmargin 3181, 3192, 3220, 3231
\leftmargini 3181, 3192, 3220, 3231
\let 2456, 2463, 3174, 3213, 3782, 3788
\linespread 3249
linespread 10, 35
linestretch 10, 75
\linewidth 1970, 1978, 2010
\lishu 3747
\listfigurename 2086
listfigurename 11
\listtablename 2087
listtablename 11
\LoadClass 405, 410, 415
\long 3796

`\lstlistingname` 762, 763
`\lstlistlistingname` 764, 765
`\ltjdefcharrange` 671, 672, 673, 675, 679, 680, 683, 686
`LTJFONTUID` 62
`\ltjgetparameter` 1601, 1805, 1829
`\ltjsetkanjiskip` 1819
`\ltjsetmathletter` 729
`\ltjsetparameter` 687, 693, 1618
`\ltjsetxkanjiskip` 1604
lua commands:
`\lua_escape_x:n`
 .. 845, 857, 884, 891, 1442, 1457, 1458, 1475, 1476, 1485
`\lua_now_x:n` 705,
 739, 819, 845, 849, 854, 859, 881, 890, 958, 1463, 1487, 1661

M
`\MAKESPA` 4085, 4127
mark commands:
`\q_mark` 4342, 4345
`\markboth` 2276, 2297
`\mathalpha` 717
`\mathbin` 717
`\mathfence` 718
`\mathgroup` 936
`\mathop` 718
`\mathord` 717
`\mathpunct` 718
`\mathrel` 717
max commands:
`\c_max_dim` 1956
`\mdefault` 956, 962, 968, 1293, 1298
minus commands:
`\c_minus_one` 148, 530, 3057
mode commands:
`\mode_if_math:T` 930
`\mode_if_math:TF` 735
msg commands:
`\msg_critical:nnn` 3274
`\msg_critical:nxx` 51, 662
`\msg_error:nn` 3325
`\msg_error:nnn` .. 19, 22, 38, 41, 653, 666, 669, 3075, 3157
`\msg_error:nxx` 2078
`\msg_error:nxxx` 3291, 3320
`\msg_fatal:nn` 4066
`\msg_new:nn` 4064
`\msg_new:nnn` 136, 138, 140, 142, 144, 656,
 1030, 1099, 1181, 1754, 2097, 3006, 3275, 3334, 4168, 4245
`\msg_new:nnnn` 12, 30, 43, 95, 2080, 3077, 3299, 3328
`\msg_warning:nn` 1176, 1756, 1887, 2108
`\msg_warning:nnn` 165, 171, 199, 209, 224, 233,
 242, 271, 281, 295, 323, 329, 337, 342, 348, 1094, 3262, 3312
`\msg_warning:nnnn` 424, 430, 1765, 1772, 2034, 2040
`\msg_warning:nxx` 3025, 4174, 4244
`\msg_warning:nxxx` 1026

N
nan commands:
`\c_nan_fp` 178
`\newCJKfontfamily` 57, 1188
`\NewDocumentCommand` 419, 422, 428, 1060, 1186,
 1188, 1196, 1201, 1207, 1213, 1218, 1223, 1225, 1352,
 1444, 1549, 1584, 1924, 1995, 2032, 2038, 2057, 2059,
 2193, 3069, 3254, 3742, 3743, 3744, 3745, 3747, 3748, 3750
`\newfontfeature` 973, 974, 975
`\newpage` 2338, 2343
no commands:
`\q_no_value` 4342
`\nobreak` 2271,
 2278, 2319, 2400, 2406, 2409, 2420, 2423, 2526, 2827, 2864
`\nobreakspace` 2271, 2302
`nocap` 27
`\noexpand` 3780, 3782, 3783, 3784
`nofonts` 27
`noindent` 27
`nopunct` 27
`\normalem` 1737
`\normalfont` 519, 521, 522, 1210,
 1216, 1221, 1251, 1253, 1254, 1276, 2268, 2269, 2299,
 2300, 2313, 2314, 2327, 2328, 2394, 2395, 2416, 2417, 2572
`\normalsize` 2913, 2929, 2942, 3168, 3207, 3245, 3253
`nospace` 27
`nozhmap` 27
`\null` 2249, 2250, 2341
`\numberline` 91,
 2370, 2371, 2481, 2482, 2497, 2498, 2721, 2724, 2730, 2734

O
one commands:
`\c_one` 158, 1445, 1550, 2024, 2194, 4105
`\onecolumn` 2244
or commands:
`\or:` 399, 3163

P
`\pagestyle` 3052
`\par` 2224, 2271, 2302, 2305, 2316, 2329,
 2400, 2406, 2420, 2430, 2526, 2813, 2827, 2838, 2856, 2864
`paragraph/afterskip` 29
`paragraph/beforeskip` 29
`paragraph/numbering` 14
`paragraph/runin` 18
`\parindent` 1991, 1992, 2014, 2035, 2036, 2041,
 2042, 2268, 2269, 2313, 2314, 2394, 2395, 2416, 2417, 2986
`\parsep` 3183, 3184, 3194, 3195, 3222, 3223, 3233, 3234
`\part` 81, 2222, 2235
`part/beforeskip` 29
`part/numbering` 14
`part/pagestyle` 20, 80
`\partname` 2271, 2302, 2815
`\PassOptionsToClass` 404, 409, 414
`\PassOptionsToPackage` 485, 1707, 2044, 4176
`\pdffontattr` 466

