

CT_EX 套件手册

ctex.org

2015/05/15 v2.0.1*

简介

CT_EX 套件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 套件支持 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。

*[ctex-kit rev. 31d1c02](#).

目录

第 1 节 介绍	3	第 11 节 C _T _E X 套件的配置文件	25
第 2 节 简明教程	4	11.1 修改宏包默认选项	25
2.1 C _T _E X 套件的组成	4	11.2 宏包载入后的配置	26
2.2 使用 C _T _E X 文档类	4	11.3 配置标题中文翻译	26
2.3 使用 ctex 宏包	4	11.4 自定义字体集	26
第 3 节 依赖与安装	5	第 12 节 对旧版本的兼容性	26
第 4 节 宏包选项与 \ctexset 命令	6	12.1 C _T _E X 0.8a 及以前的版本	26
第 5 节 编译方式、编码与中文字库	6	12.2 C _T _E X 0.9–C _T _E X 1.0d	27
5.1 编译方式	6	12.3 C _T _E X 1.02c 以后的 SVN 开发版	28
5.2 中文编码	7	第 13 节 开发人员	29
5.3 中文字库	7	第 14 节 参考文献	29
第 6 节 排版格式设定	9	第 15 节 代码实现	30
6.1 文档默认字号	9	15.1 内部函数与变量	30
6.2 章节标题风格	10	15.2 宏包选项	36
6.3 排版方案选项	10	15.3 用户设置接口	41
第 7 节 文档汉化	12	15.4 特定引擎支持与设置	41
7.1 日期汉化	12	15.5 字距与缩进	72
7.2 文档标题汉化	12	15.6 中文数字与日期	76
7.3 页面格式设置与汉化	13	15.7 其它中文标题定义	76
第 8 节 章节标题格式设置	15	15.8 中文化的标题结构	77
第 9 节 实用命令	21	15.9 chinese 方案的其他设置	90
9.1 字号与间距	21	15.10 中文字号	90
9.2 中文数字转换	23	15.11 更新行距	93
9.3 杂项	23	15.12 其它功能	94
第 10 节 Lua _A T _E X 下的中文支持方式	24	15.13 载入中文字体	95
10.1 Lua _A T _E X 下替代字体的设置	24	15.14 宏包配置文件	96
		15.15 字体定义文件	97
		第 16 节 版本历史	112
		第 17 节 代码索引	112

第 1 节 介绍

历史

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

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

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

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

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

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

另一方面，随着 T_EX 引擎和 L^AT_EX 宏包的不断发展，L^AT_EX 的中文支持方式从早期的专用系统（如 CCT）发展为适用于不同引擎的多种方式³。这些方式的适用情况和使用方式有不少细节上的差异，同时操作系统的不同、语言环境的不同等客观情况又进一步带来了更多的细节差异。我们设计 CT_EX 套件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 套件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。

关于套件名字的说明

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 套件是由 [CT_EX 社区](#) 发起并维护的 LaTeX 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 套件并非是一事物。

ctex 则是本套件中的 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 套件，不过现在则特指 ctex.sty 这一宏包。

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 C_TE_X 套件的组成

为了适应用户不同的需求,我们将 C_TE_X 套件的主要功能分散在三个中文文档类和三个宏包当中,具体的组成见表 1。

表 1 C_TE_X 套件的组成

类别	文件	说明
文档类	ctexart.cls	标准文档类 <code>article</code> 的中文化版本,一般适用于短篇幅的文章
	ctexrep.cls	标准文档类 <code>report</code> 的中文化版本,一般适用于中篇幅的报告
	ctexbook.cls	标准文档类 <code>book</code> 的中文化版本,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,可在 <code>ctex</code> 宏包或 C _T E _X 中文文档类之外单独调用

2.2 使用 C_TE_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 C_TE_X 套件提供的三个中文文档类。

C_TE_X 套件提供了三个中文文档类: `ctexart`、`ctexrep` 和 `ctexbook`, 分别对应 L^AT_EX 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⁴。

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}
```

2.3 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
\end{itemize}
\end{frame}
\end{document}
```

⁴使用 (pdf)L^AT_EX 时也能够使用 GBK 编码,但不推荐。(见 5.2 节)

```

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

例 3

```

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

```

第3节 依赖与安装

CT_EX 是一个 L^AT_EX 宏包,只有一个源文件 ctex.dtx。使用不同的编译方式时,CT_EX 依赖的宏包略有不同。在安装 CT_EX 套件之前,请确保你的系统里已经正确安装了这些宏包。CT_EX 依赖宏包的详情叙述如下:

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 套件,它们还依赖
 - etex 宏包。
- ifpdf 宏包,属于 oberdiek 套件。
- etoolbox 宏包。
- everyisel 宏包,属于 ms 套件。
- zhnumber 宏包。
- ⇒ 以上是各种编译方式都必需的依赖项。
- CJK 套件,它的下划线功能依赖 ulem 宏包。
- CJKpunct 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包。
- ⇒ 以上是使用 pdfL^AT_EX 或 L^AT_EX + DVIPDFM_x 的编译方式所需要的依赖项,其中 zhmCJK 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_YL^AT_EX 编译时的依赖项。
- LuaTeX-ja 套件,它还依赖
 - luaotfload 宏包,它还依赖 luatexbase 宏包。
 - xkeyval 宏包。
- ⇒ 以上是使用 LuaL^AT_EX 编译时的依赖项。

CT_EX 依赖的各个宏包和套件均已被 T_EX Live 和 MiK_TE_X 收录。如果你是这两个 T_EX 发行版的用户,可以使用他们提供的宏包管理器来安装这些宏包,或将他们更新到最新版本。如果你不使用这两个发行版,请参考你使用的 T_EX 发行版和相关宏包的说明文档。

出于一些原因,zhmCJK 尚未被收入 T_EX Live 和 MiK_TE_X。因此,若你希望使用 zhmCJK 作为 CT_EX 套件的底层中文支持方式,需要自行安装该宏包。zhmCJK 的安装较为复杂。我们

建议你从 CTAN 下载 zhmcjk 的 **TDS 安装包**，按目录结构将文件复制到 $\text{T}_{\text{E}}\text{X}$ 发行版的本地 TDS 根目录，然后刷新文件名数据库。关于安装的详细内容，可以参照其 **宏包手册** 中第 3 节的指导。

$\text{CT}_{\text{E}}\text{X}$ 的正式版本也已被 $\text{T}_{\text{E}}\text{X}$ Live 和 MiK $\text{T}_{\text{E}}\text{X}$ 收录。因此，使用这两个发行版的用户可以如常使用宏包管理器来安装。如果你不使用者两个发行版，我们建议你从 CTAN 下载 $\text{CT}_{\text{E}}\text{X}$ 的 **TDS 安装包** 按目录结构将文件复制到 $\text{T}_{\text{E}}\text{X}$ 发行版的本地 TDS 根目录，然后刷新文件名数据库。

第 4 节 宏包选项与 \ctexset 命令

$\text{CT}_{\text{E}}\text{X}$ 套件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 $\text{CT}_{\text{E}}\text{X}$ 套件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， $\text{CT}_{\text{E}}\text{X}$ 套件也提供了一系列选项。用户可以使用这些选项来控制 $\text{CT}_{\text{E}}\text{X}$ 套件的行为。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 $\text{CT}_{\text{E}}\text{X}$ 的默认设置。

另一方面，在这些选项里，有些选项只能作为宏包/文档类选项，需要在引入宏包/文档类的时候指定；有些选项只能通过 $\text{CT}_{\text{E}}\text{X}$ 套件提供的用户接口 `\ctexset` 来设定；也有一些选项既可以作为宏包/文档类选项，也可以通过 `\ctexset` 来设定。后续文档中将在使用说明中对每个选项的使用方式做说明。

`\ctexset` `\ctexset {<键值列表>}`

是 $\text{CT}_{\text{E}}\text{X}$ 套件的通用控制命令，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(7.2 节)就可以使用：

例 4

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

`\ctexset` 采用 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}3$ 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 8 节。

第 5 节 编译方式、编码与中文字库

5.1 编译方式

$\text{CT}_{\text{E}}\text{X}$ 套件会根据用户使用的编译方式⁵，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 $\text{CT}_{\text{E}}\text{X}$ 套件的行为，比如宏包对编码的处理。在用户使用 $\text{X}_{\text{L}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 及 $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译时， $\text{CT}_{\text{E}}\text{X}$ 套件使用(且仅能使用)UTF-8 编码；而因为历史原因，在用户使用 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 及 $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译时，宏包默认使用 GBK 编码。用户需要保证编译方式、源文件编码、宏包编码选项三者一致。关于宏包编码选项，可以参考 5.2 节。

⁵ $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 、 $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 、 $\text{X}_{\text{L}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 以及 $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 。

表 2 CT_EX 套件的中文支持方式

(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 CT_EX 套件对字体选择、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。

5.2 中文编码

GBK
UTF8

该选项只能在调用 CT_EX 宏包/文档类时作为选项使用

指明编写文档时使用的编码格式。CT_EX 套件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,CT_EX 套件强制使用 UTF-8 编码,此时 GBK 选项无效。使用 (pdf)L_AT_EX 编译时,CT_EX 套件默认启用 GBK 选项;不过,用户也可以显式声明 UTF8 选项,使 CT_EX 套件工作在 UTF-8 编码下。

5.3 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,CT_EX 套件能自动检测用户使用的编译方式(参见 5.1 节)和操作系统⁶,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 CT_EX 套件自动配置字体策略

	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 10 节有特别说明。

通常,由 CT_EX 套件进行的自动配置已经足够使用,无需用户手工干预;但是 CT_EX 仍然提供了一系列选项,供在 CT_EX 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。

⁶CT_EX 套件现在能够识别 Mac OS X 系统以及 Windows 系统。

zhmap `zhmap = <true|false|zhmCJK>`
 该选项只能在调用 \CTEX 宏包/文档类时作为选项使用,并且只在使用 $\text{pdf}\text{\CTEX}/\text{\CTEX}$ 编译时有意义

指定字体映射机制。

选项为 `true` 时,使用 `zhmetrics` 宏包,将 CJK 字库通过 `\special` 命令映射到 `.ttf` 文件;选项为 `false` 时,使用传统的 CJK 字库 (Type 1)⁷;选项为 `zhmCJK` 时,将载入 `zhmCJK` 宏包⁸,由 `zhmCJK` 宏包提供从 CJK 字库到 `.ttf` 的映射。

fontset `fontset = <none|adobe|fandol|founder|mac|ubuntu|windows|windowsnew|windowsold...>`
 该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置

指定 \CTEX 套件加载的字库。

如果没有指定 `fontset` 的值, \CTEX 套件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)。

\CTEX 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 $\text{pdf}\text{\LATEX}$ 。
<code>fandol</code>	使用 Fandol 中文字体,不支持 $\text{pdf}\text{\LATEX}$ 。
<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 $(\text{pdf})\text{\LATEX}$ 。
<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 $(\text{pdf})\text{\LATEX}$ 编译的时候,若设置 `zhmap = false`(比如需要使用 $\text{\LATEX} + \text{DVIPS}$ 编译),则需要按照传统方式⁹在本地安装好 CJK 字体。

如果不想使用 \CTEX 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。 \CTEX 套件将载入名为 <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 \CTEX 套件 <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 \tn{ctexset}
指定字体。
\end{document}
```

\CTEX 套件预定义的中文字库还定义了一些字体命令。

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

⁷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。

⁸`zhmCJK` 宏包基于 `zhmetrics` 和 `CJK` 宏包,提供与 `xeCJK` 宏包类似的用户接口。

⁹可以使用 `zhmetrics` 宏包提供的脚本 `CTeXFonts.lua`。

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

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

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

在 windowsnew 字库中, 还有 `\yahei`。

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

第 6 节 排版格式设定

6.1 文档默认字号

`zihao = {-4|5|false}`
该选项只能在调用 `CTEX` 宏包/文档类时作为选项使用

将文章默认字号(`\normalsize`)设置为小四号字或五号字,`\false` 禁用本功能。这两个选项可以用于三个 `CTEX` 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。

`cs4size` 和 `c5size` 选项还会将标准文档类中的字体大小命令调整为中文字号(见表 4)。

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 该选项只能在调用 `CTEX` 宏包/文档类时作为选项使用

11pt `CTEX` 文档类是在 `LATEX` 标准文档类之上开发的。因此, 除了可以使用 `CTEX` 宏包定义的字号
12pt 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, `CTEX` 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 `CTEX` 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。`CTEX` 会将这些选项传给

标准文档类¹⁰。

6.2 章节标题风格

heading `heading = <true|false>`
 该选项只能在调用 `ctex.sty` 时作为宏包选项使用

CT_EX 套件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 8 节和 7.3 节。

CT_EX 套件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 套件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 该选项只能在调用 CT_EX 宏包/文档类时作为选项使用
sub4section

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

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

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

具体格式可参考 8 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。

6.3 排版方案选项

scheme `scheme = <chinese|plain>`
 该选项只能在调用 CT_EX 宏包/文档类时作为选项使用

选择文章的排版方案, 预设 `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 7.2 节); 在 `heading = true` 的情况下¹¹ (6.2 节), 还会将章节标题的风格修改为中文样式 (见 8 节)。

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 `\pagestyle`, 并禁用 `autoindent` 选项。事实上, 此时的 CT_EX 套件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>`
 该选项既可以作为宏包/文档类选项使用, 也可以使用 `\ctexset` 来设置

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¹²用占一个汉字宽度, 标号和句内点号占半个汉字宽度;

¹⁰事实上, L^AT_EX 在文档类中的选项是全局设定的, 除了对使用的文档类有影响外, 也可能会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。

¹²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。

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)。

space `space = <true|false|auto>`
该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置
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¹³。
false 总是忽略掉汉字后面的空格,不论汉字后是什么(使用 (pdf)LaTeX 编译时);等同于 auto 的效果(使用 XeLaTeX 编译时)。不建议使用该选项。
auto 根据空格后面的情况决定是否保留:如果空格后面是汉字,则忽略该空格,否则保留。

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

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

linespread `linespread = <数值>`
该选项只能在调用 CTeX 宏包/文档类时作为选项使用
接受一个浮点数值,设置行距倍数。

`scheme = chinese` 时, 默认值为 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` 时, CTeX 套件默认不调整行距倍数,文档中的行距由所选文档类和其他宏包或用户设置决定。

autoindent `autoindent = <true|false|数值|带单位的数值>`
该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置

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

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

`autoindent = true` 等价于设置 `autoindent = 2`。

当 `autoindent = false` 时,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}
```

¹³LaTeX 将单个换行视作一个空格。

linestretch `linestretch = {数值或长度}`
 该选项只能用 `\ctexset` 来设置

`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。

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度。此时汉字间的弹性间距则固定为 `\baselineskip` 的 0.08 倍。

第 7 节 文档汉化

7.1 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“2015 年 5 月 15 日”。

today `today = {small|big|old}`
 该选项只能用 `\ctexset` 来设置

该选项用来控制 `\today` 命令的输出格式:

small 效果为“2015 年 5 月 15 日”。使用阿拉伯数字和汉字的日期格式。
big 效果为“二〇一五年五月十五日”。使用全汉字的日期格式。
old 效果为“May 15, 2015”。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。

7.2 文档标题汉化

这里主要介绍由宏包 `scheme` 选项(6.3 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}`
 该选项只能用 `\ctexset` 来设置

设置目录标题名 `\contentsname`。中文默认为“目录”。

listfigurename `listfigurename = {名字}`
 该选项只能用 `\ctexset` 来设置

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

<u>listtablename</u>	<p><code>listtablename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置表格目录标题名 <code>\listtablename</code>。中文默认为“表格”。</p>
<u>figurename</u>	<p><code>figurename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置图片环境标题名 <code>\figurename</code>。中文默认为“图”。</p>
<u>tablename</u>	<p><code>tablename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置表格环境标题名 <code>\tablename</code>。中文默认为“表”。</p>
<u>abstractname</u>	<p><code>abstractname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置摘要 <code>abstract</code> 环境标题名 <code>\abstractname</code>。中文默认为“摘要”。注意 <code>book</code> 类没有摘要, 该选项无效。</p>
<u>indexname</u>	<p><code>indexname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置索引标题名 <code>\indexname</code>。中文默认为“索引”。</p>
<u>appendixname</u>	<p><code>appendixname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置附录标题名 <code>\appendixname</code>。中文默认为“附录”。</p>
<u>bibname</u>	<p><code>bibname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置</p> <p>设置参考文献标题名 <code>\refname</code>(对 <code>article</code>)或 <code>\bibname</code>(对 <code>report</code> 和 <code>book</code>)。中文默认为“参考文献”。</p>

7.3 页面格式设置与汉化

当使用了 C_TE_X 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, C_TE_X 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。

C_TE_X 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。C_TE_X 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 8 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 L^AT_EX 页面标记的涵义与使用细节,已经超出了本文档讨论的范围。可以参考 [1, Chapter 23]、[2, §4.3, §4.4] 等书籍。

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

例 10

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

在上例中,我们设置了页眉的形式是用破折号分开的节编号与节标题,即“第 1 节——天地玄黄”、“第 2 节——宇宙洪荒”。

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 8 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(6.2 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。

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 name, number, format, nameformat, numberformat, aftername, titleformat, 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。

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

name = {<前名字>,<后名字>}
name = {<前名字>}
该选项只能用 \ctexset 来设置

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 12

```
\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 的默认值	scheme = plain 的默认值	注
part	{第,部分}	{\partname\space}	原 \partname 为 Part
chapter	{第,章}	{\chaptername\space}	原 \chaptername 为 Chapter
section	{}	{}	
subsection	{}	{}	
subsubsection	{}	{}	
paragraph	{}	{}	
subparagraph	{}	{}	

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

number = {⟨数字输出命令⟩}
该选项只能用 `\ctexset` 来设置

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 13

```
\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 的默认值	scheme = plain 的默认值	原 <code>\the⟨标题⟩</code> 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。

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 `section` 一级标题的 `name` 与 `number` 选项:

例 14