\pdflastobj 467
 \pdfmapline 3808,
 3809, 3810, 3811, 3812, 3813, 3814, 3816, 3817, 3818,
 3819, 3820, 3821, 3822, 3824, 3825, 3826, 3827, 3828,
 3829, 3831, 3832, 3833, 3834, 3835, 3836, 3937, 3938,
 3939, 3940, 3941, 3942, 3943, 3945, 3946, 3947, 3948,
 3949, 3950, 3951, 3953, 3954, 3955, 3956, 3957, 3958,
 3960, 3961, 3962, 3963, 3964, 3965, 4004, 4005, 4006,
 4007, 4008, 4009, 4011, 4012, 4013, 4014, 4015, 4016,
 4018, 4019, 4020, 4021, 4022, 4024, 4025, 4026, 4027, 4028
 \pdfobj 463
 \pdfoutput 3791
 \pdfstringdefDisableCommands 3257
 \phantomsection 2219
 prg commands:
 \prg_do_nothing:
 940, 1391, 1483, 1781, 1844, 1874, 2558, 4369
 \prg_new_conditional:Npnn 843, 1822
 \prg_new_protected_conditional:Npnn 1073, 4338
 \prg_return_false: 846, 1083, 1826, 1830, 4349
 \prg_return_true: ... 846, 1076, 1081, 1826, 1830, 4354
 \primitive 4161
 \ProcessKeysOptions 394
 prop commands:
 \prop_get:NnN 1168
 \prop_get:NnNT 1384
 \prop_get:NnNTF 1086, 1127, 1165, 1432, 3073, 3128, 3155
 \prop_get:NxNTF 1075
 \prop_gpop:NnNT 1015, 1019, 1519
 \prop_gput:Nnn 992, 1155, 3090
 \prop_gput:Nno 993, 1041, 1507, 1543
 \prop_gremove:Nn 1024
 \prop_if_empty:Nf 1089, 1257
 \prop_map_break:n 1271
 \prop_map_inline:Nn 1269
 \prop_new:N 1010, 1011, 1012, 1160, 1527, 1548, 3084
 \protect .. 2370, 2371, 2383, 2384, 2481, 2482, 2497, 2498,
 2675, 2677, 2681, 2683, 2687, 2689, 2697, 2702, 4299, 4300
 \ProvidesFile 3796, 3804, 3871, 3902, 3933, 4000
 punct 9, 27, 37, 73
 \punctstyle 564, 1895

Q

\quad . 2452, 2740, 2741, 2744, 2745, 2751, 2755, 2763, 2766,
 2771, 2772, 2775, 2833, 2871, 2880, 2898, 2914, 2930, 2943
 quark commands:
 \quark_if_no_value:Ntf 4348
 \quotation 102, 3270

R

\raggedright 2268, 2313, 2394, 2416, 2825, 2862
 \refname 2093, 2099, 2105, 2106
 \refstepcounter 92, 2257, 2287, 2366, 2459, 2800, 2801
 \relax 2226, 2250, 2269, 2280, 2285, 2301, 2314, 2321, 2337,
 2393, 2395, 2411, 2415, 2417, 2425, 2431, 2446, 2460,
 2466, 2471, 2488, 2505, 2510, 2517, 2518, 2523, 2528,
 2540, 2542, 2762, 2765, 3754, 3758, 3761, 3789, 3790, 3795

\renewcommand 2222, 2235, 2351
 \RenewDocumentCommand 2801
 \RequirePackage 27,
 54, 55, 344, 355, 486, 489, 490, 491, 645, 664, 667,
 701, 709, 761, 1716, 1725, 1736, 1742, 2045, 3065, 4221
 \RequirePackageWithOptions 4177
 resetalternatfont 25, 64
 reverse commands:
 \reverse_if:N 566
 \rmdefault 1634
 \rmfamily 507, 510, 1239, 1242

S

scan commands:
 \scan_stop:
 ... 63, 116, 817, 936, 1484, 1485, 1962, 1970, 4089, 4187
 scheme 9, 38
 \scriptsize 3198, 3237
 \secdef 2232, 2251, 2361
 section/afterskip 29
 section/beforeskip 29
 section/numbering 14
 section/runin 18
 \selectfont .. 1067, 1510, 1523, 1846, 1852, 1866, 1926, 3074
 seq commands:
 \seq_clear:N 986
 \seq_concat:NNN 1325
 \seq_gput_left:Nn 2120, 2121
 \seq_gput_right:Nn 1093, 3096
 \seq_gset_eq:NN 2119
 \seq_gset_from_clist:Nn 2116, 4069
 \seq_if_empty:Nf 1340
 \seq_if_in:NnF 1091
 \seq_if_in:NnT 443
 \seq_map_function:NN 2621
 \seq_map_inline:Nn 1348, 2210, 2562, 2789, 4092
 \seq_new:N .. 436, 1098, 1333, 1334, 2115, 2118, 3085, 4068
 \seq_put_right:No 1596
 \seq_set_filter:NNn 1323
 \seq_set_from_clist:Nn 437
 \seq_set_split:NnV 1322
 \seq_use:Nnnn 3082
 \setbox 2536
 \setCJKfamilyfont 57, 1112, 1186,
 3458, 3463, 3470, 3471, 3472, 3473, 3474, 3475, 3500,
 3501, 3509, 3510, 3512, 3513, 3514, 3515, 3517, 3538,
 3540, 3542, 3544, 3571, 3572, 3573, 3574, 3597, 3602,
 3607, 3609, 3635, 3637, 3639, 3640, 3650, 3651, 3652,
 3653, 3665, 3666, 3667, 3668, 3669, 3670, 3695, 3696,
 3697, 3698, 3699, 3700, 3712, 3713, 3714, 3736, 3737, 3738
 \setCJKmainfont .. 58, 1108, 1207, 1212, 1228, 3448, 3496,
 3504, 3530, 3564, 3585, 3630, 3647, 3661, 3690, 3708, 3732
 \setCJKmathfont 58, 1223, 1231
 \setCJKmonofont 58, 1110, 1218, 1230, 3469, 3499,
 3508, 3537, 3570, 3596, 3634, 3649, 3664, 3694, 3711, 3735
 \setCJKromanfont 1212, 1232

- \setCJKsansfont
..... 58, 1109, 1213, 1229, 3451, 3457, 3462, 3498,
3506, 3536, 3569, 3591, 3633, 3648, 3663, 3692, 3710, 3734
 - \SetSymbolFont 969, 1297, 1301
 - \sfdefault 1635
 - \sffamily 511, 514, 1243, 1246
 - six commands:
 \c_six 3093
 - skip commands:
 \skip_horizontal:N 1812, 1816
 - \skip_if_eq:nnT 1600
 - \skip_if_eq:nnTF 1829
 - \skip_new:N 1607, 1821
 - \skip_set:Nn ... 1603, 1608, 1805, 1963, 1974, 2019, 2027
 - \skip_zero:N 1983
 - \small 3176, 3215
 - \songti 3742
 - \space 2367, 2400, 2815, 2860, 2994
 - space 10, 27, 37, 73
 - space commands:
 \c_space_tl 467
 - \special 1723, 3839, 3840, 3841, 3842, 3843, 3844,
3845, 3847, 3848, 3849, 3850, 3851, 3852, 3853, 3855,
3856, 3857, 3858, 3859, 3860, 3862, 3863, 3864, 3865,
3866, 3867, 3878, 3879, 3880, 3881, 3882, 3884, 3885,
3886, 3887, 3888, 3890, 3891, 3892, 3893, 3895, 3896,
3897, 3898, 3909, 3910, 3911, 3912, 3913, 3915, 3916,
3917, 3918, 3919, 3921, 3922, 3923, 3924, 3926, 3927,
3928, 3929, 3968, 3969, 3970, 3971, 3972, 3973, 3974,
3976, 3977, 3978, 3979, 3980, 3981, 3982, 3984, 3985,
3986, 3987, 3988, 3989, 3991, 3992, 3993, 3994, 3995,
3996, 4031, 4032, 4033, 4034, 4035, 4037, 4038, 4039,
4040, 4041, 4043, 4044, 4045, 4046, 4048, 4049, 4050, 4051
 - \SplitArgument 1445, 1550, 2194
 - stop commands:
 \q_stop 116, 121, 895,
898, 1349, 1353, 1580, 1585, 4307, 4312, 4321, 4342, 4345
 - str commands:
 \str_case:nnTF 4297
 - \str_case:onF 1632
 - \str_case:onTF 4312
 - \str_case_x:nnF 1106, 1117
 - \str_const:Nn 1697
 - \str_const:Nx 48, 1689
 - \str_if_empty:NF 4428
 - \str_if_empty:NTF 4384, 4420
 - \str_if_eq:nnF 3331
 - \str_if_eq:nnTF 944
 - \str_if_eq:onF 567, 3338
 - \str_if_eq:onTF 488, 1712, 1720, 3053, 3314
 - \str_if_eq_x:nnF 563
 - \str_if_eq_x:nnTF 1261, 3311, 4303
 - \str_lower_case:n 462
 - \str_new:N 4221, 4332, 4333, 4334
 - \str_set:Nn 4351, 4352, 4353
 - \str_upper_case:n 1690
 - \strutbox 3251
 - sub3section 9, 38
 - sub4section 9, 38
 - subparagraph/afterskip 29
 - subparagraph/beforeskip 29
 - subparagraph/numbering 14
 - subparagraph/runin 18
 - subsection/afterskip 29
 - subsection/beforeskip 29
 - subsection/numbering 14
 - subsection/runin 18
 - subsubsection/afterskip 29
 - subsubsection/beforeskip 29
 - subsubsection/numbering 14
 - subsubsection/runin 18
 - sys commands:
 \c_sys_engine_str 46, 49, 51
 - \sys_if_engine_pdftex:TF
..... 126, 3443, 3523, 3578, 3644, 3657, 3704
 - \sys_if_engine_xetex:F 4062
 - \sys_if_engine_xetex:TF 3693
 - \sys_if_output_pdf:TF 101, 472, 1715, 3525, 3580
- T**
- \tablename 2089
 - tablename 12
 - \TeX 3255
 - TeX and L^AT_EX 2_ε commands:
 \@italiccorr 67, 1620
 - \@par 2472, 2511, 2881, 2899, 2915, 2956, 2972
 - \@afterheading 2281, 2322, 2389, 2529
 - \@afterindentfalse 2230, 2355, 2359, 2434, 2440
 - \@afterindenttrue 2228, 2357, 2432, 2438
 - \@Alph 2990, 2995
 - \@chapapp 2367, 2400, 2750, 2770
 - \@chapter 83, 2361, 2362
 - \@clubpenalty 2544
 - \@currentlabel 2780, 2784
 - \@currentx 4, 4167, 4216
 - \@currname 4, 4167, 4217
 - \@dblarg 2450
 - \@defaultunits 33
 - \@empty 92, 2456, 2463
 - \@endpart 83, 2308, 2332, 2335
 - \@EverySelectfont@Init 1849
 - \@hangfrom 2471, 2510
 - \@ifclassloaded 3002
 - \@ifpackagelater 37, 40, 652, 665, 668, 749, 1743
 - \@ifpackageloaded 18, 21, 661, 1700, 2660, 2725, 2797, 4198
 - \@ifpackagewith 2605, 2614, 2665
 - \@ifstar 2448
 - \@listI 3174, 3213
 - \@listi 3174, 3181, 3192, 3213, 3220, 3231
 - \@M 2267,
2298, 2312, 2326, 2405, 2419, 2472, 2473, 2511, 2512, 2537
 - \@makechapterhead 84, 2386, 2388, 2391