```
\ctexset{
  section = {
    name = {第, 节},
    number = \chinese{section}
  }
}
```

part/format
chapter/format
section/format
subsection/format
subsubsection/format
paragraph/format
subparagraph/format
.../format+

format = {<格式命令>}

format+= {<格式命令>}

该选项只能用 \ctexset 来设置

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 的默认值	scheme = plain 的默认值
part (article)	\centering	\raggedright
part	同右	\centering
chapter	\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

part/nameformat
chapter/nameformat
section/nameformat
subsection/nameformat
subsubsection/nameformat
paragraph/nameformat
subparagraph/nameformat
.../nameformat+

nameformat = {<格式命令>}

nameformat+= {<格式命令>}

该选项只能用 \ctexset 来设置

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。

表 8 nameformat 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 的默认值
part (article)	同右	\Large\bfseries
part	同右	\huge\bfseries
chapter	同右	\huge\bfseries
section	同右	{}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

part/numberformat
chapter/numberformat
section/numberformat
subsection/numberformat
subsubsection/numberformat
paragraph/numberformat
subparagraph/numberformat
.../numberformat+

numberformat = {(格式命令)}
numberformat+= {(格式命令)}
该选项只能用 \ctexset 来设置

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。

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 cap 选项下可以做出类似这样的章标题效果：

第 4 章

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

aftername={ (代码)}
该选项只能用 \ctexset 来设置

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。

表 9 aftername 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 的默认值
part (article)	\quad	\par\nobreak
part	同右	\par\vskip 20pt
chapter	\quad	\par\nobreak\vskip 20pt
section	同右	\quad
subsection	同右	\quad
subsubsection	同右	\quad
paragraph	同右	\quad
subparagraph	同右	\quad

part/titleformat
chapter/titleformat
section/titleformat
subsection/titleformat
subsubsection/titleformat
paragraph/titleformat
subparagraph/titleformat
.../titleformat+

titleformat = {(格式命令)}
titleformat+= {(格式命令)}
该选项只能用 \ctexset 来设置

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 的默认值	scheme = plain 的默认值
part (article)	\Large\bfseries	\huge\bfseries
part	\huge\bfseries	\Huge\bfseries
chapter	\huge\bfseries	\Huge\bfseries
section	同右	{}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

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

beforekip = {<弹性间距>}
该选项只能用 \ctexset 来设置

beforekip 选项用于设置章节标题前的垂直间距及章节标题后首段的缩进。

<弹性间距> 的绝对值被用于设置标题间的垂直间距，而 <弹性间距> 的正负号用于设置标题后第一段的首行缩进。当参数是负值时，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；参数是正值时，则保留首行缩进。

使用 sub3section 或 sub4section 宏包选项（见 6.2 节）后，\paragraph 与 \subparagraph 这两级标题会改为排在不同段，会影响 beforekip 选项的默认值。

表 11 beforekip 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 的默认值
part (article)	4ex	-4ex
part	无效	无效
chapter	50pt	-50pt
section	3.5ex plus 1ex minus .2ex	-3.5ex plus -1ex minus -.2ex
subsection	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex
subsubsection	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex
paragraph	同右	3.25ex plus 1ex minus .2ex
(sub3section)	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex
(sub4section)	同上	同上
subparagraph	同右	3.25ex plus 1ex minus .2ex
(sub4section)	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex

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

afterskip = {<弹性间距>}
该选项只能用 \ctexset 来设置

afterskip 选项控制章节标题与后面下方之间的距离。<弹性间距> 的正负号确定标题与后面正文是否排在同一段。如果是正值则正文另起一段，<弹性间距> 给出垂直间距；如果是负值则章节标题与正文第一段排在同一段，<弹性间距> 的绝对值给出水平间距。

默认情况下，\paragraph、\subparagraph 两级标题是与后面正文排在同一段的，afterskip 选项取负数；但使用 sub3section 或 sub4section 宏包选项（见 6.2 节）后，则这两级标题会改为排在不同段。

afterskip 选项的默认值，在 scheme 选项的不同取值下相同。

表 12 afterskip 选项的默认设置

标题名	默认值
part (article)	3ex
part	无效
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 13 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)	0pt

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

indent = {<缩进间距>}
该选项只能用 \ctexset 来设置

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 18

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

part/pagestyle
chapter/pagestyle

pagestyle = {<页面格式>}
该选项只能用 \ctexset 来设置

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 14 pagestyle 选项的默认设置

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

appendix/name

name = {<前名字>,<后名字>}

name = {<前名字>}

该选项只能用 \ctexset 来设置

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 (7.2 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 15 appendix/name 选项的默认设置

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

appendix/number

number = {<数字输出命令>}

该选项只能用 \ctexset 来设置

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 16 appendix/number 选项的默认设置

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

第 9 节 实用命令

9.1 字号与间距

\zihao

\zihao{<字号>}

用于调整字号大小。其中 <字号> 的有效值共有 16 个, 如表 17 所示。使用 \zihao 命令调整字体大小时, 西文字号大小会始终和中文字号保持一致。

表 17 中文字号

〈字号〉	大小 (bp)	意义
0	42	初号
-0	36	小初号
1	26	一号
-1	24	小一号
2	22	二号
-2	18	小二号
3	16	三号
-3	15	小三号
4	14	四号
-4	12	小四号
5	10.5	五号
-5	9	小五号
6	7.5	六号
-6	6.5	小六号
7	5.5	七号
8	5	八号

<u><code>\ziju</code></u>	<code>\ziju{<中文字符宽度的倍数>}</code> 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 <code>\ccwd</code> 的值,但不会影响英文字距。
<u><code>\ccwd</code></u>	当前汉字的字宽保存在长度寄存器 <code>\ccwd</code> 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。
<u><code>\CTEXsetfont</code></u>	更新当前的中文字体信息,包括当前字距(<code>\ccwd</code>)和段首缩进(<code>\parindent</code>)。一般来说,用户无需使用这个命令。

9.2 中文数字转换

CT_EX 套件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。

<u><code>\chinese</code></u>	<code>\chinese {<counter>}</code> <code>\chinese</code> 命令与 <code>\roman</code> 等命令的用法类似,作用在一个 L ^A T _E X 计数器上,将计数器的值以中文数字的形式输出。
<u><code>\zhnumber</code></u>	<code>\zhnumber {<number>}</code> 以中文格式输出数字。这里的数字可以是整数、小数和分数。
<u><code>\zhdigits</code></u>	<code>\zhdigits {<number>}</code> 将阿拉伯数字转换为中文数字串。
<u><code>\CTEXnumber</code></u>	<code>\CTEXnumber \<macro> {<number>}</code> <code>\<macro></code> 必须是一个 T _E X 宏,不需预先定义。 <code>\CTEXnumber</code> 通过 <code>\zhnumber</code> 将 <code><number></code> 转为中文数字,最后将结果存储在 <code>\<macro></code> 里。对 <code>\<macro></code> 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 <code>\CTEXnumber</code> ,直接使用 <code>\zhnumber</code> 即可。但是,如果在文档中需要多次使用同一个数字 <code><number></code> 的中文形式,就可以先用 <code>\CTEXnumber</code> 将结果保存起来备用,而不是每次使用时都用 <code>\zhnumber</code> 现场转换一次。
<u><code>\CTEXdigits</code></u>	<code>\CTEXdigits \<macro> {<number>}</code> <code>\CTEXdigits</code> 与 <code>\CTEXnumber</code> 类似,但其转换的结果是中文数字串,而不是中文数字。

9.3 杂项

<u><code>\CTeX</code></u>	用于显示 CT _E X 标志。
---------------------------	----------------------------

第 10 节 *Lua^AT_EX* 下的中文支持方式

在 *Lua^AT_EX* 下, *CT_EX* 套件依赖 *LuaTeX-j*a 宏包来完成中文支持。该宏包是日本 *T_EX* 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 *Lua^AT_EX* 引擎下实现日本 *pT_EX* 引擎的(大部分)功能。它为了兼容 *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, 因而 *CT_EX* 禁用了它在 *L^AT_EX* 格式下的大部分设置, 只保留了必要的部分。同时修改了它的字体设置方式, 使得相关命令与 *xeCJK* 宏包大致相同。

20150420 版以后的 *LuaTeX-j*a 宏包开始支持竖排, 但 *CT_EX* 暂不支持竖排。

10.1 *Lua^AT_EX* 下替代字体的设置

AlternateFont

```
\setCJKfamilyfont {<family>}
[
  AlternateFont =
  {
    {<character range1>} [<alternate font features1>] {<alternate font name1>} ||
    {<character range2>} [<alternate font features2>] {<alternate font name2>} ||
    .....
  } ,
  <base font features>
] {<base font name>}
```

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

CharRange

```
\setCJKfamilyfont {<family>}
[
  CharRange = {<character range>} ,
  <alternate font features>
] {<alternate font name>}
```

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

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

例 19

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

declarecharrange

```
\ctexset
{
  declarecharrange =
  {
    {<name1>} {<character range1>} ,
    {<name2>} {<character range2>} ,
    ...
  }
}
```

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

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

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

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

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

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

第 11 节 C_TE_X 套件的配置文件

C_TE_X 套件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 套件的默认行为。

在多数情况下,并不需要修改配置文件,C_TE_X 套件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 套件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 套件的源代码一样,配置文件采用 L^AT_EX3 的语法编写。

C_TE_X 套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 套件的更新而丢失,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。

11.1 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 20

```
% 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_T_EX 套件的开始处，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。

11.2 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 21

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

例 22

```
% 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。
\etex_if_engine:T
{
  \xeCJKsetup { CheckSingle }
}
% 在使用 LuaTeX 编译时，设置 LuaTeX-jan 的 jcharwidowpenalty 参数。
\luatex_if_engine:T
{
  \ltjsetparameter { jcharwidowpenalty = 10000 }
}
```

11.3 配置标题中文翻译

由于 C_T_EX 套件需要同时支持 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。

11.4 自定义字体集

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

第 12 节 对旧版本的兼容性

12.1 C_T_EX 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

12.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

<u>cs4size</u> <u>c5size</u>	分别相当于 zihao=-4 和 zihao=5, 过时选项。
<u>CCT</u> <u>CCTfont</u>	相关选项已删除。
<u>indent</u> <u>noindent</u>	使用 indent 宏包选项会载入 indentfirst 宏包。过时选项。 noindent 什么也不做, 也不会使章节首行停止缩进。需要使用 \ctexset 命令设置章节的 beforeskip 选项为正数才能完成原来 noindent 选项的工作。另外, 设置 scheme = plain 可以保持原有英文文档类的效果, 使章节首行停止缩进。过时选项。
<u>zhmap</u> <u>nozhmap</u>	zhmap 宏包选项增加了参数, 扩充了功能, 除了支持真假值参数外, 还支持选择 zhmcjk 作为底层中文处理宏包。(5.3 节) nozhmap 选项相当于 zhmap=false。过时选项。
<u>winfonts</u> <u>adobefonts</u> <u>nofonts</u>	宏包选项 winfonts 相当于 fontset=windows, adobefonts 相当于 fontset=adobe, nofonts 相当于 fontset=none。这几个选项是过时选项, 对于新文档, 应使用 fontset 选项设置不同字体集。 另外, 第 2 版 C _T E _X 宏包的默认字体不再是 Windows 系统字体, 而是根据检测到的操作系统选择使用 Windows、Mac 的系统字体还是 Fandol 字体(5.3 节)。
<u>punct</u> <u>nopunct</u>	旧版本中宏包 punct 选项没有参数, 现在可以用参数设定标点风格(6.3 节)。原有无参形式的 punct 选项相当于 punct=quanjiao。 旧版宏包中 nopunct 选项的效果大致相当于 punct=plain。过时选项, 不推荐使用。
<u>cap</u> <u>nocap</u>	原有的 cap 和 nocap 选项由新的 scheme 选项代替。(6.3 节) cap 选项相当于 scheme = chinese, nocap 选项相当于 scheme = plain。它们均已过时, 仅因兼容性而保留。
<u>space</u> <u>nospace</u>	新版本宏包 space 选项增加真假值参数。(6.3 节) nospace 选项相当于 space=false, 成为过时选项。

<u>fancyhdr</u>	<p>新版本宏包中总是自动处理对 fancyhdr 宏包的兼容性, 而由用户自己使用 \usepackage 载入 fancyhdr 宏包。</p> <p>fancyhdr 选项过时, 因兼容性保留, 功能是载入 fancyhdr 宏包。</p>
<u>hyperref</u>	<p>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。</p> <p>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。</p>
<u>fntef</u>	<p>旧版本的 fntef 选项用于统一 CCTfntef 与 CJKfntef 的界面, 新版本 CT_EX 套件不再支持 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。</p>
<u>\CTEXunderdot</u> <u>\CTEXunderline</u> <u>\CTEXunderdblline</u> <u>\CTEXunderwave</u> <u>\CTEXsout</u> <u>\CTEXxout</u> <u>\CTEXfilltwosides</u>	<p>在调用 fntef 宏包选项的同时, 旧版本 CT_EX 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 CT_EX 套件中已失去意义。</p> <p>此外, 在 pdf_TE_X 引擎下, 用于设置格式的 \CJKunderdotbasesep 等宏也被换名为 \CTEXunderdotbasesep 等宏。</p> <p>在新版本中, 上述由 fntef 衍生的相关宏都成为过时命令。</p>
<u>\CTEXindent</u>	更新 \ccwd 宽度后设置 \parindent=2\ccwd。过时命令。
<u>\CTEXnoindent</u>	设置 \parindent=0pt。过时命令。
<u>\CTEXsetup</u>	<p>\CTEXsetup[⟨选项⟩]{⟨标题⟩}</p> <p>相当于设置了 \ctexset{⟨标题⟩ = {⟨选项⟩}}。过时命令。</p>
<u>\CTEXoptions</u>	<p>\CTEXoptions[⟨选项⟩]</p> <p>相当于设置了 \ctexset{⟨选项⟩}。过时命令。</p>
<u>\Chinese</u>	<p>\Chinese[⟨counter⟩]</p> <p>新版套件中 \chinese 统一了旧版本中 \chinese 和 \Chinese 的功能。因此, 该命令已过时。</p>
<u>captiondelimter</u>	<p>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 CT_EX 宏包中失效。</p> <p>可以使用 caption 宏包的 labelsep 选项来完成同样的功能。</p>

例 23

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

12.3 CT_EX 1.02c 以后的 SVN 开发版

CT_EX 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。

CT_EX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

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)

第 14 节 参考文献

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

第 15 节 代码实现

```

1 <@@=ctex>
2 <*ctexcap>
3 \PassOptionsToPackage { heading = true } { ctexcap }
4 \RequirePackageWithOptions { ctex }
5 </ctexcap>
6 <*class|style|ctexsize>
7 \RequirePackage { xparse , l3keys2e }
8 </class|style|ctexsize>

```

检查 expl3 和 l3keys2e 的版本。

```

9 <*class>
10 \msg_new:nnnn { ctex } { l3-too-old }
11 { Support~package~`#1'~too~old. }
12 {
13   Please~update~an~up~to~date~version~of~the~bundles\\
14   \l3kernel'~and~\l3packages'\\
15   using~your~TeX~package~manager~or~from~CTAN.
16 }
17 \@ifpackagelater { expl3 } { 2014/07/20 } { }
18 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
19 \@ifpackagelater { l3keys2e } { 2014/05/05 } { }
20 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
21 </class>
22 <*class|style>
23 \RequirePackage { etoolbox , ifpdf , fix-cm , everyysel }

```

15.1 内部函数与变量

```

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_dim
\l__ctex_tmp_box
24 \tl_new:N \l__ctex_tmp_tl
25 \int_new:N \l__ctex_tmp_int
26 \dim_new:N \l__ctex_tmp_dim
27 \box_new:N \l__ctex_tmp_box

```

(End definition for \l__ctex_tmp_tl and others.)

对旧版本的宏包给出错误信息。

```

28 \msg_new:nnnn { ctex } { package-too-old }
29 { Support~package~`#1'~too~old. }
30 {
31   Please~update~an~up~to~date~version~of~the~package~`#1'\
32   using~your~TeX~package~manager~or~from~CTAN.
33 }

```

\ctex_lua_now_x:n 最新的 expl3 去掉了 l3luatex 模块,因而 \lua_now_x:n 不再有定义。

```

34 \cs_new_eq:NN \ctex_lua_now_x:n \luatex_directlua:D

```

(End definition for \ctex_lua_now_x:n.)

\ctex_if_pdfmode:TF \ifpdf 的简单 wrapper。

```

35 \ifpdf
36 \cs_new_eq:NN \ctex_if_pdfmode:TF \use_i:nn
37 \else:
38 \cs_new_eq:NN \ctex_if_pdfmode:TF \use_ii:nn
39 \fi:

```

(End definition for \ctex_if_pdfmode:TF.)

`\ctex_if_preamble:TF` 测试是否在 $\text{\LaTeX}2_{\epsilon}$ 的导言区。 $\text{\LaTeX}2_{\epsilon}$ 中的 `\@onlypreamble` 命令可以处理宏参数,使其在 document 环境后被重定义为 `\@notprerr`,而又由于 `\@onlypreamble` 本身也被这样处理过,因此可以测试 `\@onlypreamble` 是否与 `\@notprerr` 相同来确定是否在导言区。

```

40 \prg_new_conditional:Npnn \ctex_if_preamble: { TF }
41 {
42   \if_meaning:w \@onlypreamble \@notprerr
43   \prg_return_false:
44   \else:
45     \prg_return_true:
46   \fi:
47 }

```

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

`\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。

```

48 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
49 {
50   \ifpackageloaded { #1 }
51     { #2 }
52     { \ctex_add_hook:cn { g__ctex_at_end_ #1 _hook_tl } { #2 } }
53 }

```

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

`\ctex_add_hook:Nn` 给钩子附加内容。

```

\ctex_add_hook:cn
54 \cs_new_protected:Npn \ctex_add_hook:Nn #1#2
55 {
56   \tl_if_exist:NF #1 { \tl_new:N #1 }
57   \tl_gput_right:Nn #1 { #2 }
58 }
59 \cs_generate_variant:Nn \ctex_add_hook:Nn { c }

```

(End definition for `\ctex_add_hook:Nn` and `\ctex_add_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
60 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
61 {
62   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
63   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
64 }
65 \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。

```

66 \tl_put_left:Nn \@popfilename
67 { \ctex_package_end_hook:o { \@currname } }

```

`\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 $\text{\LaTeX}3$ 语法环境,并设置 `@` 的 `\catcode` 为 11。

```

68 \cs_new_protected:Npn \ctex_file_wrapper:nnn #1#2#3
69 {
70   \use:x
71   {
72     \ExplSyntaxOff
73     \char_set_catcode_letter:n { 64 }
74     #1
75     \exp_not:n {#2}

```

```

76      \bool_if:NTF \l__kernel_expl_bool
77      { \ExplSyntaxOn }
78      { \ExplSyntaxOff }
79      \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
80      #3
81    }
82  }

```

(End definition for `\ctex_file_wrapper:nnn`.)