<code>\@makeschapterhead</code>	84, 2413
<code>\@minus</code>	2882, 2900, 2916, 2931, 2944, 3170, 3172, 3178, 3180, 3182, 3183, 3189, 3191, 3193, 3194, 3209, 3211, 3217, 3219, 3221, 3222, 3228, 3230, 3232, 3233
<code>\@namedef</code>	2675, 2677, 2681, 2683, 2687, 2689, 2697, 2702
<code>\@nameuse</code>	2675, 2677, 2687, 2689, 2697
<code>\@ne</code>	2658
<code>\@nil</code>	870
<code>\@nobreakfalse</code>	2531
<code>\@noskipsecfalse</code>	2535
<code>\@noskipsectrue</code>	2532
<code>\@onlypreamble</code>	33, 501, 1228, 1229, 1230, 1231, 1232, 1233, 3298
<code>\@part</code>	81, 2232, 2251, 2254, 2284
<code>\@pkgextension</code>	6, 7, 8, 9, 20, 23, 663, 4216
<code>\@plus</code>	2839, 2840, 2882, 2883, 2900, 2901, 2916, 2917, 2931, 2944, 2957, 2973, 3170, 3171, 3172, 3178, 3179, 3180, 3182, 3183, 3189, 3190, 3191, 3193, 3194, 3209, 3210, 3211, 3217, 3218, 3219, 3221, 3222, 3228, 3229, 3230, 3232, 3233
<code>\@popfilename</code>	120, 4214
<code>\@schapter</code>	2361
<code>\@secntformat</code>	85, 2451, 2460
<code>\@secpenalty</code>	2445, 2446
<code>\@sect</code>	85, 2450, 2454
<code>\@setfontsize</code>	3160
<code>\@spart</code>	82, 2232, 2251, 2311, 2325
<code>\@ssect</code>	86, 2449, 2504
<code>\@startsection</code>	85, 87, 88, 93, 2428, 2567
<code>\@svsec</code>	2456, 2460, 2463, 2471, 2489, 2490
<code>\@svsechd</code>	2487, 2517, 2518, 2538
<code>\@tempdima</code>	2710, 2712
<code>\@tempskipa</code>	2431, 2433, 2434, 2445, 2466, 2467, 2505, 2506, 2523, 2524, 2527, 2540, 2541
<code>\@tempwafalse</code>	2247
<code>\@tempwatrue</code>	2245
<code>\@topnewpage</code>	2386
<code>\@topnum</code>	2354
<code>\@xsect</code>	86, 2503, 2521, 2522
<code>\abstractname</code>	12
<code>\addCJKfontfeature</code>	63
<code>\AfterEndPreamble</code>	119
<code>\appendixname</code>	12, 20, 20, 20
<code>\AtBeginDocument</code>	33, 43
<code>\AtEndOfClass</code>	34
<code>\AtEndOfPackage</code>	34
<code>\AtEndOfPackageFile*</code>	119
<code>\AtEndPreamble</code>	119
<code>\baselineskip</code>	10, 11, 66, 75, 75
<code>\bfseries</code>	45
<code>\bibname</code>	12
<code>\c@chapter</code>	2995
<code>\c@secnumdepth</code>	2255, 2270, 2285, 2301, 2363, 2397, 2455, 2479, 2495
<code>\c@section</code>	2990
<code>\caption</code>	28
<code>\catcode</code>	32, 32, 42, 42, 112, 120
<code>\ccwd</code>	10, 10, 11, 21, 23, 28, 28, 33, 71, 72, 72, 74, 75, 75, 76
<code>\chapter</code>	9, 14, 20, 20, 21, 96
<code>\chaptermark</code>	12
<code>\chaptername</code>	15
<code>\char</code>	47
<code>\CheckCommand</code>	72
<code>\Chinese</code>	28, 28
<code>\chinese</code>	23, 23, 28, 28, 77, 126
<code>\CJK@ignorespaces</code>	44, 580
<code>\CJK@encoding</code>	53, 53, 55, 62, 776, 780, 864, 951, 952, 956, 958, 960, 961, 962, 964, 968, 970, 1292, 1295, 1297, 1301, 1442, 1457, 1458, 1475, 1476
<code>\CJK@envStart</code>	43, 43, 44, 574
<code>\CJK@family</code>	48, 48, 771, 773, 776, 781, 828, 864, 1064, 1509, 1519, 1521, 1866, 1867
<code>\CJK@ignorespaces</code>	44, 44, 585, 589
<code>\CJK@input</code>	42, 43, 539
<code>\CJK@loadBinding</code>	42, 43, 570
<code>\CJK@loadEncoding</code>	42
<code>\CJK@makeActive</code>	43, 569
<code>\CJK@plane</code>	43, 73, 451, 452, 462, 1854
<code>\CJK@surr</code>	43, 43, 43, 540, 559
<code>\CJK@upperReset</code>	44
<code>\CJKfamily</code>	55
<code>\CJKfamilydefault</code>	59, 67, 67, 67, 67, 67
<code>\CJKglue</code>	75, 75, 101
<code>\CJKhook</code>	44, 44
<code>\CJKrmdefault</code>	42
<code>\CJKsymbol</code>	73
<code>\CJKunderdotbasesep</code>	28
<code>\CJKunderline</code>	28
<code>\clearpage</code>	44
<code>\contentsname</code>	11, 11
<code>\CS</code>	49, 49
<code>\csname</code>	101
<code>\CTEX@afterindent</code>	2437, 2554, 2559
<code>\CTEX@aftertitle</code>	2514, 2519, 2553, 2558
<code>\CTEX@appendix@number</code>	2580, 2592, 2598
<code>\CTEX@appendix@numbering</code>	2581, 2594, 2600
<code>\CTEX@chapter@afterindent</code>	2356
<code>\CTEX@chapter@afterskip</code>	2411, 2425
<code>\CTEX@chapter@aftertitle</code>	2408, 2422
<code>\CTEX@chapter@beforeskip</code>	2393, 2415
<code>\CTEX@chapter@format</code>	2396, 2418
<code>\CTEX@chapter@indent</code>	2395, 2417
<code>\CTEX@chapter@numbering</code>	2365, 2399, 2600, 2751, 2771, 2772
<code>\CTEX@chapter@pagestyle</code>	2207, 2353
<code>\CTEX@chapter@titleformat</code>	2407, 2421
<code>\CTEX@chaptername</code>	2401
<code>\CTEX@document@left@hook</code>	4184, 4190
<code>\CTEX@document@right@hook</code>	4186, 4192
<code>\CTEX@Family@CMap</code>	448, 476, 479