`\ctex_file_input:n` 输入文件。

```

83 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
84 { \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。

```

85 \cs_new_protected_nopar:Npn \ctex_scheme_input:n #1
86 {
87   \ctex_file_wrapper:nnn
88   { }
89   {
90     \tl_if_exist:NTF \c__ctex_class_tl
91     {
92       \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
93       { \file_input:n { ctex-scheme- #1 .def } }
94     }
95     { \file_input:n { ctex-scheme- #1 .def } }
96   }
97   { }
98 }
99 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

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

`\ctex_parse_name:NN` 用 `\DeclareRobustCommand` 定义的宏或者由 `\newcommand` 或 `\newrobustcmd` 定义的带一个可选参数的宏第一次展开的结果都不是其实际定义, 实际定义被保存在另外的宏中。由这些命令定义的宏的第一次展开结果可以有下面的形式(细节可查阅 `xpatch` 的文档):

```

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

```

`etoolbox` 的 `\patchcmd` 的主要原理是先对宏的 `\meaning` 作字符串替换, 然后再用 `\scantokens` 来重建它。我们希望对宏的实际定义打补丁, 为此需要先得到对应的名字。`letltxmacro`、`show2e` 和 `xpatch` 宏包中都有类似的工作。我们不想依赖 `xpatch`, 主要是因为它与同作者的 `regexpatch` 宏包共用了主要函数的名字, 从而将导致用户不能使用 `regexpatch`。

```

100 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
101 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
102 \group_begin:
103 \char_set_lccode:nn { `< } { `< }
104 \char_set_lccode:nn { `\/ } { `\/ }
105 \char_set_lccode:nn { `A } { `t }
106 \tl_map_function:nN { \A \E \S \O \P } \char_set_catcode_other:N
107 \tex_lowercase:D
108 {
109   \group_end:

```

```

110 \cs_new_protected:Npn \ctex_parse_name:NNn #1#2#3
111 {
112   \bool_if:nTF { \cs_if_exist_p:c { #3 ~ } || \cs_if_exist_p:c { /#3 } }
113   {
114     \group_begin:
115     \use:x
116     {
117       \__ctex_parse_name:nNNNnN { \token_get_replacement_spec:N #2 }
118       \exp_not:N #2 \exp_not:c { #3 ~ } \exp_not:c { /#3 } {#3}
119     } #1
120   }
121   { #1#2 }
122 }
123 \cs_new_protected:Npn \__ctex_parse_name:nNNNnN #1#2#3#4#5#6
124 {
125   \group_end:
126   \exp_args:Nc #6
127   {
128     \str_case:nnTF {#1}
129     {
130       { \protect #3 } { }
131       { \x@protect #2 \protect #3 } { }
132     }
133     {
134       \str_if_eq_x:nnTF { \exp_not:n { /@protected@ #3 /#3 } }
135       {
136         \exp_last_unbraced:Nf \__ctex_parse_name:w
137         \token_get_replacement_spec:N #3 AESAOPA ~ < \q_stop
138       }
139       { /#5 ~ } { #5 ~ }
140     }
141     {
142       \str_case:onTF { \__ctex_parse_name:w #1 AESAOPA ~ < \q_stop }
143       {
144         { /@protected@ #2 #4 } { }
145         { /@ #4 } { }
146       }
147       { /#5 } {#5}
148     }
149   }
150 }
151 \cs_new:Npn \__ctex_parse_name:w #1 AESAOPA ~ #2 < #3 \q_stop { #1#2 }
152 }
153 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

(End definition for `\ctex_parse_name:NN`.)

```

\ctex_patch_cmd:NnnTF
\ctex_preto_cmd:NnTF
\ctex_appto_cmd:NnTF

```

在打补丁前先解析实际名字。

```

154 \cs_new_protected:Npn \ctex_patch_cmd:NnnTF { \ctex_parse_name:NN \patchcmd }
155 \cs_new_protected:Npn \ctex_preto_cmd:NnTF { \ctex_parse_name:NN \pretocmd }
156 \cs_new_protected:Npn \ctex_appto_cmd:NnTF { \ctex_parse_name:NN \apptocmd }

```

(End definition for `\ctex_patch_cmd:NnnTF`, `\ctex_preto_cmd:NnTF`, and `\ctex_appto_cmd:NnTF`.)

`\ctex_patch_cmd:Nnn` 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。并且在补丁的时候关闭 \LaTeX 3 语法。

```

157 \cs_new_protected:Npn \ctex_patch_cmd:Nnn
158 {
159   \group_begin:
160   \char_set_catcode_other:N \#
161   \__ctex_patch_cmd:Nnn
162 }
163 \cs_new_protected:Npn \__ctex_patch_cmd:Nnn #1#2#3
164 {

```

```

165 \group_end:
166 \group_begin:
167 \ExplSyntaxOff
168 \ctex_patch_cmd:NnnTF #1 {#2} {#3}
169 {
170   \cs_gset_eq:NN \__ctex_tmp:w #1
171   \group_end:
172   \cs_set_eq:NN #1 \__ctex_tmp:w
173   \cs_undefine:N \__ctex_tmp:w
174 }
175 { \group_end: \ctex_patch_failure:N #1 }
176 }
177 \cs_new_protected:Npn \ctex_patch_failure:N #1
178 { \msg_warning:nxx { ctex } { patch-failure } { \token_to_str:N #1 } }
179 \msg_new:nnn { ctex } { patch-failure }
180 {
181   Patching~command~`#1'~failed.\\
182   ctex~may~not~work~as~expected.
183 }

```

(End definition for \ctex_patch_cmd:Nnn.)

\ctex_set_default_ccwd:Nn 若参数 #2 带长度单位,则设置它为 t1 变量 #1 的值,否则以 \ccwd 为单位。

```

184 \cs_new:Npn \ctex_set_default_ccwd:Nn #1#2
185 {
186   \dim_compare:nNnTF
187     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
188     { \tl_set:Nn #1 {#2} }
189     { \tl_set:Nn #1 { #2 \ccwd } }
190 }

```

(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。

```

191 \cs_new:Npn \ctex_default_pt:n #1
192 {
193   \exp_after:wN \__ctex_default_pt:w
194   \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
195 }
196 \group_begin:
197 \char_set_catcode_other:N \P
198 \char_set_catcode_other:N \T
199 \tex_lowercase:D
200 {
201   \group_end:
202   \cs_new:Npn \__ctex_default_pt:w #1 PT #2 \q_stop { #1 PT }
203 }

```

(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。

```

204 \tl_new:N \l__ctex_encoding_tl
205 \tl_set:Nx \l__ctex_encoding_tl
206 { \pdfTeX_if_engine:TF { GBK } { UTF8 } }

```

(End definition for \l__ctex_encoding_tl.)

\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。

```

207 \cs_new_eq:NN \g__ctex_section_depth_flag \c_two

```

(End definition for \g__ctex_section_depth_flag.)

\g__ctex_zhmCJK_bool 是否使用 zhmCJK 宏包。

208 \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。

209 \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。

210 \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。

211 \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。

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

213 <style>\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。

214 <*class>

215 \clist_new:N \g__ctex_std_options_clist

216 </class>

(End definition for \g__ctex_std_options_clist.)

对无效选项给出警告。

217 \msg_new:nnn { ctex } { invalid-option }

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

219 \msg_new:nnn { ctex } { invalid-value }

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

对过时选项或命令给出警告。

221 \msg_new:nnn { ctex } { deprecated-option }

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

223 \msg_new:nnn { ctex } { deprecated-command }

224 { Command~ #1 is~ deprecated.\\ #2 }

225 \msg_new:nnn { ctex } { deprecated-environment }

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

227 </class|style>

228 <*class|style|ctexsize>

\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。

229 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one

(End definition for \g__ctex_font_size_flag.)

15.2 宏包选项

```
230 \keys_define:nn { ctex / option }
231 {
```

zihao

```
232   zihao .choice: ,
233   zihao .default:n = { 5 } ,
234   zihao / 5 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero } ,
235   zihao / -4 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_one } ,
236   zihao / false .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
237   <texsize> }
238 </class|style|ctexsize>
239 <*class|style>
240   c5size .code:n =
241   {
242     \msg_warning:nnn { ctex } { deprecated-option }
243     { Option~`zihao=5'~ is~ set. }
244     \keys_set:nn { ctex / option } { zihao = 5 }
245   } ,
246   cs4size .code:n =
247   {
248     \msg_warning:nnn { ctex } { deprecated-option }
249     { Option~`zihao=-4'~ is~ set. }
250     \keys_set:nn { ctex / option } { zihao = -4 }
251   } ,
252   c5size .value_forbidden: ,
253   cs4size .value_forbidden: ,
```

(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。

```
254   linespread .fp_set:N = \l__ctex_line_spread_fp ,
255   linespread .initial:n = { \c_nan_fp } ,
256   linespread .value_required: ,
```

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

autoindent 自动调整段落的首行缩进功能。

```
257   autoindent .choice: ,
258   autoindent .default:n = { true } ,
259   autoindent / true .code:n =
260   {
261     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
262     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
263   } ,
264   autoindent / false .code:n =
265   {
266     \tl_clear:N \l__ctex_autoindent_tl
267     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
268   } ,
269   autoindent / unknown .code:n =
270   {
271     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
272     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
273   } ,
```

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

indent 仅为兼容性保留, 已过时。

```
274   indent .code:n =
275   {
276     \msg_warning:nnn { ctex } { deprecated-option }
```



```

277     {
278         The~ indentfirst~ package~ will~ be~ loaded~ but~ the~
279         functionality~ may~ be~ removed~ in~ later~ version.
280         It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
281         options.~
282     }
283     \keys_set:nn { ctex / option } { autoindent = true }
284 },
285 indent .value_forbidden: ,
286 noindent .code:n =
287 {
288     \msg_warning:nnn { ctex } { deprecated-option }
289     {
290         The~ functionality~ has~ been~ removed.~
291         It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
292         options.~
293     }
294     \keys_set:nn { ctex / option } { autoindent = false }
295 },
296 noindent .value_forbidden: ,

```

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

GBK
UTF8

```

297 GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
298 UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
299 GBK .value_forbidden: ,
300 UTF8 .value_forbidden: ,

```

(End definition for GBK and UTF8. These functions are documented on page 7.)

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。

```

301 fontset .tl_gset:N = \g__ctex_fontset_tl ,
302 nofonts .code:n =
303 {
304     \msg_warning:nnn { ctex } { deprecated-option }
305     {
306         Option~ `fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
307         fontset~ option.
308     }
309     \keys_set:nn { ctex / option } { fontset = none }
310 },
311 adobefonts .code:n =
312 {
313     \msg_warning:nnn { ctex } { deprecated-option }
314     {
315         Option~ `fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
316         fontset~ option.
317     }
318     \keys_set:nn { ctex / option } { fontset = none }
319 },
320 winfonts .code:n =
321 {
322     \msg_warning:nnn { ctex } { deprecated-option }
323     {
324         Option~ `fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
325         fontset~ option.
326     }
327     \keys_set:nn { ctex / option } { fontset = none }
328 },
329 nofonts .value_forbidden: ,
330 winfonts .value_forbidden: ,
331 adobefonts .value_forbidden: ,

```

(End definition for `fontset`. This function is documented on page 8.)

zhmap

```

332     zhmap .choice: ,
333     zhmap .default:n = { true } ,
334     zhmap / zhmCJK .code:n =
335     {
336         \bool_gset_true:N \g__ctex_zhmCJK_bool
337         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
338     } ,
339     zhmap / true .code:n =
340     {
341         \bool_gset_false:N \g__ctex_zhmCJK_bool
342         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
343     } ,
344     zhmap / false .code:n =
345     {
346         \bool_gset_false:N \g__ctex_zhmCJK_bool
347         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
348     } ,
349     nozhmap .code:n =
350     {
351         \msg_warning:nnn { ctex } { deprecated-option }
352         { Option~`zhmap=false'~ is~ set. }
353         \keys_set:nn { ctex / option } { zhmap = false }
354     } ,
355     nozhmap .value_forbidden: ,

```

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

punct 设置标点符号输出格式。

```

356     punct .tl_set:N = \l__ctex_punct_tl ,
357     punct .default:n = { quanjiao } ,
358     punct .initial:n = { quanjiao } ,
359     nopunct .code:n =
360     {
361         \msg_warning:nnn { ctex } { deprecated-option }
362         { Option~`punct=plain'~ is~ set. }
363         \keys_set:nn { ctex / option } { punct = plain }
364     } ,
365     nopunct .value_forbidden: ,

```

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

space

```

366     space .choices:nn =
367     { true , auto , false }
368     {
369         \exp_args:Nx \ctex_at_end:n
370         { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
371     } ,
372     space .default:n = { true } ,
373     nospace .code:n =
374     {
375         \msg_warning:nnn { ctex } { deprecated-option }
376         { Option~`space=false'~ is~ set. }
377         \keys_set:nn { ctex / option } { space = false }
378     } ,
379     nospace .value_forbidden: ,

```

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

heading

```

380     heading .bool_set:N = \l__ctex_heading_bool ,

```

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

scheme

```

381 scheme .tl_set:N = \l__ctex_scheme_tl ,
382 scheme .default:n = { chinese },
383 scheme .initial:n = { chinese },
384 cap .code:n =
385 {
386   \msg_warning:nnn { ctex } { deprecated-option }
387   { Option~ `scheme = chinese' ~ is~ set. }
388   \keys_set:nn { ctex / option } { scheme = chinese }
389 } ,
390 nocap .code:n =
391 {
392   \msg_warning:nnn { ctex } { deprecated-option }
393   { Option~ `scheme = plain' ~ is~ set. }
394   \keys_set:nn { ctex / option } { scheme = plain }
395 } ,
396 cap .value_forbidden: ,
397 nocap .value_forbidden: ,

```

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

sub3section

sub4section

```

398 sub3section .code:n =
399   { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
400 sub4section .code:n =
401   { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
402 sub3section .value_forbidden: ,
403 sub4section .value_forbidden: ,

```

(End definition for sub3section and sub4section. These functions are documented on page 10.)

fntef

fancyhdr

hyperref

```

404 fntef .code:n =
405 {
406   \msg_warning:nnn { ctex } { deprecated-option }
407   { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
408 } ,
409 fancyhdr .code:n =
410 {
411   \msg_warning:nnn { ctex } { deprecated-option }
412   { `fancyhdr'~ package~ is~ loaded. }
413   \RequirePackage{ fancyhdr }
414 } ,
415 hyperref .code:n =
416 {
417   \msg_warning:nnn { ctex } { deprecated-option }
418   { `hyperref'~ package~ will~ be~ loaded. }
419   \AtEndPreamble { \RequirePackage { hyperref } }
420 } ,
421 }

```

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

```

422 </class|style>
423 <*class|style|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

```

424 \tl_clear:N \l__ctex_tmp_tl
425 \clist_map_inline:nn

```

```

426 {
427     10pt , 11pt , 12pt ,
428     8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
429 }
430 {
431     \tl_put_right:Nn \l__ctex_tmp_tl
432     {
433         #1 .code:n =
434         <!*class>
435             { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
436         </!class>
437         <*class>
438             {
439                 \cs_gset_eq:NN \g__ctex_font_size_flag \c_two
440                 \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
441             } ,
442         </class>
443         #1 .value_forbidden: ,
444     }
445 }
446 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
447 \tl_clear:N \l__ctex_tmp_tl

```

(End definition for 10pt, 11pt, and 12pt. These functions are documented on page 9.)

将未知选项传给标准文档类。

```

448 <*class>
449 \keys_define:nn { ctex / option }
450 {
451     unknown .code:n =
452         { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
453 }
454 </class>
455 <|ctexsize>\ctex_file_input:n { ctexopts.cfg }
456 \ProcessKeysOptions { ctex / option }
457 </class|style|ctexsize>
458 <*class|style>

```

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。

```

459 <*class>
460 \if_case:w \g__ctex_font_size_flag
461     \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
462 \or:
463     \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
464 \fi:

```

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

```

465 <*article>
466 \tl_const:Nn \c__ctex_class_tl { article }
467 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
468 \LoadClass { article }
469 </article>
470 <*book>
471 \tl_const:Nn \c__ctex_class_tl { book }
472 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
473 \LoadClass { book }
474 </book>
475 <*report>
476 \tl_const:Nn \c__ctex_class_tl { report }
477 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
478 \LoadClass { report }
479 </report>
480 </class>

```

```

481 \tl_set_eq:Nc \l__ctex_tmp_tl { ver@ \@currname . \@currentx }
482 <*class>
483 \cs_new_eq:cN { ver@ctex. \@pkgextension } \l__ctex_tmp_tl
484 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \l__ctex_tmp_tl
485 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \l__ctex_tmp_tl
486 </class>
487 <*style>
488 \msg_new:nnnn { ctex } { ctexsize-loaded }
489 { Package~`ctexsize'~can~not~be~loaded~before~`ctex'. }
490 {
491   `ctexsize'~is~actually~a~part~of~`ctex'.\\
492   It~is~not~necessary~to~load~it~separately.
493 }
494 \@ifpackageloaded { ctexsize }
495 { \msg_error:nn { ctex } { ctexsize-loaded } }
496 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \l__ctex_tmp_tl }
497 </style>

```

15.3 用户设置接口

\ctexset

```

498 \NewDocumentCommand \ctexset { +m }
499 { \keys_set:nn { ctex } {#1} }

```

(End definition for \ctexset. This function is documented on page 6.)