<code>\CTEX@makeheadinganchor</code>	
.....	81, 2217, 2219, 2261, 2291, 2373, 2462
<code>\CTEX@part@afterindent</code>	2227
<code>\CTEX@part@aftername</code>	2272, 2303
<code>\CTEX@part@afterskip</code>	2280, 2321, 2337
<code>\CTEX@part@aftertitle</code>	2277, 2307, 2318, 2331
<code>\CTEX@part@beforeskip</code>	2226, 2250
<code>\CTEX@part@format</code>	2269, 2300, 2315, 2328
<code>\CTEX@part@indent</code>	2269, 2314
<code>\CTEX@part@numbering</code>	2256, 2270, 2286, 2301
<code>\CTEX@part@pagestyle</code>	2206, 2242
<code>\CTEX@part@titleformat</code>	2275, 2306, 2317, 2330
<code>\CTEX@partname</code>	2272, 2303
<code>\CTEX@postappendix</code>	2585, 2593, 2599
<code>\CTEX@postchapter</code>	2599
<code>\CTEX@postsection</code>	2593
<code>\CTEX@preappendix</code>	2584, 2591, 2597
<code>\CTEX@prechapter</code>	2597
<code>\CTEX@presection</code>	2591
<code>\CTEX@runin</code>	2468, 2507, 2525, 2555, 2560
<code>\CTEX@save@appendix</code>	2586, 2589
<code>\CTEX@save@refstepcounter</code>	2800, 2803
<code>\CTEX@section@numbering</code> ..	2594, 2741, 2755, 2763, 2775
<code>\CTEX@selectfont@hook</code>	
.....	72, 1841, 1842, 1844, 1847, 1854, 1857
<code>\CTEX@setcurrentlabel@n</code>	2778, 2804
<code>\CTEX@subsection@numbering</code>	2745, 2766
<code>\CTEX@thechapter</code>	2598
<code>\CTEX@thesection</code>	2592
<code>\CTEX@titleformat@n</code>	87, 2513, 2519, 2552, 2557
<code>\CTEX@titleplabel@clear</code>	90, 2684, 2699
<code>\CTEX@titleplabel@set</code>	90, 2678, 2690, 2694
<code>\CTEX@toc@width@n</code>	2707, 2722, 2732
<code>\CTEX@todayold</code>	2061, 2066
<code>\CTEX@update@sectionformat@n</code>	87, 2436, 2550
<code>\CTEX@verbatim@font@hook</code>	1613, 1617
<code>\ctex@zhmap@endinput</code>	4056
<code>\CTEXdigits</code>	23, 23
<code>\CTEXindent</code>	126
<code>\CTEXnoindent</code>	126
<code>\CTEXnumber</code>	23, 23, 23, 23, 23
<code>\CTEXoptions</code>	28, 28, 126
<code>\ctexset</code> 5, 5, 5, 5, 5, 5, 7, 11, 13, 14, 24, 25, 28, 28, 103, 126	
<code>\CTEXsetup</code>	28, 126
<code>\CTEXsout</code>	126
<code>\CTEXthechapter</code>	12, 12, 13, 15
<code>\CTEXthesection</code>	12, 12, 15, 92
<code>\CTEXunderdblline</code>	126
<code>\CTEXunderdot</code>	126
<code>\CTEXunderdotbasesep</code>	28
<code>\CTEXunderline</code>	28, 126
<code>\CTEXunderwave</code>	126
<code>\CTEXxout</code>	126
<code>\curr@fontshape</code>	782, 812, 821, 825, 871
<code>\DeclareAlternateKanjiFont</code>	62
<code>\DeclareFontEncoding</code>	52
<code>\DeclareFontFamily</code>	40, 49
<code>\DeclareFontShape</code>	49
<code>\DeclareFontShape@</code>	62, 1401
<code>\DeclareRobustCommand</code>	121
<code>\define@newfont</code>	49, 804
<code>\do@subst@correction</code>	49, 50, 803
<code>\end</code>	126
<code>\escapechar</code>	52
<code>\EverySelectfont</code>	72
<code>\external@font</code>	814, 815, 817
<code>\extract@font</code>	49, 802
<code>\f@baselineskip</code>	1962
<code>\f@encoding</code>	780, 820, 933
<code>\f@family</code>	48, 781, 820, 828
<code>\f@series</code>	776, 864
<code>\f@shape</code>	776, 864
<code>\f@size</code>	776, 782, 859, 867, 870, 871, 1801, 1837, 2713
<code>\fangsong</code>	7, 8
<code>\figurename</code>	11
<code>\font</code>	49, 49
<code>\font@name</code>	48, 49, 785, 792, 793, 796, 799, 817, 818, 835, 913, 914, 915, 916, 917, 924
<code>\fontcharwd</code>	72, 72
<code>\footnotesep</code>	101, 101, 101, 126
<code>\footnotesize</code>	101
<code>\get@external@font</code>	811
<code>\getanddefine@fonts</code>	52, 909, 910
<code>\globaljfont</code>	49
<code>\heiti</code>	7
<code>\Hy@driver</code>	1720
<code>\Hy@unicodetrue</code>	69
<code>\hypersetup</code>	126
<code>\HyPsd@ConvertToUnicode</code>	69
<code>\HyPsd@LoadUnicode</code>	70
<code>\HyPsd@pdfencoding</code>	70, 70
<code>\HyPsd@ToBigChars</code>	70
<code>\if@mainmatter</code>	2364, 2398, 2771, 3020, 3021
<code>\if@nobreak</code>	2442
<code>\if@noskipsec</code>	2223, 2429, 2534
<code>\if@openright</code>	2236, 2340, 2351
<code>\if@tempswa</code>	2346
<code>\if@twocolumn</code>	2243, 2385
<code>\if@twoside</code>	2339, 2742, 2752
<code>\iffalse</code>	52
<code>\ifin@</code>	52, 892
<code>\iftrue</code>	52
<code>\indexname</code>	12
<code>\itdefault</code>	50
<code>\itshape</code>	45
<code>\kaishu</code>	7
<code>\labelformat</code>	92, 92
<code>\leftmark</code>	12
<code>\linewidth</code>	75, 75
<code>\lishu</code>	8, 8
<code>\listfigurename</code>	11
<code>\listtablename</code>	11