\CTEXsetup 过时命令。
\CTEXoptions

```

500 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
501 {
502   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
503   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
504   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
505 }
506 \NewDocumentCommand \CTEXoptions { +o }
507 {
508   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
509   { \ctexset~ {~ #1~ }~ is~ set. }
510   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
511 }

```

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

```

512 </class|style>

```

15.4 特定引擎支持与设置

15.4.1 ctex-engine-pdfTeX.def

```

513 <*pdfTeX>

```

首先检查选项, 决定是否载入 zhmCJK 宏包。

```

514 \if_bool:N \g__ctex_zhmCJK_bool
515 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
516 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

517 \else:
518 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
519 { \RequirePackage { CJK } }
520 { \RequirePackage { CJKutf8 } }
521 \RequirePackage { CJKpunct , CJKspace }

```

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

```

522 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
523 {
524     \tl_set:Nn \CJKrmdefault {#1}
525     \tl_set:Nn \CJKsfdefault {#2}
526     \tl_set:Nn \CJKttdefault {#3}
527     \AtBeginDvi { \file_input:n {#4} }
528     \ctex_at_end_package:nn { atbegshi }
529     { \AtBeginShipoutFirst { \file_input:n {#4} } }
530 }
531 \@onlypreamble \ctex_load_zhmap:n

(End definition for \ctex_load_zhmap:nnnn.)

532 \tl_if_exist:NF \CJKfamilydefault
533 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
534 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
535 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
536 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
537 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { } { }
538 { \ctex_patch_failure:N \rmfamily }
539 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { } { }
540 { \ctex_patch_failure:N \sffamily }
541 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { } { }
542 { \ctex_patch_failure:N \ttfamily }
543 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
544 { \cs_set_eq:NN \reset@font \normalfont }
545 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```
546 \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。

```

547 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
548 {
549     \ctex_file_wrapper:nnn
550     {
551         \char_set_catcode_other:n { 60 } % <
552         \char_set_catcode_math_superscript:n { 94 } % ^
553         \int_set_eq:NN \tex_endlinechar:D \c_minus_one
554     }
555     { \file_input:n {#1} }
556     {
557         \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
558         \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
559         \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
560     }
561 }
562 \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¹⁴。我们在这里给出 `\CJK@surr` 的一个不依赖 `\uppercase` 的实现。

```

563 \if_cs_exist:N \CJK@surr
564 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
565 {

```

¹⁴<https://github.com/CTeX-org/ctex-kit/issues/146>

```

566 \tl_set:Nx \l__ctex_tmp_tl {#2}
567 \int_set:Nn \l__ctex_tmp_int
568 { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
569 \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
570 { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
571 {
572 \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
573 \tl_gset:Nx #1
574 {
575 \int_to_Hex:n
576 { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
577 \int_to_Hex:n
578 { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
579 }
580 }
581 }
582 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
583 \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。

```

584 \AtBeginDocument
585 {
586 \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
587 { \punctstyle { \l__ctex_punct_tl } }
588 }

```

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。

```

589 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
590 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
591 {
592 \CJK@makeActive
593 \CJK@loadBinding { standard }
594 }

```

在导言区结束时调用 CJK@envStart 启用完整的中文功能。

CJK@envStart 的定义是

```

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

```

CJK@upperReset 可能会有一定风险，因此我们直到导言区末尾才使用 CJK@envStart。

这样可以避免将 CJK 环境内置入 document 环境的最里层, 最后也就不需要 `\clearpage`。
zhmCJK 已经提供类似功能。

```

595 \exp_args:Nx \AtEndPreamble
596 {
597   \exp_not:N \CJK@envStart
598   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
599   \exp_not:N \CJKtilde
600 }

```

zhmCJK 判断结束。

```

601 \fi:

```

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

```

602 <@@=>
603 \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。

```

604 <@@=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。

```

605 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
606 {
607   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
608   { \cs_set_eq:NN \CJK@ignorespaces #1 }
609   \ctex_set_ignorespaces:
610 }
611 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
612 { \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。

```

613 \ctex_add_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。

```

614 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
615 {
616   \clist_map_inline:Nn \c__ctex_punct_family_clist
617   {
618     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
619     {
620       \cs_set_eq:cc
621       { CJKpunct@ ##1 @spaces }
622       { c__ctex_ #1 ##1 _punct_spaces_tl }
623     }
624   }
625 }
626 \clist_const:Nn \c__ctex_punct_family_clist
627 {
628   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
629   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
630 }

```


(End definition for \ctex_punct_set:n.)

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

```
631 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
632 {
633   \cs_if_free:cF { CJKpunct@ #2 @spaces }
634   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
635 }
```

(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。

```
636 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
637 {
638   \clist_map_inline:nn {#1}
639   {
640     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
641     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
642   }
643 }
644 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
645 {
646   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
647   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
648   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
649   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
650   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
651   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
652 }
```

(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。

```
653 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
654 {
655   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
656   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
657   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
658   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
659   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
660   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
661 }
```

(End definition for \ctex_punct_map_itshape:nn.)

\ctex_punct_space:nn 定义标点的边界信息。

\ctexspadef

```
662 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
663 { \tl_const:cn { c_#1 _punct_spaces_tl } {#2} }
664 \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。

```
665 \ctex_file_input:n { ctexspa.def }
666 </pdfTeX>
```

15.4.2 ctex-engine-xetex.def

```
667 <*xetex>
668 \RequirePackage { xeCJK }
```

```

669 \exp_args:Nx \xeCJKsetup
670 {
671     LoadFandol = false ,
672     AutoFakeBold = true ,
673     PunctStyle = \l__ctex_punct_tl
674 }

```

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。

```

675 \@ifpackagelater { fontspec } { 2014/05/25 } { }
676 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
677 </xetex>

```

15.4.3 ctex-engine-luatex.def

```

678 <*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。

```

679 \msg_new:nnn { ctex } { luatexja-loaded }
680 {
681     Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
682     Loading~file~`#1'~will~abort!
683 }
684 \@ifpackageloaded { luatexja }
685 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
686 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
687 \RequirePackage { luatexja }
688 \@ifpackagelater { luatexja-core } { 2015/02/22 } { }
689 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
690 \RequirePackage { fontspec }
691 \@ifpackagelater { fontspec } { 2014/05/25 } { }
692 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

15.4.3.1 LuaTeX-ja 的默认设置

```

693 \ExplSyntaxOff

```

以下设置抄录自 ltjdefs.sty。

```

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

```

以下设置抄录自 ltj-latex.sty。

```

712 \directlua{
713     local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
714     luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
715 }
716 \ltjsetparameter{kanjiskip}{\z@ plus .4pt minus .4pt,

```

```

717 xkanjiskip=.25\zw plus 1pt minus 1pt,
718 autospacing, autoxspacing, jacharrange={-1},
719 yalbaselineshift=\z@, yjabaselineshift=\z@,
720 jcharwidowpenalty=500, differentjfm=paverage
721 }
722 \ExplSyntaxOn

```

15.4.3.2 LuaTeX-jā 的补丁

```
723 <@@=ctex_ltj>
```

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

```

724 \RequirePackage { xunicode-addon }
725 \AtBeginUTFCommand
726 {
727   \group_begin:
728   \ctex_lua_now_x:n { tex.globaldefs = 0 }
729   \ltj@allalchar
730 }
731 \AtEndUTFCommand { \group_end: }

```

对 fontspec 沿用 LuaTeX-jā 的补丁。

```
732 \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。

```

733 \group_begin:
734 \char_set_catcode_other:n { \c_zero }
735 \cs_new_protected:Npn \__ctex_ltj_um_char:Nw #1 = #2 \q_nil
736 {
737   \group_begin:
738   \char_set_lccode:nn { \c_zero } {#2}
739   \tex_lowercase:D
740   {
741     \group_end:
742     \cs_gset_protected_nopar:Npn #1
743     {
744       \mode_if_math:TF { ^^@ }
745       { {
746         \ctex_lua_now_x:n { tex.globaldefs = 0 }
747         \ltj@allalchar ^^@
748       } }
749     }
750   }
751   \ltjsetmathletter {#2}
752 }
753 \group_end:
754 \ctex_at_end_package:nn { unicode-math }
755 {
756   \cs_if_free:NF \um_cs_set_eq_active_char:Nw
757   { \cs_set_eq:NN \um_cs_set_eq_active_char:Nw \__ctex_ltj_um_char:Nw }
758 }

```

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

```

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

```

```

766         \tl_set:Nn \exp_not:N \lstlistlistingname
767         { \exp_not:o { \lstlistlistingname } }
768     }
769 }

```

15.4.3.3 字体切换方式

`\ctex_ljt_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。

```

\CTK@family
770 \cs_new_protected_nopar:Npn \ctex_ljt_select_font:
771 {
772     \cs_if_exist_use:cF { \l__ctex_ljt_current_font_tl }
773     { \tl_if_empty:NF \CJK@family { \__ctex_ljt_select_font_aux: } }
774 }
775 \tl_new:N \CJK@family
776 \tl_new:N \l__ctex_ljt_current_font_tl
777 \tl_set:Nn \l__ctex_ljt_current_font_tl
778 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

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

`__ctex_ljt_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。

```

779 \cs_new_protected_nopar:Npn \__ctex_ljt_select_font_aux:
780 {
781     \group_begin:
782     \tl_set_eq:NN \f@encoding \CJK@encoding
783     \tl_set_eq:NN \f@family \CJK@family
784     \__ctex_ljt_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
785     \ctex_ljt_pickup_font:
786     \group_end:
787     \font@name
788     \__ctex_ljt_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_ljt_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。

```

789     \cs_if_exist:cF { \l__ctex_ljt_current_font_tl }
790     { \__ctex_ljt_select_font_aux: }
791 }
792 \cs_new_protected_nopar:Npn \__ctex_ljt_push_fontname:n #1
793 {
794     \cs_gset_eq:NN \__ctex_ljt_save_fontname:w \font@name
795     \cs_gset_nopar:Npx \font@name {#1}
796 }
797 \cs_new_protected_nopar:Npn \__ctex_ljt_pop_fontname:
798 { \cs_gset_eq:NN \font@name \__ctex_ljt_save_fontname:w }

```

(End definition for `__ctex_ljt_select_font_aux:`.)

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

```

799 \cs_new_protected_nopar:Npn \ctex_ljt_pickup_font:
800 {
801     \exp_after:wN \cs_if_exist:NF \font@name
802     {
803         \group_begin:
804         \cs_set_eq:NN \extract@font \ctex_ljt_extract_font:
805         \cs_set_eq:NN \do@subst@correction \ctex_ljt_subst_font:
806         \define@newfont
807         \group_end:
808     }
809 }
810 \cs_new_eq:NN \pickup@font \ctex_ljt_pickup_font:

```

(End definition for \ctex_ltj_pickup_font:.)

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

```
%%%%\globaljfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont{%
  \afterassignment\ltj@jfont\directlua{luatexja.jfont.jfontdefX(false, 'yoko')}}
\protected\def\globaljfont{%
  \afterassignment\ltj@jfont\directlua{luatexja.jfont.jfontdefX(true, 'yoko')}}
\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 ',
  cstamp, '\\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。

```
811 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
812 {
813   \get@external@font
814   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
815   {
816     \tl_set:Nx \external@font
817     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
818   }
819   \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。

```
820   \font@name
821   \ctex_lua_now_x:n { font.current(tex.attribute['ltj@curjfont']) }
822   \use:c { \f@encoding + \f@family }
823   \use:c { \curr@fontshape }
824 }
```

(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。

```
825 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
826 {
827   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
828   {
829     \group_begin:
830     \tl_set_eq:NN \CJK@family \f@family
831     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
832     {
```

```

833         \cs_gset_protected_nopar:Npx \subst@correction
834         {
835             \cs_new_eq:NN
836             \exp_not:c { \l__ctex_ltj_current_font_tl }
837             \font@name
838         }
839         \group_insert_after:N \group_insert_after:N
840         \group_insert_after:N \subst@correction
841     }
842     \group_end:
843 }
844 }

```

(End definition for \ctex_ltj_subst_font:.)

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

```

845 \prg_new_conditional:Npnn \ctex_ltj_if_alterate_shape_exist:n #1 { T , F , TF }
846 {
847     \ctex_lua_now_x:n { luatexja.jfont.does_alt_set (' \luatexluaescapestring {#1} ') }
848     \prg_return_true: \else: \prg_return_false: \fi:
849 }

```

(End definition for \ctex_ltj_if_alterate_shape_exist:nTF.)

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

```

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

```

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

```

850 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
851 { #1 \ctex_lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for __ctex_ltj_patch_external_font:w.)

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

```

852 \cs_new_protected_nopar:Npn \ctex_ltj_select_alterate_font:
853 {
854     \ctex_ltj_if_alterate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
855     {
856         \ctex_lua_now_x:n
857         {
858             luatexja.jfont.output_alt_font_cmd
859             ('y', '\luatexluaescapestring { \l__ctex_ltj_current_shape_tl } ')
860         }
861         \ctex_lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size' ) }
862     }
863 }
864 \tl_new:N \l__ctex_ltj_current_shape_tl
865 \tl_set:Nn \l__ctex_ltj_current_shape_tl
866 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for \ctex_ltj_select_alterate_font:.)

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

```

867 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
868 {
869     \cs_if_exist:cF { #1/\f@size }
870     {

```

```

871     \group_begin:
872     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
873     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
874     \ctex_ltj_pickup_font:
875     \group_end:
876     \__ctex_ltj_pop_fontname:
877   }
878 }

```

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

```
879 <@@=>
```

\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。

```

880 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
881 {
882   \ltj@@getjfontnumber #1
883   \ctex_lua_now_x:n
884   {
885     luatexja.jfont.pickup_alt_font_b
886     ( \the\ltj@tempcntc, '\luatexluaescapestring {#2}' )
887   }
888 }

```

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

```
889 \ExplSyntaxOff
```

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

```

890 \def\ltj@@IsFontJapanese#1{%
891   \directlua{luatexja.jfont.is_kenc(string.match(
892     '\luatexluaescapestring{#1}', '[~/+'])}}
893 {\catcode`M=12%
894 \gdef\ltj@@mathJapaneseFonts#1M#2#3\relax{\ltj@@IsFontJapanese{#3}}
895 \let\ltj@@al@getanddefine@fonts=\getanddefine@fonts
896 \def\ltj@@ja@getanddefine@fonts#1#2{%
897   \xdef\font@name{\csname \string#2/\tf@size\endcsname}%
898   \pickup@jfont\let\textfont@name\font@name
899   \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
900   \pickup@jfont\let\scriptfont@name\font@name
901   \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
902   \pickup@jfont
903   \edef\math@fonts{\math@fonts\ltj@setpar@global%
904     \ltj@@set@stackfont#1,\textfont@name:{MJT}}%
905     \ltj@@set@stackfont#1,\scriptfont@name:{MJS}}%
906     \ltj@@set@stackfont#1,\font@name:{MJSS}}%
907   }%
908 }
909 \def\getanddefine@fonts#1#2{%
910   \ltj@tempcnta=#1\ltj@@IsFontJapanese{\string#2}%
911   \ifin@ \let\ltj@temp=\ltj@@ja@getanddefine@fonts%
912   \else \let\ltj@temp=\ltj@@al@getanddefine@fonts\fi
913   \ltj@temp{#1}{#2}%
914 }
915 \def\use@mathgroup#1#2{\relax\ifmmode
916   \math@bgroup
917   \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
918     #1\fi\ltj@tempcnta=#2 \expandafter\ltj@@mathJapaneseFonts\string#1\relax%
919     \ifin@\jfam#2\relax\else\mathgroup#2\relax\fi
920   \expandafter\math@egroup\fi}%
921 \let\@@italiccorr=\/
922 <@@=ctex_ltj>

```

923 \ExplSyntaxOn

\ctex_mono_jfm:n LuaTeX-jā 中与标点格式 plain 对应的 JFM 是 mono。
 \l__ctex_ltj_jfm_tl

```
924 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
925 {
926   \str_if_eq:nnTF {#1} { plain }
927   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
928   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
929 }
930 \tl_new:N \l__ctex_ltj_jfm_tl
931 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
932 \ctex_mono_jfm:o { \l__ctex_punct_tl }
```

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

\CJK@encoding
 __ctex_ltj_change_encoding:

在 L^AT_EX 下, LuaTeX-jā 依赖字体编码来实现特殊设置。例如上述的 \ltj@@IsFontJapanese 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。

```
933 \tl_const:Nn \CJK@encoding { LTJY3 }
934 \DeclareFontEncoding { \CJK@encoding } { } { }
935 \DeclareFontSubstitution { LTJY3 } { song } { \mddefault } { \updefault }
936 \ctex_lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
937 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
938 { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
939 \DeclareFontFamily { \CJK@encoding } { song } { }
940 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
941 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
942 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
943 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
944 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
945 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
946 { \CJK@encoding } { song } { \mddefault } { \updefault }
947 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
948 { \CJK@encoding } { song } { \bfdefault } { \updefault }
949 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
950 \jfam \c__ctex_ltj_math_fam_int
```

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

15.4.3.4 字体族的定义与使用

这是 luatexja-fontspec 中新增的一些字体选项。

```
951 \newfontfeature { CID } { cid = #1 }
952 \newfontfeature { JFM } { jfm = #1 }
953 \newfontfeature { JFM-var } { jfmvar = #1 }
```

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

```
954 \keys_define:nn { fontspec-preparse-external }
955 {
956   NoEmbed .code:n =
957   { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
958 }
959 \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。

```
960 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
961 {
962   \group_begin:
963   \clist_clear:N \l__ctex_ltj_char_range_clist
```



```

964 \seq_clear:N \l__ctex_ltj_alternate_seq
965 \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
966 \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
967 \clist_set:N \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
968 \ctex_ltj_set_alternate_family:nnF {#1} {#3}
969 {
970   \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
971   \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
972     {#1} { \l__ctex_ltj_font_options_clist }
973   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
974   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
975   \__ctex_ltj_gset_family_cs:nn {#1} {#3}
976 }
977 \group_end:
978 }
979 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
980 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
981 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
982 \clist_new:N \l__ctex_ltj_font_options_clist
983 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
984 {
985   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
986   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
987 }

```

(End definition for \ctex_ltj_set_family:nnn.)

\g__ctex_ltj_family_name_prop
 \g__ctex_ltj_family_font_name_prop
 \g__ctex_ltj_family_font_options_prop

分别保存 fontspec 设置的字体族名、字体名称和字体选项。

```

988 \prop_new:N \g__ctex_ltj_family_name_prop
989 \prop_new:N \g__ctex_ltj_family_font_name_prop
990 \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。

```

991 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
992 {
993   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
994   {
995     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
996     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
997     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
998     {
999       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1000       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1001       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1002       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1003     }
1004     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1005   }
1006 }
1007 \tl_new:N \l__ctex_ltj_tmp_tl
1008 \msg_new:nnn { ctex } { redefine-family }
1009 { 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。

```

1010 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1011 {
1012   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }

```

```

1013 {
1014   \group_begin:
1015   \__ctex_ltj_change_encoding:
1016   \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1017   \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1018   { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1019   \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1020   { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1021   \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1022   \exp_not:N \g__ctex_ltj_fontspec_family_tl
1023   \__ctex_ltj_set_alternate_family:n {#1}
1024   \group_end:
1025 }
1026 }
1027 \tl_new:N \l__ctex_ltj_base_family_tl
1028 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1029 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1030 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1031 {
1032   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1033   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1034   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1035   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1036 }
1037 \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。

```

1038 \NewDocumentCommand \CJKfamily { m }
1039 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1040 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1041 {
1042   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1043   {
1044     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1045     \selectfont
1046   }
1047   { \__ctex_ltj_family_unknown_warning:n {#1} }
1048 }
1049 \tl_new:N \l_ctex_ltj_family_tl
1050 \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。

```

1051 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1052 {
1053   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1054   { \prg_return_true: }
1055   {
1056     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1057     {
1058       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1059       \prg_return_true:
1060     }
1061     { \prg_return_false: }
1062   }
1063 }
1064 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

__ctex_ltj_family_unknown_warning:n

```

1065 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1066 {
1067   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1068   {
1069     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1070     {
1071       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1072       \msg_warning:nnn { ctex } { family-unknown } {#1}
1073     }
1074   }
1075 }
1076 \seq_new:N \g__ctex_ltj_unknown_family_seq
1077 \msg_new:nnn { ctex } { family-unknown }
1078 {
1079   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1080   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1081 }
1082 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1083 {
1084   \str_case:x:nnF {#1}
1085   {
1086     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1087     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1088     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1089   }
1090   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1091   [...] \{ ... \}
1092 }
1093 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1094 {
1095   \str_case:x:nnF {#1}
1096   {
1097     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1098     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1099     \CJKttdefault { \token_to_str:N \CJKttdefault }
1100   }
1101   {#1}
1102 }

```

(End definition for __ctex_ltj_family_unknown_warning:n.)

\ctex_ltj_fontspec:nn

```

1103 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1104 {
1105   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1106   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1107   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1108   {
1109     \int_gincr:N \g__ctex_ltj_family_int
1110     \__ctex_ltj_fontspec:xnn
1111     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1112     {#1} {#2}
1113   }
1114 }
1115 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1116 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1117 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1118 {
1119   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1120   {
1121     \cs_if_free:cF
1122     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1123     {
1124       \cs_gset_eq:cc
1125       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1126       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }

```

```

1127         \cs_gset_eq:cc
1128         { \__ctex_ltj_altername_cs:n { clear / #1 } }
1129         { \__ctex_ltj_altername_cs:n { clear / \l_ctex_ltj_family_tl } }
1130     }
1131     \bool_set_false:N \l__ctex_ltj_add_altername_bool
1132 }
1133 \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1134 \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1135 \ctex_ltj_switch_family:n {#1}
1136 }
1137 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1138 \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
1139 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1140 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1141 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1142 {
1143     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1144     {#1} \l__ctex_ltj_tmp_tl
1145     {
1146         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1147         {#1} \l__ctex_ltj_font_options_clist
1148         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1149         \bool_set_true:N \l__ctex_ltj_add_altername_bool
1150         \ctex_ltj_fontspec:xx
1151         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1152         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1153     }
1154     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1155 }
1156 \bool_new:N \l__ctex_ltj_add_altername_bool
1157 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1158 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1159 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1160 {
1161     \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1162     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1163 }

```

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

```

\setCJKfamilyfont
\newCJKfontfamily
\CJKfontspec
\addCJKfontfeatures
1164 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
1165 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1166 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
1167 {
1168     \tl_set:Nx \l__ctex_ltj_tmp_tl
1169     { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1170     \cs_new_protected_nopar:Npx #2
1171     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1172     \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1173 }
1174 \NewDocumentCommand \CJKfontspec { O { } m }
1175 {
1176     \ctex_ltj_fontspec:xx {#1} {#2}
1177     \tex_ignorespaces:D
1178 }
1179 \NewDocumentCommand \addCJKfontfeatures { m }
1180 {
1181     \ctex_ltj_add_font_features:x {#1}
1182     \tex_ignorespaces:D
1183 }

```

```
1184 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
```

(End definition for \setCJKfamilyfont and others.)

```
\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1185 \NewDocumentCommand \setCJKmainfont { 0 { } m }
1186 {
1187   \ctex_ljtj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1188   \normalfont
1189 }
1190 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1191 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1192 {
1193   \ctex_ljtj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1194   \normalfont
1195 }
1196 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1197 {
1198   \ctex_ljtj_set_family:xxx { \CJKttdefault } {#1} {#2}
1199   \normalfont
1200 }
1201 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1202 { \ctex_ljtj_set_family:xxx { \c__ctex_ljtj_math_tl } {#1} {#2} }
1203 \NewDocumentCommand \defaultCJKfontfeatures { m }
1204 { \clist_gset:Nn \g__ctex_ljtj_default_features_clist {#1} }
1205 \clist_new:N \g__ctex_ljtj_default_features_clist
1206 \@onlypreamble \setCJKmainfont
1207 \@onlypreamble \setCJKsansfont
1208 \@onlypreamble \setCJKmonofont
1209 \@onlypreamble \setCJKmathfont
1210 \@onlypreamble \setCJKromanfont
1211 \@onlypreamble \defaultCJKfontfeatures
```

(End definition for \setCJKmainfont and others.)

```
1212 \tl_if_exist:NF \CJKfamilydefault
1213 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1214 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1215 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1216 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1217 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
1218 { \ctex_patch_failure:N \rmfamily }
1219 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }
1220 { \ctex_patch_failure:N \sffamily }
1221 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
1222 { \ctex_patch_failure:N \ttfamily }
1223 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
1224 { \cs_set_eq:NN \reset@font \normalfont }
1225 { \ctex_patch_failure:N \normalfont }
```

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

```
1226 \cs_new_protected_nopar:Npn \ctex_ljtj_ensure_default_family:
1227 {
1228   \prop_if_empty:NF \g__ctex_ljtj_family_font_name_prop
1229   {
1230     \ctex_ljtj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ljtj_tmpa_tl
1231     {
1232       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1233       { \use:n }
1234       {
1235         \ctex_ljtj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ljtj_tmpa_tl
1236         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1237         { \use:n }
1238       }
1239     }
1240   }
```

```

1240             \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1241             {
1242                 \prop_map_break:n
1243                 { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1244             }
1245         }
1246     }
1247     \normalfont
1248     \ctex_ltj_update_mathfont:
1249 }
1250 }

```

(End definition for \ctex_ltj_ensure_default_family:.)

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

```

1251 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1252 {
1253     \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1254     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1255     {
1256         \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1257         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1258     }
1259 }
1260 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1261 {
1262     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1263     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1264     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1265     \cs_if_free:cTF
1266     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1267     {
1268         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1269         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1270     }
1271     {
1272         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1273         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1274     }
1275 }

```

(End definition for \ctex_ltj_update_mathfont:.)

15.4.3.5 替代字体的设置

AlternateFont 设置替代字体的选项。

CharRange

```

1276 \keys_define:nn { ctex_ltj / fontspec }
1277 {
1278     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1279     AlternateFont .value_required: ,
1280     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1281     CharRange .value_required:
1282 }

```

(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。

```

1283 \group_begin:
1284 \char_set_catcode_active:N \
1285 \char_set_lccode:nn { `\/ } { `| }
1286 \tex_lowercase:D

```

```

1287 {
1288   \group_end:
1289   \cs_new_protected:Npn \ctex_ljt_set_alternate_seq:n #1
1290   {
1291     \clist_if_empty:NT \l__ctex_ljt_char_range_clist
1292     {
1293       \tl_set:Nn \l__ctex_ljt_tmp_tl {#1}
1294       \tl_replace_all:Nnn \l__ctex_ljt_tmp_tl { // } { || }
1295       \seq_set_split:NnV \l__ctex_ljt_tmp_seq { || } \l__ctex_ljt_tmp_tl
1296       \seq_set_filter:Nnn \l__ctex_ljt_tmp_seq \l__ctex_ljt_tmp_seq
1297       { ! \tl_if_blank_p:n { ##1 } }
1298       \seq_concat:Nnn \l__ctex_ljt_alternate_seq
1299       \l__ctex_ljt_alternate_seq \l__ctex_ljt_tmp_seq
1300     }
1301   }
1302 }
1303 \seq_new:N \l__ctex_ljt_tmp_seq
1304 \seq_new:N \l__ctex_ljt_alternate_seq

```

(End definition for \ctex_ljt_set_alternate_seq:n)

\ctex_ljt_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。

```

1305 \cs_new_protected_nopar:Npn \ctex_ljt_set_alternate_family:nnF #1#2#3
1306 {
1307   \clist_if_empty:NTF \l__ctex_ljt_char_range_clist
1308   {
1309     \__ctex_ljt_check_family:n {#1}
1310     \seq_if_empty:NF \l__ctex_ljt_alternate_seq
1311     { \ctex_ljt_save_alternate_seq:cn { \__ctex_ljt_alternate_cs:n {#1} } {#2} }
1312     #3
1313   }
1314   { \ctex_ljt_set_alternate_family:nn {#1} {#2} }
1315 }

```

(End definition for \ctex_ljt_set_alternate_family:nnF)

\ctex_ljt_save_alternate_seq:Nn 保存由 AlternateFont 设置的替代字体序列。

```

\ctex_ljt_save_alternate_seq:Nnnwn
1316 \cs_new_protected_nopar:Npn \ctex_ljt_save_alternate_seq:Nn #1#2
1317 {
1318   \seq_map_inline:Nn \l__ctex_ljt_alternate_seq
1319   { \ctex_ljt_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1320 }
1321 \cs_generate_variant:Nn \ctex_ljt_save_alternate_seq:Nn { c }
1322 \NewDocumentCommand \ctex_ljt_save_alternate_seq:Nnnwnw
1323 { m m m +O{ } m u{ \q_stop } }
1324 {
1325   \clist_set:Nn \l__ctex_ljt_char_range_clist {#3}
1326   \clist_set:Nn \l__ctex_ljt_alternate_options_clist {#4}
1327   \__ctex_ljt_use_global_options:N \l__ctex_ljt_alternate_options_clist
1328   \tl_if_blank:nTF {#5}
1329   { \tl_set:Nn \l__ctex_ljt_tmp_tl {#2} }
1330   {
1331     \tl_set:Nn \l__ctex_ljt_tmp_tl {#5}
1332     \tl_replace_all:Nnn \l__ctex_ljt_tmp_tl { * } {#2}
1333   }
1334   \use:x
1335   {
1336     \ctex_ljt_save_alternate_family:Nnnn \exp_not:N #1
1337     { \exp_not:o { \l__ctex_ljt_char_range_clist } }
1338     { \exp_not:o { \l__ctex_ljt_alternate_options_clist } }
1339     { \exp_not:o { \l__ctex_ljt_tmp_tl } }
1340   }
1341 }
1342 \clist_new:N \l__ctex_ljt_alternate_options_clist

```

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

\ctex_ltj_set_altername_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。

```

1343 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_family:nn #1#2
1344 {
1345   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1346   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1347   \ctex_ltj_set_altername_family:coonn
1348   { \__ctex_ltj_altername_cs:n {#1} }
1349   { \l__ctex_ltj_char_range_clist }
1350   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1351 }
1352 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_family:Nnnnn #1#2#3#4#5
1353 {
1354   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1355   { \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} }
1356   \ctex_ltj_save_altername_family:Nnnn #1 {#2} {#3} {#4}
1357 }
1358 \cs_generate_variant:Nn \ctex_ltj_set_altername_family:Nnnnn { coo }

```

(End definition for \ctex_ltj_set_altername_family:nn.)

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

```

1359 \cs_new_protected_nopar:Npn \ctex_ltj_save_altername_family:Nnnn #1#2#3#4
1360 {
1361   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1362   \cs_gset_protected_nopar:Npx #1
1363   { \exp_not:o { #1 \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} } }
1364 }

```

(End definition for \ctex_ltj_save_altername_family:Nnnn.)

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

```

1365 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_family:nnn #1#2#3
1366 {
1367   \group_begin:
1368   \__ctex_ltj_change_encoding:
1369   \cs_set_eq:NN \CJKfamily \use_none:n
1370   \ctex_ltj_swap_cs:NN
1371   \DeclareFontShape@ \ctex_ltj_declare_altername_shape:nnnnnn
1372   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1373   \fontspec_set_family:Nnn \l__ctex_ltj_altername_family_tl {#2} {#3}
1374   \group_end:
1375 }
1376 \tl_new:N \l__ctex_ltj_altername_family_tl

```

(End definition for \ctex_ltj_set_altername_family:nnn.)

\ctex_ltj_swap_cs:NN 交换两个控制序列的意义。

```

1377 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1378 {
1379   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1380   \cs_set_eq:NN #1 #2
1381   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1382   \cs_undefine:N \__ctex_ltj_tmp:w
1383 }

```

(End definition for \ctex_ltj_swap_cs:NN.)

LTJFONTUID
 _ctex_ltj_update_family_uid:N

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

```
1384 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1385 \cs_new_protected_nopar:Npn \_ctex_ltj_update_family_uid:N #1
1386 {
1387   \int_gincr:N \g__ctex_ltj_family_int
1388   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1389 }
1390 \int_new:N \g__ctex_ltj_family_int
```

(End definition for LTJFONTUID and _ctex_ltj_update_family_uid:N.)

\ctex_ltj_declare_alter_nate_shape:nnnnnn

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

```
1391 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1392 {
1393   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1394   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1395   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1396   { \l_fontspec_family_tl } {#3} {#4}
1397 }
```

(End definition for \ctex_ltj_declare_alter_nate_shape:nnnnnn.)

\ctex_ltj_set_alter_nate_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。

```
1398 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1399 {
1400   \clist_map_inline:Nn #1
1401   {
1402     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1403     {
1404       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1405       \l__ctex_ltj_char_range_tl
1406     }
1407     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1408   }
1409   \_ctex_ltj_save_alter_nate_shape:cnn
1410   { \_ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1411   { luatexja.jfont.clear_alt_font_latex }
1412   { '\luatexluaescapestring { \CJK@encoding/#2/#3/#4 }' }
1413 }
```

(End definition for \ctex_ltj_set_alter_nate_shape:Nnnnnnn.)

\ctex_ltj_set_alter_nate_shape:nnn

我们使用 `->` 而不是像 LuaTeX-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。

```
1414 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1415 { m m > { \SplitArgument { \c_one } { -> } } m }
1416 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1417 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1418 {
1419   \ctex_ltj_set_alter_nate_shape:n
1420   {
1421     \IfNoValueTF {#4}
1422     { \int_eval:n {#3} , \int_eval:n {#3} , }
1423     {
1424       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
```

```

1425         \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1426     }
1427     '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1428     '\luatexluaescapestring { \CJK@encoding/#1 }'
1429 }
1430 }
1431 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1432 {
1433     \ctex_lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1434     \__ctex_ltj_save_alternate_shape:cnn
1435     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1436     { luatexja.jfont.set_alt_font_latex } {#1}
1437 }

```

(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。

```

1438 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1439 {
1440     \tl_map_inline:Nn #3
1441     {
1442         \ctex_ltj_set_alternate_shape:n
1443         {
1444             ##1 ,
1445             '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1446             '\luatexluaescapestring { \CJK@encoding/#1 }'
1447         }
1448     }
1449 }

```

(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。

```

1450 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1451 {
1452     \group_begin:
1453     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1454     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1455     \cs_set_eq:NN \luatexluaescapestring \scan_stop:
1456     \cs_gset_protected_nopar:Npx #1
1457     { \exp_not:o {#1} \exp_not:N \ctex_lua_now_x:n { #2 ( #3 ) } }
1458     \group_end:
1459 }
1460 \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

```

1461 \keys_define:nn { ctex }
1462 {
1463     clearalternatefont .code:n =
1464     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1465     resetalternatefont .code:n =
1466     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1467     clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1468     resetalternatefont .default:n = \l_ctex_ltj_family_tl
1469 }
1470 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1471 {
1472     \group_begin:
1473     \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl

```

```

1474     {
1475         \cs_if_exist_use:cT { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1476         {
1477             \prop_gput:Nno \g__ctex_ltj_reset_alterate_prop
1478             {#1} { \l__ctex_ltj_base_family_tl }
1479             \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1480             \selectfont
1481         }
1482     }
1483     { \__ctex_ltj_family_unknown_warning:n {#1} }
1484 \group_end:
1485 }
1486 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alterate_font:n #1
1487 {
1488     \group_begin:
1489     \prop_gpop:NnNT \g__ctex_ltj_reset_alterate_prop {#1} \CJK@family
1490     {
1491         \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1492         \use:c { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1493         \selectfont
1494     }
1495     \group_end:
1496 }
1497 \prop_new:N \g__ctex_ltj_reset_alterate_prop
1498 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalteratefont` and `resetalteratefont`. These functions are documented on page 25.)

declarecharrange 预先声明字符范围。

```

1499 \keys_define:nn { ctex }
1500 {
1501     declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1502     declarecharrange .value_required:
1503 }
1504 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1505 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1506 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1507 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1508 { \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`

```

1509 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1510 {
1511     \tl_clear:N \l__ctex_ltj_char_range_tl
1512     \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1513     \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1514     \ctex_ltj_def_char_range_key:n {#1}
1515     \tl_clear:N \l__ctex_ltj_char_range_tl
1516 }
1517 \tl_new:N \l__ctex_ltj_char_range_tl
1518 \prop_new:N \g__ctex_ltj_char_range_prop

```

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

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

```

1519 \NewDocumentCommand \ctex_ltj_save_char_range:n
1520 { > { \SplitArgument { \c_one } { -> } } m }
1521 { \ctex_ltj_save_char_range:nn #1 }
1522 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1523 {
1524     \tl_put_right:Nx \l__ctex_ltj_char_range_tl

```

```

1525     { {
1526         \IfNoValueTF {#2}
1527         { \int_eval:n {#1} , \int_eval:n {#1} }
1528         {
1529             \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1530             \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1531         }
1532     } }
1533 }

```

(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。

```

1534 \cs_new_protected_nopar:Npn \ctex_ljt_def_char_range_key:n #1
1535 {
1536     \keys_if_exist:nnF { ctex_ljt / fontspec } {#1}
1537     {
1538         \keys_define:nn { ctex_ljt / fontspec }
1539         { #1 .code:n = \ctex_ljt_char_range_key:nn {#1} { ##1 } }
1540     }
1541 }

```

(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。

```

1542 \cs_new_protected:Npn \ctex_ljt_char_range_key:nn #1#2
1543 {
1544     \tl_if_blank:nTF {#2}
1545     { \tl_set:Nn \l__ctex_ljt_char_range_clist {#1} }
1546     {
1547         \clist_if_empty:NT \l__ctex_ljt_char_range_clist
1548         {
1549             \tl_set:Nn \l__ctex_ljt_tmp_tl { {#1} }
1550             \__ctex_ljt_char_range_parse_feature:w #2 \q_stop
1551         }
1552     }
1553 }

```

(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]。

```

1554 \NewDocumentCommand \__ctex_ljt_char_range_parse_feature:w
1555 { +o o u { \q_stop } }
1556 {
1557     \exp_args:NNf \tl_put_right:Nn \l__ctex_ljt_tmp_tl
1558     {
1559         \IfNoValueTF {#1} { {#3} }
1560         {
1561             \IfNoValueTF {#2}
1562             { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] { #3 } } }
1563             { [ { #1 } ] { [ #2 ] } }
1564         }
1565     }
1566     \seq_put_right:No \l__ctex_ljt_alternate_seq { \l__ctex_ljt_tmp_tl }
1567 }

```

(End definition for __ctex_ljt_char_range_parse_feature:w.)