<code>\lowercase</code>	126	<code>\roman</code>	23
<code>\ltj@does@alt@set</code>	50	<code>\scantokens</code>	121
<code>\ltj@getjfontnumber</code>	51, 880	<code>\scriptfont@name</code>	916, 923
<code>\ltj@set@stackfont</code>	922, 923, 924	<code>\section</code>	9, 9, 14, 14, 18, 18, 19, 19, 20, 21, 29, 79
<code>\ltj@allalchar</code>	706, 740	<code>\section*</code>	14
<code>\ltj@curjfnt</code>	49, 73	<code>\sectionmark</code>	12, 12
<code>\ltj@pickup@altfont@auxy</code>	51, 865	<code>\selectfont</code>	51, 66, 72, 72, 72, 74
<code>\ltj@pickup@altfont@copy</code>	51, 878	<code>\setCJKfamilyfont</code>	24, 24, 24
<code>\ltj@setpar@global</code>	921	<code>\setCJKmainfont</code>	24
<code>\ltj@tempcntc</code>	51, 884	<code>\sf@size</code>	915
<code>\ltjalchar</code>	47	<code>\sffamily</code>	46
<code>\ltjsetparameter</code>	66	<code>\size@update</code>	74, 75, 1926, 1940
<code>\m@ne</code>	2255, 2270, 2363, 2397, 3358, 3359, 3360, 3361, 3362, 3363	<code>\sldefault</code>	50
<code>\markboth</code>	12	<code>\songti</code>	7
<code>\markright</code>	12	<code>\special</code>	7, 69
<code>\math@bgroup</code>	932	<code>\split@name</code>	870
<code>\math@egroup</code>	937	<code>\SplitArgument</code>	80
<code>\math@fonts</code>	919	<code>\ssf@size</code>	917
<code>\mathbb</code>	101	<code>\strutbox</code>	101
<code>\maxdimen</code>	11, 75	<code>\subparagraph</code>	9, 9, 9, 9, 18, 32
<code>\meaning</code>	121, 123, 123, 123	<code>\subst@correction</code>	831, 838
<code>\newcommand</code>	121	<code>\tablename</code>	12
<code>\newrobustcmd</code>	121	<code>\textfont@name</code>	914, 922
<code>\newtitlemark</code>	90, 90	<code>\tf@size</code>	913
<code>\nfss@catcodes</code>	53	<code>\thechapter</code>	12, 15, 21
<code>\normalsize</code>	8, 8, 97	<code>\thesection</code>	12, 15, 15, 21, 92
<code>\nouppercase</code>	43, 126	<code>\titleformat</code>	88
<code>\p@</code>	1801, 1837, 2302, 2383, 2384, 2392, 2400, 2410, 2414, 2424, 2713, 2838, 2857, 2858, 2864, 3170, 3171, 3172, 3178, 3179, 3180, 3182, 3183, 3189, 3190, 3191, 3193, 3194, 3209, 3210, 3211, 3217, 3218, 3219, 3221, 3222, 3228, 3229, 3230, 3232, 3233	<code>\titlespacing</code>	88
<code>\pagestyle</code>	9	<code>\today</code>	11, 11
<code>\paragraph</code>	9, 9, 9, 9, 9, 18, 32, 32	<code>\ttfamily</code>	46
<code>\parindent</code>	10, 28, 74, 76, 101	<code>\ttl@extract</code>	88
<code>\part</code>	14, 18, 20	<code>\ttl@setsubmark</code>	90, 2680, 2686
<code>\partname</code>	15	<code>\ttl@settopmark</code>	90, 2674, 2705
<code>\PassOptionsToClass</code>	40	<code>\ttlh@hang</code>	2639, 2649
<code>\PassOptionsToPackage</code>	69	<code>\ttlh@runin</code>	2638, 2648
<code>\pdfmapline</code>	107, 109	<code>\uppercase</code>	43, 43
<code>\pdfstringdef</code>	69	<code>\use@mathgroup</code>	52, 751, 941
<code>\pickup@font</code>	48	<code>\usepackage</code>	27, 28
<code>\pickup@jfont</code>	808	<code>\verbatim@font</code>	1611, 1615
<code>\ProcessKeysOptions</code>	26	<code>\wrong@fontshape</code>	48
<code>\protected</code>	47, 90	<code>\x@protect</code>	4300
<code>\protected@edef</code>	2058, 2060, 2460, 2780	<code>\xdef</code>	90
<code>\ProvidesFile</code>	112	<code>\XeTeXglyphbounds</code>	117
<code>\ps@fancy</code>	92, 2761, 2764, 2769, 2773	<code>\yahei</code>	8, 8
<code>\ps@headings</code>	91, 2739, 2743, 2749, 2753	<code>\youyuan</code>	8, 8
<code>\refname</code>	12	<code>\z@</code>	693, 696, 2268, 2313, 2354, 2394, 2416, 2433, 2467, 2506, 2524, 2536, 2658, 3171, 3179, 3190, 3210, 3218, 3229
<code>\refstepcounter</code>	92	<code>\zhdigits</code>	23
<code>\relax</code>	43, 74	<code>\zhnum</code>	77, 126
<code>\reset@font</code>	521, 1253	<code>\zhnumber</code>	23, 23, 23, 23
<code>\rightmark</code>	12	<code>\zihao</code>	21, 21
<code>\rmfamily</code>	46	<code>\ziju</code>	21
		<code>\zw</code>	66, 66, 72
		tex commands:	
		<code>\tex_def:D</code>	4377
		<code>\tex_divide:D</code>	75, 1971, 2012