15.4.3.6 其它设置

`\ctex_ljtj_update_xkanjiskip:` `\l__ctex_ljtj_xkanjiskip_skip` `\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。

```

1568 \cs_new_protected_nopar:Npn \ctex_ljtj_update_xkanjiskip:
1569 {
1570   \skip_if_eq:nnT
1571     { \ltjgetparameter { xkanjiskip } } { \l__ctex_ljtj_xkanjiskip_skip }
1572   {
1573     \skip_set:Nn \l__ctex_ljtj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }
1574     \ltjsetxkanjiskip \l__ctex_ljtj_xkanjiskip_skip
1575   }
1576 }

```

注意,此时还没有实际设置字体,所以 `\zw` 是 `0pt`。

```

1577 \skip_new:N \l__ctex_ljtj_xkanjiskip_skip
1578 \skip_set:Nn \l__ctex_ljtj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }

```

(End definition for `\ctex_ljtj_update_xkanjiskip:` and `\l__ctex_ljtj_xkanjiskip_skip.`)

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

```

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

1579 \AtBeginDocument
1580 {
1581   \ctex_appto_cmd:NnTF \verbatim@font { \CTEX@verbatim@font@hook }
1582   { } { \ctex_patch_failure:N \verbatim@font }
1583 }
1584 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1585 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

1586 <@@=ctex>

1587 </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。

```

1588 <*pdfTeX||luatex>
1589 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1590 {
1591   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1592   {
1593     \group_begin:
1594     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1595     \tl_gset:Nx \CJKfamilydefault

```

```

1596         {
1597             \str_case:onF { \familydefault }
1598             {
1599                 { \rmdefault } { \exp_not:N \CJKrmdefault }
1600                 { \sfdefault } { \exp_not:N \CJKsfdefault }
1601                 { \ttdefault } { \exp_not:N \CJKttdefault }
1602             }
1603             { \CJKfamilydefault }
1604         }
1605     \group_end:
1606 }

```

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

```

1607 <*luatex>
1608     \ctex_ltj_ensure_default_family:
1609 </luatex>
1610 }
1611 \AtEndPreamble { \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。

```

1612 \tl_new:N \l__ctex_family_default_init_tl
1613 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1614 \tl_set:Nx \l__ctex_family_default_init_tl
1615 {
1616     \exp_not:N \__ctex_family_default_wrap:n
1617     { \exp_not:o { \CJKfamilydefault } }
1618 }
1619 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1620 </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 Lua_{TeX} 下直接用调用 os.name 来判断。

```

1621 <*luatex>
1622 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1623 {
1624     \tl_gset:Nx \g__ctex_fontset_tl
1625     {
1626         \ctex_lua_now_x:n
1627         {
1628             if ~ os.name == 'windows' then ~
1629                 tex.sprint ( 'windows' )
1630             elseif ~ os.name == 'macosx' then ~
1631                 tex.sprint ( 'mac' )
1632             else ~
1633                 tex.sprint ( 'fandol' )
1634             end
1635         }
1636     }
1637 }
1638 </luatex>

```

pdf_{TeX} 和 X_{TeX} 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!

```

1639 <*xetex|pdfTeX>

```

```

1640 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1641 {
1642   \file_if_exist:nTF { /dev/null }
1643   {
1644     \file_if_exist:nTF { nul: }
1645     {
1646       \file_if_exist:nTF { \c__ctex_engine_file_tl }
1647       { \ctex_if_macosx:TF { mac } { windows } }
1648       { \ctex_if_macosx:TF { mac } { fandol } }
1649     }
1650     { \ctex_if_macosx:TF { mac } { fandol } }
1651   }
1652   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1653 }
1654 \tex_uppercase:D \exp_after:wN
1655 {
1656   \exp_after:wN \tl_const:Nn \exp_after:wN \c__ctex_engine_file_tl
1657   \exp_after:wN { \g_file_current_name_tl }
1658 }

```

(End definition for \ctex_detected_platform:.)

\ctex_if_macosx:TF 以 /mach_kernel 为特征文件判断 Mac OS X。

```

1659 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1660 {
1661   \file_if_exist:nTF { \c__ctex_macosx_file_tl }
1662   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1663   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1664 }
1665 \tl_const:Nn \c__ctex_macosx_file_tl { /mach_kernel }
1666 </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_hypersetup:n 如果已经载入 hyperref 宏包, 则直接使用其定义设置选项; 否则 \ctex_hypersetup:n 的效果与 \PassOptionsToPackage 一致, 只传递宏包参数。如果用户不载入 hyperref 宏包, 相关参数即被丢弃。

```

1667 <*class|style>
1668 \@ifpackageloaded { hyperref }
1669 {
1670   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1671   { \hypersetup {#1} }
1672 }
1673 {
1674   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1675   { \PassOptionsToPackage {#1} { hyperref } }
1676 }
1677 \ctex_hypersetup:n { colorlinks = true }
1678 </class|style>

```

(End definition for \ctex_hypersetup:n.)

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

```

1679 <*pdftex>
1680 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1681 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1682 {
1683   \ctex_hypersetup:n { CJKbookmarks = true }
1684   \ctex_if_pdfmode:TF
1685   { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1686   {
1687     \ctex_at_end_package:nn { hyperref }
1688     {
1689       \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1690       {
1691         \AtBeginShipoutFirst
1692         { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1693       }
1694       { \RequirePackage { xCJK2uni } }
1695     }
1696   }
1697 }
1698 { \ctex_hypersetup:n { pdfencoding = unicode } }
1699 </pdftex>

```

在 XeTeX 下, `hyperref` 在处理带有非 ASCII 字符和 `\%` 的书签时有问题¹⁵。事实上, `hyperref` 在驱动文件 `hxdetex.def` 中设置了 `\Hy@unicodetrue`, 从而书签总是会被 `\HyPsd@ConvertToUnicode` 转化成 UTF-16BE 编码的形式(抄录自 `\pdfstringdef` 的定义):

```

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

```

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

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

XeTeX 的维护者 Khaled Hosny 已经注意到了这个问题¹⁷。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 CTeX 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 XeTeX 下更保险的 `pdfencoding` 选项来设置。为了与 XeTeX 下的行为一致(使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 LuaTeX 下也启用这个选项。

```

1700 <*xetex|luatex>
1701 \ctex_hypersetup:n { pdfencoding = unicode }
1702 </xetex|luatex>

1703 <*pdftex|xetex|luatex>

```

¹⁵<https://github.com/CTeX-org/ctex-kit/issues/39>

¹⁶<http://project.ktug.org/dvipdfmx/mailman/dvipdfmx/2009-December/000153.html>

¹⁷<http://tug.org/pipermail/tex-live/2013-December/034613.html>

15.4.7 CJKfntef、xeCJKfntef 相关设置

CT_EX 套件对 pdf_TE_X 与 X_EL_AT_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。

```

1704 <pdfTeX>\RequirePackage { CJKfntef } \normalem
1705 <xetex>\RequirePackage { xeCJKfntef }
1706 <*luatex>
1707 \msg_new:nnn { ctex } { fntef-not-available }
1708 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1709 \msg_warning:nn { ctex } { fntef-not-available }
1710 </luatex>
1711 \clist_map_inline:nn
1712 { underdot , underline , underdblline , underwave , sout , xout }
1713 <*pdfTeX|xetex>
1714 {
1715 <pdfTeX> \tl_clear:c { CJK#1color }
1716 <xetex> \keys_set:nn { xeCJK / options } { #1 / format = { } }
1717 \cs_new_protected_nopar:cpx { CTEX#1 }
1718 {
1719 \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1720 { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1721 \exp_not:c { CJK#1 }
1722 }
1723 }
1724 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1725 {
1726 \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1727 { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1728 \CJKfilltwosides
1729 }
1730 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1731 </pdfTeX|xetex>
1732 <*luatex>
1733 { \cs_new_eq:cN { CTEX#1 } \use:n }
1734 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1735 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1736 </luatex>
1737 <*pdfTeX>
1738 \clist_map_inline:nn
1739 {
1740 underdotbasesep , underdotsep , underlinebasesep ,
1741 underlinesep , underdbllinesep , underdbllinebasesep ,
1742 underwavebasesep , underwavesep , southeight ,
1743 underdotcolor , underwavecolor , underlinecolor ,
1744 underdbllinecolor , soutcolor , xoutcolor
1745 }
1746 {
1747 \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1748 \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1749 }
1750 </pdfTeX>

```

15.4.8 \ccwd 的更新

\ctex_update_ccwd:

\ccwd

```

1751 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1752 <*pdfTeX|xetex>
1753 {
1754 \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1755 \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }

```

```

1756 }
1757 </pdf|tex|xetex>
1758 <*luatex>
1759 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1760 </luatex>
1761 \dim_new:N \ccwd

```

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

\ctex_update_ccglue: 更新字间距。

```

1762 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1763 <*pdf|tex>
1764 {
1765   \cs_set_protected_nopar:Npn \CJKglue
1766   { \skip_horizontal:N \l__ctex_ccglue_skip }
1767 }
1768 </pdf|tex>
1769 <*xetex>
1770 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1771 </xetex>
1772 <*luatex>
1773 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1774 </luatex>
1775 \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

```

1776 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1777 {
1778   <*pdf|tex|xetex>
1779   \if_meaning:w \CJKglue \__ctex_ccglue:
1780   \prg_return_false: \else: \prg_return_true: \fi:
1781 </pdf|tex|xetex>
1782 <*luatex>
1783   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1784   { \prg_return_false: } { \prg_return_true: }
1785 </luatex>
1786 }

```

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

```

1787 <*pdf|tex|xetex>
1788 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1789 </pdf|tex|xetex>

```

(End definition for \ctex_if_ccglue_touched:TF.)

\ctex_update_em_unit: 将当前汉字的宽度保存到 \ccwd 中备用。不采用 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 pdf_{TeX} 和 Xe_{TeX} 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 Xe_{TeX} 可以用 \fontcharwd 来改进。而在 pdf_{TeX} 下, 若使用 zhmetrics 技术, 所有的汉字共享同一个 .tfm, \fontcharwd 也就没有意义。在 Lua_{TeX} 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。

```

1790 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1791 <pdf|tex|xetex> { \dim_set:Nn \ccwd { \f@size \p@ } }
1792 <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`。 `everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。

```

1793 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1794 {
1795   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1796   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1797 }
1798 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1799 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1800 \use:n
1801 {
1802   \ExplSyntaxOff
1803   \ctex_preto_cmd:NnTF \selectfont { \CTEX@selectfont@hook }
1804   {
1805     \tl_put_left:Nn \@EverySelectfont@Init
1806     { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1807   }
1808   { \ctex_patch_failure:N \selectfont }
1809 }
1810 \ExplSyntaxOn

```

(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。

```

1811 <*pdfTeX>
1812 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1813 </pdfTeX>
1814 <*xetex|luatex>
1815 \EverySelectfont { \CTEX@selectfont@hook }
1816 </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。

```

1817 <*luatex>
1818 \ctex_add_to_selectfont:n
1819 {
1820   \ctex_ltj_select_font:
1821   \ctex_ltj_select_alternate_font:
1822   \ctex_ltj_update_xkanjiskip:
1823 }
1824 \tl_set:Nn \CJK@family { song } \selectfont
1825 \tl_clear:N \CJK@family
1826 </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。

```

1827 \keys_define:nn { ctex }
1828 {
1829   <*pdfTeX|xetex>
1830     space .choice: ,
1831     space / true .code:n =
1832     <pdfTeX> { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
1833     <xetex> { \xeCJKsetup { CJKspace = true } } ,
1834     space / auto .code:n =
1835     <pdfTeX> { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
1836     <xetex> { \xeCJKsetup { CJKspace = false } } ,
1837     space / false .code:n =
1838     <pdfTeX> { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,

```

```

1839 <xetex>      { \xeCJKsetup { CJKspace = false } } ,
1840     space .default:n = { true } ,
1841     space .initial:n = { auto }
1842 </pdfTeX|xetex>
1843 <*luatex>
1844     space .code:n =
1845     { \msg_warning:nn { ctex } { invalid-option } }
1846 </luatex>
1847 }

```

(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。

```

1848 \keys_define:nn { ctex }
1849 {
1850     punct .code:n =
1851     {
1852         \tl_set:Nx \l__ctex_punct_tl { #1 }
1853 <pdfTeX>      \punctstyle { \l__ctex_punct_tl }
1854 <xetex>      \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
1855 <luatex>     \ctex_mono_jfm:o { \l__ctex_punct_tl }
1856     } ,
1857     punct .default:n = { quanjiao } ,
1858 }

```

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

```

1859 </pdfTeX|xetex|luatex>
1860 <*class|style>

```

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。

```

1861 \pdfTeX_if_engine:TF
1862 {
1863     \tl_set:Nx \l__ctex_encoding_tl { \l__ctex_encoding_tl }
1864     \ctex_file_input:n { ctex-engine-pdfTeX.def }
1865 }
1866 {
1867     \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
1868     \xetex_if_engine:TF
1869     { \ctex_file_input:n { ctex-engine-xetex.def } }
1870     { \ctex_file_input:n { ctex-engine-luatex.def } }
1871 }

```

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。

```

1872 \keys_define:nn { ctex }
1873 {
1874     autoindent .choice: ,
1875     autoindent .default:n = { true } ,
1876     autoindent / true .code:n =
1877     {
1878         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
1879         \ctex_select_size:
1880     } ,
1881     autoindent / false .code:n =
1882     { \tl_clear:N \l__ctex_autoindent_tl } ,
1883     autoindent / unknown .code:n =

```

```

1884     {
1885         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
1886         \ctex_select_size:
1887     }
1888 }

```

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

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

1889 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
1890 \cs_new_protected_nopar:Npn \ctex_select_size:
1891 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

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

\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。

```

1892 \cs_new_protected_nopar:Npn \ctex_update_size:
1893 {
1894     \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
1895     {
1896         \ctex_update_stretch:
1897         \ctex_update_parindent:
1898     }
1899     { \ctex_update_ziju: }
1900 }
1901 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
1902 \tl_new:N \l__ctex_ziju_tl
1903 \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。

```

1904 \ctex_add_to_selectfont:n
1905 { \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。

```

1906 \keys_define:nn { ctex }
1907 {
1908     linestretch .code:n =
1909     {
1910         \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
1911         \ctex_select_size:
1912     } ,
1913     linestretch .value_required:
1914 }
1915 \tl_new:N \l__ctex_line_stretch_tl
1916 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

(End definition for linestretch. This function is documented on page 12.)

\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。

```

1917 \cs_new_protected_nopar:Npn \ctex_update_stretch:
1918 {
1919   \ctex_update_em_unit:
1920   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
1921   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
1922   {
1923     \ctex_if_ccglue_touched:TF
1924     { \ctex_update_ccwd: }
1925     {
1926       \dim_set:Nn \l__ctex_tmp_dim
1927       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
1928       \skip_set:Nn \l__ctex_ccglue_skip
1929       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
1930       \ctex_update_ccglue:
1931     }
1932   }
1933   {
1934     \int_set:Nn \l__ctex_tmp_int
1935     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
1936     \tex_divide:D \l__ctex_tmp_int \ccwd
1937     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
1938     {
1939       \skip_set:Nn \l__ctex_ccglue_skip
1940       {
1941         \c_zero_dim plus \dim_eval:n
1942         {
1943           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
1944           \l__ctex_tmp_int
1945         }
1946       }
1947     }
1948     { \skip_zero:N \l__ctex_ccglue_skip }
1949     \ctex_update_ccglue:
1950   }
1951 }

```

(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。

```

1952 \cs_new_protected_nopar:Npn \ctex_update_parindent:
1953 {
1954   \tl_if_empty:NF \l__ctex_autoindent_tl
1955   {
1956     \dim_compare:nNnF \parindent = \c_zero_dim
1957     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
1958   }
1959 }

```

(End definition for \ctex_update_parindent:.)

\ziju 若参数为 0, 则恢复正常间距。

```

1960 \NewDocumentCommand \ziju { m }
1961 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
1962 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
1963 {
1964   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
1965   \ctex_select_size:
1966 }

```

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

`\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。

```

1967 \cs_new_protected_nopar:Npn \ctex_update_ziju:
1968 {
1969   \ctex_update_em_unit:
1970   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
1971   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
1972   \dim_compare:nNnTF \ccwd > \c_zero_dim

```

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

```

1973   {
1974     \dim_set:Nn \l__ctex_tmp_dim
1975       { \linewidth - \ccwd + \l__ctex_ziju_dim }
1976     \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
1977     \tex_divide:D \l__ctex_tmp_int \ccwd
1978     \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。

```

1979     \dim_compare:nNnF \parindent = \c_zero_dim
1980     {
1981       \int_compare:nNnF \l__ctex_tmp_int < \c_three
1982       { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
1983     }
1984     \skip_set:Nn \l__ctex_ccglue_skip
1985     {
1986       \l__ctex_ziju_dim
1987       plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
1988       minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
1989       { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
1990     }
1991   }
1992   { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
1993   \ctex_update_ccglue:

```

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

```

1994   \ctex_update_parindent:
1995 }
1996 \dim_new:N \l__ctex_ziju_dim

```

(End definition for `\ctex_update_ziju:`)

`\CTEXindent` 过时命令。
`\CTEXnoindent`

```

1997 \NewDocumentCommand \CTEXindent { }
1998 {
1999   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2000   { \parindent is~ set~ to~ 2\ccwd. }
2001   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2002 }
2003 \NewDocumentCommand \CTEXnoindent { }
2004 {
2005   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2006   { \parindent is~ set~ to~ 0pt. }
2007   \dim_zero:N \parindent
2008 }

```

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

```

2009 \bool_if:NT \l__ctex_indent_bool
2010 { \RequirePackage { indentfirst } }

```

15.6 中文数字与日期

```
2011 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2012 \RequirePackage { zhnumber }
```

\chinese

```
2013 \cs_new_eq:NN \chinese \zhnum
2014 \cs_new_eq:NN \Chinese \chinese
2015 \cs_new_eq:NN \CTEXcounter \use_none:n
```

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

\CTEXnumber

\CTEXdigits

```
2016 \NewDocumentCommand \CTEXnumber { m m }
2017 { \protected@edef #1 { \zhnumber {#2} } }
2018 \NewDocumentCommand \CTEXdigits { m m }
2019 { \protected@edef #1 { \zhdigits {#2} } }
```

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

today

```
2020 \cs_set_eq:NN \CTEX@todayold \today
2021 \keys_define:nn { ctex }
2022 {
2023   today .choice: ,
2024   today / old .code:n =
2025     { \cs_set_eq:NN \today \CTEX@todayold } ,
2026   today / small .code:n =
2027     {
2028       \cs_set_eq:NN \today \zhtoday
2029       \zhnumsetup { time = Arabic }
2030     } ,
2031   today / big .code:n =
2032     {
2033       \cs_set_eq:NN \today \zhtoday
2034       \zhnumsetup { time = Chinese }
2035     } ,
2036   today / unknown .code:n =
2037     { \msg_error:nxx { ctex } { today-undef } {#1} }
2038 }
2039 \msg_new:nnnn { ctex } { today-undef }
2040 { Today~format~#1'~is~undefined. }
2041 { Available~today~formats~are~`old',~`small',~and~`big'. }
```

(End definition for today. This function is documented on page 12.)

15.7 其它中文标题定义

```
2042 \keys_define:nn { ctex }
2043 {
2044   contentsname .tl_set:N = \contentsname ,
2045   listfigurename .tl_set:N = \listfigurename ,
2046   listtablename .tl_set:N = \listtablename ,
2047   figurename .tl_set:N = \figurename ,
2048   tablename .tl_set:N = \tablename ,
2049   abstractname .tl_set:N = \abstractname ,
2050   indexname .tl_set:N = \indexname ,
2051   appendixname .tl_set:N = \appendixname ,
2052   <article> bibname .tl_set:N = \refname
2053   <book|report> bibname .tl_set:N = \bibname
2054 }
2055 <*style>
```



```

2056 \msg_new:nnn { ctex } { ctexbibname }
2057 {
2058   Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.\\
2059   The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2060 }
2061 \tl_if_exist:NTF \bibname
2062 { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2063 {
2064   \tl_if_exist:NTF \refname
2065   { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2066   {
2067     \msg_warning:nn { ctex } { ctexbibname }
2068     \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2069   }
2070 }
2071 </style>
2072 </class|style>

```

15.8 中文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```

2073 <*class|heading>

```

15.8.1 定义标题格式选项

```
\c__ctex_headings_seq
```

```

2074 \seq_new:N \c__ctex_headings_seq
2075 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2076 {
2077   <article>    part , section , subsection , subsubsection ,
2078   <book|report> part , chapter , section , subsection , subsubsection ,
2079   paragraph , subparagraph
2080 }

```

(End definition for \c__ctex_headings_seq.)

```
\__ctex_initial_heading:n
```

```

2081 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2082 {
2083   \tl_new:c { CTEX@pre#1 }
2084   \tl_new:c { CTEX@post#1 }
2085   \tl_const:cx { CTEX@the#1 }
2086   {
2087     \exp_not:c { CTEX@pre#1 }
2088     \exp_not:c { CTEX@the#1 }
2089     \exp_not:c { CTEX@post#1 }
2090   }
2091   \tl_const:cx { CTEX@#1name }
2092   {
2093     \exp_not:c { CTEX@#1nameformat }
2094     \exp_not:c { CTEX@pre#1 }
2095     \exp_not:N \tl_if_empty:NTF \exp_not:c { CTEX@#1numberformat }
2096     { \exp_not:c { CTEX@the#1 } }
2097     {
2098       \group_begin:
2099         \exp_not:c { CTEX@#1numberformat }
2100         \exp_not:c { CTEX@the#1 }
2101       \group_end:
2102     }
2103     \exp_not:c { CTEX@post#1 }
2104     \exp_not:c { CTEX@#1aftername }
2105   }
2106 }

```

(End definition for `_ctex_initial_heading:n`)

`_ctex_def_heading_keys:n`

```

2107 \cs_new_protected_nopar:Npn \_ctex_def_heading_keys:n #1
2108 {
2109   \tl_put_right:Nx \l__ctex_tmp_tl
2110   {
2111     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2112     #1 / name .code:n =
2113       { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2114     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2115     #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2116     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2117     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2118     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2119     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2120     #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1@beforekip } ,
2121     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2122     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2123     #1 / format+ .code:n =
2124       { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2125     #1 / nameformat+ .code:n =
2126       { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2127     #1 / numberformat+ .code:n =
2128       { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2129     #1 / aftername+ .code:n =
2130       { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2131     #1 / titleformat+ .code:n =
2132       { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2133     #1 / beforekip .initial:n = \c_zero_skip ,
2134     #1 / afterskip .initial:n = \c_zero_skip ,
2135     #1 / indent .initial:n = \c_zero_dim ,
2136     #1 / beforekip .value_required: ,
2137     #1 / afterskip .value_required: ,
2138     #1 / indent .value_required: ,
2139   }
2140 }

```

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

```

2141 \NewDocumentCommand \ctex_assign_heading_name:nn
2142 { m > { \SplitArgument { \c_one } { , } } +m }
2143 { \_ctex_assign_heading_name:nnn {#1} {#2} }
2144 \cs_new_protected:Npn \_ctex_assign_heading_name:nnn #1#2#3
2145 {
2146   \tl_set:cn { CTEX@pre#1 } {#2}
2147   \IfNoValueTF {#3}
2148   { \tl_clear:c { CTEX@post#1 } }
2149   { \tl_set:cn { CTEX@post#1 } {#3} }
2150 }

```

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

```

2151 \tl_clear:N \l__ctex_tmp_tl
2152 \seq_map_inline:Nn \c__ctex_headings_seq
2153 {
2154   \_ctex_initial_heading:n {#1}
2155   \_ctex_def_heading_keys:n {#1}
2156 }
2157 \use:x { \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } } }

```

```
2158 \tl_clear:N \l__ctex_tmp_tl
```

pagestyle 只在 ctexbook 和 ctexrep 下有定义。

```
2159 <*book|report>
2160 \keys_define:nn { ctex }
2161 {
2162   part      / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2163   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle
2164 }
2165 </book|report>
```

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

15.8.2 标准标题命令的修改

15.8.2.1 part 的标题

```
2166 <@@=>
```

\part

```
2167 <*article>
2168 \renewcommand\part{%
2169   \if@noskipsec \leavevmode \fi
2170   \par
2171   % \addvspace{4ex}%
2172   \@tempskipa \CTEX@part@beforeskip \relax
2173   \ifdim \@tempskipa <\z@
2174     \@tempskipa -\@tempskipa \@afterindentfalse
2175   \else
2176     \@afterindenttrue
2177   \fi
2178   \addvspace{\@tempskipa}%
2179   \secdef\@part\@spart}
2180 </article>
```

(End definition for \part.)

\@part

```
2181 <*article>
2182 \def\@part[#1]#2{%
2183   \ifnum \c@secnumdepth >\m@ne
2184     \refstepcounter{part}%
2185     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2186     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2187   \else
2188     \addcontentsline{toc}{part}{#1}%
2189   \fi
2190   {\interlinepenalty \@M
2191   % \normalfont \parindent \z@ \raggedright
2192   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2193   \ifnum \c@secnumdepth >\m@ne
2194     % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2195     \CTEX@partname
2196   \fi
2197   % \huge\bfseries #2%
2198   \CTEX@part@titleformat{#2}%
2199   \markboth{}{}\par}%
2200   \nobreak
2201   % \vskip 3ex
2202   \vskip \CTEX@part@afterskip
2203   \@afterheading}
2204 </article>
```

标准文档类是在 `\part` 和 `\chapter` 定义的最开始设置 `\thispagestyle`, 我们这里的修改出现在它之后, 可以覆盖之前的设置。

```

2205 <*book|report>
2206 \def\@part[#1]#2{%
2207   \thispagestyle{\CTEX@part@pagestyle}%
2208   \ifnum \c@secnumdepth >-2\relax
2209     \refstepcounter{part}%
2210     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2211     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2212   \else
2213     \addcontentsline{toc}{part}{#1}%
2214   \fi
2215   \markboth{}{}%
2216   {\interlinepenalty \@M
2217    % \normalfont \centering
2218    \normalfont \CTEX@part@format
2219    \ifnum \c@secnumdepth >-2\relax
2220    % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2221    \CTEX@partname
2222    \fi
2223    % \Huge\bfseries #2\par}%
2224    \CTEX@part@titleformat{#2}\par}%
2225   \@endpart}
2226 </book|report>

```

(End definition for `\@part`.)

`\@spart`

```

2227 <*article>
2228 \def\@spart#1{%
2229   {\interlinepenalty \@M
2230    % \normalfont \parindent \z@ \raggedright
2231    \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2232    % \huge \bfseries #1\par}%
2233    \CTEX@part@titleformat{#1}\par}%
2234    \nobreak
2235    % \vskip 3ex
2236    \vskip \CTEX@part@afterskip
2237    \@afterheading}
2238 </article>
2239 <*book|report>
2240 \def\@spart#1{%
2241   {\interlinepenalty \@M
2242    % \normalfont \centering
2243    \normalfont \CTEX@part@format
2244    % \Huge \bfseries #1\par}%
2245    \CTEX@part@titleformat{#1}\par}%
2246    \@endpart}
2247 </book|report>

```

(End definition for `\@spart`.)

15.8.2.2 chapter 的标题

```

2248 <*book|report>

```

`\@chapter`

```

2249 \def\@chapter[#1]#2{%
2250   \ifnum \c@secnumdepth >\m@ne
2251   <book> \if@mainmatter
2252     \refstepcounter{chapter}%
2253     % \typeout{\@chapapp\space\thechapter.}%
2254     \typeout{\CTEXthechapter}%

```

```

2255 \addcontentsline{toc}{chapter}
2256 % {\protect\numberline{\thechapter}\#1}%
2257 {\protect\numberline{\CTEXthechapter\hspace{0.3em}}\#1}%
2258 <book> \else
2259 <book> \addcontentsline{toc}{chapter}{\#1}%
2260 <book> \fi
2261 \else
2262 \addcontentsline{toc}{chapter}{\#1}%
2263 \fi
2264 \chaptermark{\#1}%
2265 \addtocontents{lof}{\protect\addvspace{10\p@}}%
2266 \addtocontents{lot}{\protect\addvspace{10\p@}}%
2267 \if@twocolumn
2268 \topnewpage[\@makechapterhead{\#2}]%
2269 \else
2270 \makechapterhead{\#2}%
2271 \@afterheading
2272 \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2273 \def\@makechapterhead#1{%
2274 \thispagestyle{\CTEX@chapter@pagestyle}%
2275 % \vspace*{50\p@}%
2276 \@tempskipa \CTEX@chapter@beforeskip \relax
2277 \ifdim \@tempskipa <\z@
2278 \@tempskipa -\@tempskipa \@afterindentfalse
2279 \else
2280 \@afterindenttrue
2281 \fi
2282 \vspace*{\@tempskipa}%
2283 % {\normalfont \parindent \z@ \raggedright
2284 {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2285 \ifnum \c@secnumdepth >\m@ne
2286 <book> \if@mainmatter
2287 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2288 \CTEX@chaptername
2289 <book> \fi
2290 \fi
2291 \interlinepenalty\@M
2292 % \Huge\bfseries #1\par\nobreak
2293 \CTEX@chapter@titleformat{\#1}\par\nobreak
2294 % \vskip 40\p@
2295 \vskip \CTEX@chapter@afterskip
2296 }}

```

(End definition for \@makechapterhead.)

\@makeschapterhead

```

2297 \def\@makeschapterhead#1{%
2298 \thispagestyle{\CTEX@chapter@pagestyle}%
2299 % \vspace*{50\p@}%
2300 \@tempskipa \CTEX@chapter@beforeskip \relax
2301 \ifdim \@tempskipa <\z@
2302 \@tempskipa -\@tempskipa \@afterindentfalse
2303 \else
2304 \@afterindenttrue
2305 \fi
2306 \vspace*{\@tempskipa}%
2307 % {\normalfont \parindent \z@ \raggedright
2308 {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2309 \interlinepenalty\@M
2310 % \Huge\bfseries #1\par\nobreak
2311 \CTEX@chapter@titleformat{\#1}\par\nobreak

```

```

2312 % \vskip 40\p@
2313 \vskip \CTEX@chapter@afterskip
2314 }

```

(End definition for \@makeschapterhead.)

```

2315 </book|report>

```

15.8.2.3 section 类的标题

\@secntformat

```

2316 \def\@secntformat#1{%
2317   \@ifundefined{CTEX@#1name}%
2318     {\csname the#1\endcsname\quad}%
2319     {\csname CTEX@#1name\endcsname}}

```

(End definition for \@secntformat.)

\@sect

```

2320 \def\@sect#1#2#3#4#5#6[#7]#8{%
2321   \ifnum #2>\c@secnumdepth
2322     \let\@svsec\@empty
2323   \else
2324     \refstepcounter{#1}%
2325     \protected@edef\@svsec{\@secntformat{#1}\relax}%
2326   \fi
2327   \@tempskipa #5\relax
2328   \ifdim \@tempskipa>\z@
2329     \begingroup
2330       #6{%
2331         \@hangfrom{\hskip #3\relax\@svsec}%
2332         % \interlinepenalty \@M #8\@@par}%
2333         \interlinepenalty \@M
2334         \csname CTEX@#1@titleformat\endcsname{#8}\@@par}%
2335       \endgroup
2336       \csname #1mark\endcsname{#7}%
2337       \addcontentsline{toc}{#1}{%
2338         \ifnum #2>\c@secnumdepth \else
2339         % \protect\numberline{\csname the#1\endcsname}%
2340         \protect\numberline{\@ifundefined{CTEXthe#1}%
2341                               {\csname the#1\endcsname}%
2342                               {\csname CTEXthe#1\endcsname}}}%
2343       \fi
2344       #7}%
2345   \else
2346     \def\@svsechd{%
2347       #6{\hskip #3\relax
2348         % \@svsec #8}%
2349         \@svsec \csname CTEX@#1@titleformat\endcsname{#8}}%
2350     \csname #1mark\endcsname{#7}%
2351     \addcontentsline{toc}{#1}{%
2352       \ifnum #2>\c@secnumdepth \else
2353       % \protect\numberline{\csname the#1\endcsname}%
2354       \protect\numberline{\@ifundefined{CTEXthe#1}%
2355                             {\csname the#1\endcsname}%
2356                             {\csname CTEXthe#1\endcsname}}}%
2357     \fi
2358     #7}}%
2359   \fi
2360   \@xsect{#5}}

```

(End definition for \@sect.)

`\@ssect` `\@ssect` 并没有参数给出当前标题的名字, 扩展它的参数会与 `hyperref` 冲突。它的第二个参数是 `BEFORESKIP` (`\@startsection` 的第四个参数), 在定义中并没有被用到, 应该可以用它来传递名字。我们这里通过函数 `\CTEX@titleformat@n` 来传递, 它将在 #4 中被重定义为相应的 `titleformat`。

```

2361 \def\@ssect#1#2#3#4#5{%
2362   \@tempskipa #3\relax
2363   \ifdim \@tempskipa>\z@
2364     \begingroup
2365       #4{%
2366         \@hangfrom{\hskip #1}%
2367         % \interlinepenalty \@M #5\@par}%
2368         \interlinepenalty \@M
2369         \CTEX@titleformat@n{#5}\@par}%
2370     \endgroup
2371   \else
2372     % \def\@svsechd{#4{\hskip #1\relax #5}}%
2373     \def\@svsechd{#4{\hskip #1\relax \CTEX@titleformat@n{#5}}}%
2374   \fi
2375   \@xsect{#3}}

```

(End definition for \@ssect.)

```

2376 \@@=ctex

```

`\CTEX@set@titleformat@n` 在 `\@startsection` 中设置 `\CTEX@titleformat@n` 为相应函数。

```

\CTEX@titleformat@n
2377 \cs_new_protected_nopar:Npn \CTEX@set@titleformat@n #1
2378 { \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat } }
2379 \cs_new_eq:NN \CTEX@titleformat@n \use:n

```

(End definition for \CTEX@set@titleformat@n and \CTEX@titleformat@n.)