- `\tex_edef:D` 4430
 - `\tex_endlinechar:D` 530, 536
 - `\tex_font:D` 4089
 - `\tex_iftrue:D` 3021
 - `\tex_ignorespaces:D`
 - 1061, 1199, 1204, 1880, 1996, 3070, 4187
 - `\tex_immediate:D` 463
 - `\the` 3758, 3761, 3762, 3781, 3785
 - `\thechapter` 2367, 2370, 2400, 2750, 2770, 2861
 - `\theparagraph` 2928
 - `\thepart` 2258, 2271, 2288, 2302, 2816
 - `\thesection` 2740, 2754, 2762, 2774, 2879
 - `\thesubparagraph` 2941
 - `\thesubsection` 2744, 2765, 2896
 - `\thesubsubsection` 2912
 - `\thispagestyle` 2241, 2242, 2342, 2352, 2353
 - three commands:
 - `\c_three` 304, 2016, 2968, 4106
 - `\tiny` 3199, 3238
 - tl commands:
 - `\tl_clear:c` 1739, 1750, 2200
 - `\tl_clear:N`
 - .. 189, 360, 383, 1541, 1545, 1867, 1917, 2216, 3125, 4385
 - `\tl_const:cn` 640, 663
 - `\tl_const:cx` 2126, 2132
 - `\tl_const:Nn` 403, 408,
 - 413, 503, 951, 966, 1235, 1236, 1237, 1238, 3003, 3022, 3024
 - `\tl_const:Nx` 3, 1291, 1936
 - `\tl_gput_right:cn` 444
 - `\tl_gput_right:Nn` 4181, 4183, 4205
 - `\tl_gset:Nn` 1265, 1687, 1694, 1695
 - `\tl_gset:Nx` 547, 550, 913, 915, 917, 1630, 1659, 3316
 - `\tl_gset_eq:NN` 1043, 1654
 - `\tl_gset_rescan:Nnn` 1272
 - `\tl_head:N` 3129
 - `\tl_if_blank:nTF` 1358, 1454, 1455, 1559, 1560, 1574, 1592
 - `\tl_if_blank_p:n` 1324
 - `\tl_if_empty:cT` 2794
 - `\tl_if_empty:NF` 771, 1989
 - `\tl_if_empty:NTF` 2138, 3279
 - `\tl_if_eq:NNT` 1626
 - `\tl_if_eq:NNTF` 1929
 - `\tl_if_exist:NF`
 - 502, 504, 505, 506, 1234, 1236, 1237, 1238, 4204
 - `\tl_if_exist:NT` 3064
 - `\tl_if_exist:NTF` 82, 2102, 2105, 3015
 - `\tl_if_in:nnT` 715
 - `\tl_if_in:NnTF` 4412
 - `\tl_map_inline:Nn` 1470
 - `\tl_map_inline:nn` 3126
 - `\tl_new:c` 2124, 2125
 - `\tl_new:N`
 - . 56, 124, 128, 504, 505, 506, 773, 774, 862, 948, 1003,
 - 1029, 1049, 1050, 1071, 1406, 1547, 1647, 1937, 1950,
 - 2584, 2585, 4194, 4195, 4204, 4265, 4266, 4388, 4389, 4390
 - `\tl_put_left:Nn` 1849, 4214
 - `\tl_put_right:Nn`
 - 367, 1587, 2173, 2175, 2177, 2179, 2181, 2183
 - `\tl_put_right:Nx` 919, 1554, 2154, 3129, 3130
 - `\tl_replace_all:Nnn` 1321, 1362
 - `\tl_rescan:nn` 123, 4438
 - `\tl_rescan:nx` 4429
 - `\tl_set:cn` 2198, 2201
 - `\tl_set:Nn` 110, 111, 184, 217, 218,
 - 494, 495, 496, 762, 764, 775, 863, 945, 946, 987, 1054,
 - 1066, 1320, 1359, 1361, 1402, 1575, 1579, 1866, 1902,
 - 1913, 1951, 2204, 3257, 3489, 3490, 3491, 4259, 4260, 4347
 - `\tl_set:Nx`
 - .. 125, 462, 543, 814, 1190, 1649, 1894, 1999, 2635, 2654
 - `\tl_set_eq:NN`
 - 780, 781, 828, 914, 916, 960, 1055, 1080, 1509, 1521, 1938
 - `\tl_set_rescan:Nnx` 4386
 - `\tl_tail:N` 2640
 - `\tl_tail:n` 2650
 - `\tl_to_str:n` 121, 122, 4328, 4358, 4399, 4400
 - `\tl_trim_spaces:n` 1538
 - `\today` 2061, 2066, 2069, 2074
 - today 11, 77
 - token commands:
 - `\token_get_replacement_spec:N` 4285, 4307
 - `\token_to_meaning:N` 4342
 - `\token_to_str:N`
 - .. 52, 895, 904, 913, 915, 917, 1108, 1109, 1110, 1112,
 - 1119, 1120, 1121, 1183, 2099, 2100, 3078, 4096, 4172, 4244
 - `\toks` 3758, 3759, 3761, 3762, 3781, 3785
 - `\topsep` 3182, 3193, 3221, 3232
 - `\TrimSpaces` 422
 - true commands:
 - `\c_true_bool` 2559
 - `\ttdefault` 1636
 - `\ttfamily` 515, 518, 1247, 1250
 - two commands:
 - `\c_two` 92, 159, 371, 375, 2017, 2713, 2952, 2984
 - `\c_two_hundred_fifty_six` 546, 549
 - `\twocolumn` 2347
 - `\typeout` 2367, 2368
- ## U
- um commands:
 - `_um_define_math_chars:` 748
 - `_um_input_math_symbol_table:` 723
 - `_um_switchto_literal:` 753
 - `_um_sym:nnn` 713
 - `\undefined` 3783
 - `\unless` 2468, 2507, 2525
 - `\unskip` 2539
 - `\updefault` ... 956, 962, 964, 968, 970, 1293, 1295, 1298, 1302
 - use commands:
 - `\use:c` 782, 820,
 - 821, 871, 913, 915, 917, 971, 1021, 1522, 4413, 4432, 4446
 - `\use:n` 129, 1262, 1266, 1648, 1779, 2557, 2718, 3013
 - `\use:nn` 4440