```

2380 \int_zero:N \l__ctex_tmp_int
2381 \clist_map_inline:nn
2382 { section , subsection , subsubsection , paragraph , subparagraph }
2383 {
2384   \int_incr:N \l__ctex_tmp_int
2385   \cs_gset_protected_nopar:cpx {#1}
2386   {
2387     \exp_not:N \@startsection {#1}
2388     { \int_use:N \l__ctex_tmp_int }
2389     { \exp_not:c { CTEX@#1@indent } }
2390     { \exp_not:c { CTEX@#1@beforeskip } }
2391     { \exp_not:c { CTEX@#1@afterskip } }
2392     {
2393       \CTEX@set@titleformat@n {#1}
2394       \exp_not:N \normalfont \exp_not:c { CTEX@#1@format }
2395     }
2396   }
2397 }

```

15.8.2.4 附录标题

```

2398 \tl_new:N \CTEX@preappendix
2399 \tl_new:N \CTEX@postappendix
2400 \keys_define:nn { ctex }
2401 { appendix .meta:nn = { ctex / appendix } {#1} }
2402 \keys_define:nn { ctex / appendix }
2403 {
2404   name .code:n = { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2405   number .tl_set:N = \CTEX@appendixnumber
2406 }

```

`\appendix`

```

2407 \cs_new_eq:NN \CTEX@save@appendix \appendix
2408 \cs_gset_protected_nopar:Npn \appendix
2409 {
2410   \CTEX@save@appendix
2411 }*article)
2412   \gdef \CTEX@presection { \CTEX@preappendix }
2413   \gdef \CTEX@thesection { \CTEX@appendixnumber }
2414   \gdef \CTEX@postsection { \CTEX@postappendix }
2415 }/article)
2416 }*book|report)
2417   \gdef \CTEX@prechapter { \CTEX@preappendix }
2418   \gdef \CTEX@thechapter { \CTEX@appendixnumber }
2419   \gdef \CTEX@postchapter { \CTEX@postappendix }
2420 }/book|report)
2421 }

```

(End definition for \appendix.)

15.8.3 目录标签的宽度

\numberline

```

2422 \cs_new_protected:Npn \CTEX@toc@width@n #1
2423 {
2424   \hbox_set:Nn \l__ctex_tmp_box {#1}
2425   \dim_set:Nn \@tempdima
2426   {
2427     \dim_max:nn { \@tempdima }
2428     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2429   }
2430 }
2431 \group_begin:
2432 \char_set_catcode_other:N \#
2433 \use:n
2434 {
2435   \group_end:
2436   \ExplSyntaxOff
2437   \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2438   { \ctex_patch_failure:N \numberline }
2439   \ifpackageloaded { tocloft }
2440   { }
2441   {
2442     \ctex_at_end_package:nn { tocloft }
2443     {
2444       \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2445       { \ctex_patch_failure:N \numberline }
2446     }
2447   }
2448   \ExplSyntaxOn
2449 }

```

(End definition for \numberline.)

15.8.4 页眉信息的修改

\ps@headings

```

2450 }*article)
2451 \ctex_patch_cmd:Nnn \ps@headings { \thesection } { \CTEXthesection }
2452 \if@twoside
2453 \ctex_patch_cmd:Nnn \ps@headings { \thesubsection } { \CTEXthesubsection }
2454 \fi:
2455 }/article)
2456 }*book|report)

```



```

2457 \ctex_patch_cmd:Nnn \ps@headings
2458 { \@chapapp\ \thechapter.\ } { \CTEXthechapter \quad }
2459 \if@twoside
2460 \ctex_patch_cmd:Nnn \ps@headings { \thesection.\ } { \CTEXthesection \quad }
2461 \fi:
2462 </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。

```

2463 \ctex_at_end_package:nn { fancyhdr }
2464 {
2465 <*article>
2466 \ctex_patch_cmd:Nnn \ps@fancy
2467 { \thesection \hskip 1em \relax } { \CTEXthesection \quad }
2468 \ctex_patch_cmd:Nnn \ps@fancy
2469 { \thesubsection \hskip 1em \relax } { \CTEXthesubsection \quad }
2470 </article>
2471 <*book|report>
2472 \ctex_patch_cmd:Nnn \ps@fancy
2473 <book> { \@chapapp\ \thechapter.\ } { \if@mainmatter \CTEXthechapter \quad \fi }
2474 <report> { \@chapapp\ \thechapter.\ } { \CTEXthechapter \quad }
2475 \ctex_patch_cmd:Nnn \ps@fancy
2476 { \thesection.\ } { \CTEXthesection \quad }
2477 </book|report>
2478 }

```

(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。

```

2479 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
2480 {
2481 \protected@edef \@currentlabel
2482 {
2483 \cs_if_exist:CTF { CTEX@the#1 }
2484 { \exp_args:cc { p@#1 } { CTEX@the#1 } }
2485 { \exp_not:o { \@currentlabel } }
2486 }
2487 }

```

(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 CT_EX 之后被载入,我们之前的修改将会被覆盖。

```

2488 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
2489 {
2490 \seq_map_inline:Nn \c__ctex_headings_seq
2491 { \ctex_fix_varioref_label:n { ##1 } }
2492 }

```

(End definition for \ctex_varioref_hook:.)

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

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

它的第二个参数必须放在花括号内, 否则将会被作为宏的定界符号。

```
2493 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
2494 {
2495   \tl_if_empty:cT { p@#1 }
2496   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
2497 }
```

(End definition for _ctex_fix_varioref_label:n.)

如果 `varioref` 已经被载入, 则使用它来设置。

```
2498 \@ifpackageloaded { varioref }
2499 { \ctex_varioref_hook: }
2500 {
2501   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
2502   \RenewDocumentCommand \refstepcounter { m }
2503   {
2504     \CTEX@save@refstepcounter {#1}
2505     \CTEX@setcurrentlabel@n {#1}
2506   }
2507   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
2508 }
```

15.8.6 载入 $\langle scheme \rangle$ 文件

```
2509 \ctex_scheme_input:o { \l__ctex_scheme_tl }
2510 \</class|heading>
```

15.8.7 标题格式的 `scheme` 定义

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

```
2511 \< *scheme&(article|book|report)>
2512 \keys_set:nn { ctex / part }
2513 {
2514   \< *plain>
2515     name      = \partname \space ,
2516     number    = \thepart ,
2517   \< /plain>
2518   \< *article>
2519     nameformat = \Large \bfseries ,
2520     afterskip  = 3ex ,
2521   \< *plain>
2522     format     = \raggedright ,
2523     aftername  = \par \nobreak ,
2524     titleformat = \huge \bfseries ,
2525     beforeskip = -4ex
2526   \< /plain>
2527   \< *chinese>
2528     number     = \chinese { part } ,
2529     format     = \centering ,
2530     aftername  = \quad ,
2531     titleformat = \Large \bfseries ,
2532     beforeskip = 4ex
2533   \< /chinese>
2534   \< /article>
2535   \< *book|report>
2536     format     = \centering ,
2537     nameformat = \huge \bfseries ,
2538     aftername  = \par \vskip 20 \p@ ,
2539     pagestyle  = plain ,
```

```

2540 <*plain>
2541     titleformat = \Huge \bfseries
2542 </plain>
2543 <*chinese>
2544     titleformat = \huge \bfseries
2545 </chinese>
2546 </book|report>
2547 }

2548 <*book|report>
2549 \keys_set:nn { ctex / chapter }
2550 {
2551     pagestyle      = plain ,
2552     nameformat     = \huge \bfseries ,
2553     afterskip      = 40 \p@ ,
2554 <*plain>
2555     name           = \chaptername \space ,
2556     number         = \thechapter ,
2557     format         = \raggedright ,
2558     aftername      = \par \nobreak \vskip 20 \p@ ,
2559     titleformat    = \Huge \bfseries ,
2560     beforekip      = -50 \p@
2561 </plain>
2562 <*chinese>
2563     number         = \chinese { chapter } ,
2564     format         = \centering ,
2565     aftername      = \quad ,
2566     titleformat    = \huge \bfseries ,
2567     beforekip      = 50 \p@
2568 </chinese>
2569 }
2570 </book|report>

2571 \keys_set:nn { ctex / section }
2572 {
2573     number         = \thesection ,
2574     aftername      = \quad ,
2575     afterskip      = 2.3ex \@plus .2ex ,
2576 <*plain>
2577     format         = \Large \bfseries ,
2578     beforekip      = -3.5ex \@plus -1ex \@minus -.2ex
2579 </plain>
2580 <*chinese>
2581     format         = \Large \bfseries \centering ,
2582     beforekip      = 3.5ex \@plus 1ex \@minus .2ex
2583 </chinese>
2584 }

2585 \keys_set:nn { ctex / subsection }
2586 {
2587     number         = \thesubsection ,
2588     format         = \large \bfseries ,
2589     aftername      = \quad ,
2590     afterskip      = 1.5ex \@plus .2ex ,
2591 <*plain>
2592     beforekip      = -3.25ex \@plus -1ex \@minus -.2ex
2593 </plain>
2594 <*chinese>
2595     beforekip      = 3.25ex \@plus 1ex \@minus .2ex
2596 </chinese>
2597 }

2598 \keys_set:nn { ctex / subsubsection }
2599 {
2600     number         = \thesubsubsection ,
2601     format         = \normalsize \bfseries ,
2602     aftername      = \quad ,
2603     afterskip      = 1.5ex \@plus .2ex ,

```

```

2604 <*plain>
2605     beforeskip = -3.25ex \@plus -1ex \@minus -.2ex
2606 </plain>
2607 <*chinese>
2608     beforeskip = 3.25ex \@plus 1ex \@minus .2ex
2609 </chinese>
2610 }

```

```

2611 \keys_set:nn { ctex / paragraph }
2612 {
2613     number      = \theparagraph ,
2614     format      = \normalsize \bfseries ,
2615     aftername    = \quad
2616 }

```

```

2617 \keys_set:nn { ctex / subparagraph }
2618 {
2619     number      = \thesubparagraph ,
2620     format      = \normalsize \bfseries ,
2621     aftername    = \quad
2622 }

```

处理 sub3section 与 sub4section 的格式。

```

2623 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2624 {
2625     \keys_set:nn { ctex / paragraph }
2626     {
2627 <*plain>
2628         beforeskip = -3.25ex \@plus -1ex \@minus -.2ex ,
2629 </plain>
2630 <*chinese>
2631         beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2632 </chinese>
2633         afterskip  = 1ex \@plus .2ex
2634     }
2635 }
2636 {
2637     \keys_set:nn { ctex / paragraph }
2638     {
2639         beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2640         afterskip  = -1em
2641     }
2642 }
2643 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
2644 {
2645     \keys_set:nn { ctex / subparagraph }
2646     {
2647 <*plain>
2648         beforeskip = -3.25ex \@plus -1ex \@minus -.2ex ,
2649 </plain>
2650 <*chinese>
2651         beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2652 </chinese>
2653         afterskip  = 1ex \@plus .2ex
2654     }
2655 }
2656 {
2657     \keys_set:nn { ctex / subparagraph }
2658     {
2659         beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2660         afterskip  = -1em
2661     }
2662 }
2663 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2664 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
2665 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

2666 \keys_set:nn { ctex / appendix }
2667 <*article>
2668   { number      = \@Alph \c@section }
2669 </article>
2670 <*book|report>
2671   {
2672     name        = \appendixname \space ,
2673     number      = \@Alph \c@chapter
2674   }
2675 </book|report>
2676 </scheme&(article|book|report)>

```

15.8.8 ctex.sty 的 heading 选项

```

2677 <*style>
2678 \msg_new:nnn { ctex } { not-standard-class }
2679 {
2680   None~of~the~standard~document~classes~was~loaded.\\
2681   Heading~`#1'~is~selected.\\
2682   ctex~may~not~work~as~expected.
2683 }
2684 \bool_if:NTF \l__ctex_heading_bool
2685 {
2686   \clist_map_inline:nn { article , book , report }
2687   {
2688     \@ifclassloaded {#1}
2689     { \clist_map_break:n { \tl_const:Nn \c__ctex_class_tl {#1} } } { }
2690   }
2691   \tl_if_exist:NF \c__ctex_class_tl
2692   {
2693     \cs_if_exist:NTF \chapter
2694     {
2695       \cs_if_exist:NF \if@mainmatter
2696       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
2697       \tl_const:Nn \c__ctex_class_tl { book }
2698     }
2699     { \tl_const:Nn \c__ctex_class_tl { article } }
2700     \msg_warning:nx { ctex } { not-standard-class } { \c__ctex_class_tl }
2701   }
2702   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
2703 }
2704 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
2705 </style>

```

15.8.9 标题配置文件

```

2706 <*UTF8|GBK>
2707 \keys_set:nn { ctex }
2708 {
2709   contentsname = 目录 ,
2710   listfigurename = 插图 ,
2711   listtablename = 表格 ,
2712   figurename    = 图 ,
2713   tablename     = 表 ,
2714   abstractname  = 摘要 ,
2715   indexname     = 索引 ,
2716   bibname       = 参考文献 ,
2717   appendixname  = 附录
2718 }
2719 \keys_if_exist:nnT { ctex / part } { name }
2720 {
2721   \keys_set:nn { ctex / part } { name = { 第 , 部分 } }

```

```

2722 \keys_if_exist:nnT { ctex / chapter } { name }
2723 { \keys_set:nn { ctex / chapter } { name = { 第 , 章 } } }
2724 }
2725 </UTF8|GBK>

```

15.9 chinese 方案的其他设置

```

2726 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

2727 <article|book|report>\pagestyle { headings }

```

汉化标题名字。

```

2728 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
2729 { \ctex_file_input:n { ctex-name-gbk.cfg } }
2730 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

日期格式。

```

2731 \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。

```

2732 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
2733 { \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。

```

2734 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
2735 { \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。

```

2736 \ctex_if_autoindent_touched:F
2737 { \keys_set:nn { ctex } { autoindent = true } }
2738 </scheme&chinese>

```

15.10 中文字号

```

2739 <*class|style|ctexsize>

```

\zihao

```

2740 \NewDocumentCommand \zihao { m }
2741 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

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

\ctex_zihao:n

```

2742 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
2743 {
2744   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
2745   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
2746   { \msg_error:nnn { ctex } { fontsize } {#1} }
2747 }
2748 \msg_new:nnnn { ctex } { fontsize }
2749 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
2750 {
2751   The~old~font~size~is~used~if~you~continue.\\
2752   The~available~font~sizes~are~listed~as~follow.\\
2753   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
2754 }

```

(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¹⁸。

```

2755 \prop_new:N \c__ctex_font_size_prop
2756 \seq_new:N \c__ctex_font_size_seq
2757 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
2758 {
2759   \use:x
2760   {
2761     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
2762     {
2763       { \dim_to_decimal:n {#2} }
2764       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
2765     }
2766   }
2767   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
2768 }
2769 \clist_map_inline:nn
2770 {
2771   { 8 } { 5 bp } ,
2772   { 7 } { 5.5 bp } ,
2773   { -6 } { 6.5 bp } ,
2774   { 6 } { 7.5 bp } ,
2775   { -5 } { 9 bp } ,
2776   { 5 } { 10.5 bp } ,
2777   { -4 } { 12 bp } ,
2778   { 4 } { 14 bp } ,
2779   { -3 } { 15 bp } ,
2780   { 3 } { 16 bp } ,
2781   { -2 } { 18 bp } ,
2782   { 2 } { 22 bp } ,
2783   { -1 } { 24 bp } ,
2784   { 1 } { 26 bp } ,
2785   { -0 } { 36 bp } ,
2786   { 0 } { 42 bp }
2787 }
2788 { \__ctex_save_font_size:nn #1 }

```

(End definition for `\c__ctex_font_size_prop`.)

`\ctex_declare_math_sizes:nnnn`

```

2789 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
2790 {
2791   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
2792   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
2793 }

```

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

`__ctex_get_font_sizes:Nn`

```

2794 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
2795 {
2796   \tl_clear:N #1
2797   \tl_map_inline:nn {#2}
2798   {
2799     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
2800     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
2801     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
2802   }
2803 }

```

¹⁸<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

(End definition for `_ctex_get_font_sizes:Nn`.)

```

2804 \clist_map_inline:nn
2805 {
2806   { 8 }{ 8 }{ 5pt }{ 5pt } ,
2807   { 7 }{ 7 }{ 5pt }{ 5pt } ,
2808   { -6 }{ -6 }{ 5pt }{ 5pt } ,
2809   { 6 }{ 6 }{ 5pt }{ 5pt } ,
2810   { -5 }{ -5 }{ 6pt }{ 5pt } ,
2811   { 5 }{ 5 }{ 7pt }{ 5pt } ,
2812   { -4 }{ -4 }{ 8pt }{ 6pt } ,
2813   { 4 }{ 4 }{ 5 }{ 6 } ,
2814   { -3 }{ -3 }{ -4 }{ -5 } ,
2815   { 3 }{ 3 }{ 4 }{ 5 } ,
2816   { -2 }{ -2 }{ -3 }{ -4 } ,
2817   { 2 }{ 2 }{ 3 }{ 4 } ,
2818   { -1 }{ -1 }{ -2 }{ -3 } ,
2819   { 1 }{ 1 }{ 2 }{ 3 } ,
2820   { -0 }{ -0 }{ -1 }{ -2 } ,
2821   { 0 }{ 0 }{ 1 }{ 2 }
2822 }
2823 { \ctex_declare_math_sizes:nmmm #1 }

```

15.10.2 修改默认字号大小

`\ctex_set_font_size:Nnn`

```

2824 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
2825 {
2826   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
2827   { \exp_after:wN \_ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
2828   { \msg_error:nmm { ctex } { fontsize } {#2} }
2829 }
2830 \cs_new_protected:Npn \_ctex_set_font_size:nnNn #1#2#3#4
2831 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

(End definition for `\ctex_set_font_size:Nnn`.)

```

2832 \if_case:w \g__ctex_font_size_flag
2833 \ctex_file_input:n { ctex-c5size.clo }
2834 \or:
2835 \ctex_file_input:n { ctex-cs4size.clo }
2836 \fi:
2837 </class|style|ctexsize>
2838 <*c5size>
2839 \ctex_set_font_size:Nnn \normalsize { 5 }
2840 {
2841   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2842   \abovedisplayshortskip \z@ \@plus3\p@
2843   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2844   \belowdisplayskip \abovedisplayskip
2845   \let\@listi\@listI
2846 }
2847 \ctex_set_font_size:Nnn \small { -5 }
2848 {
2849   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
2850   \abovedisplayshortskip \z@ \@plus2\p@
2851   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
2852   \def\@listi{\leftmargin\leftmarginI
2853     \topsep 4\p@ \@plus2\p@ \@minus2\p@
2854     \parsep 2\p@ \@plus\p@ \@minus\p@
2855     \itemsep \parsep}
2856   \belowdisplayskip \abovedisplayskip
2857 }
2858 \ctex_set_font_size:Nnn \footnotesize { 6 }

```



```

2859 {
2860   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
2861   \abovedisplayshortskip \z@ \@plus\p@
2862   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
2863   \def\@listi{\leftmargin\leftmargini
2864             \topsep 3\p@ \@plus\p@ \@minus\p@
2865             \parsep 2\p@ \@plus\p@ \@minus\p@
2866             \itemsep \parsep}
2867   \belowdisplayskip \abovedisplayskip
2868 }
2869 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
2870 \ctex_set_font_size:Nnn \tiny { 7 } { }
2871 \ctex_set_font_size:Nnn \large { -4 } { }
2872 \ctex_set_font_size:Nnn \Large { -3 } { }
2873 \ctex_set_font_size:Nnn \LARGE { -2 } { }
2874 \ctex_set_font_size:Nnn \huge { 2 } { }
2875 \ctex_set_font_size:Nnn \Huge { 1 } { }
2876 </cs5size>
2877 <*cs4size>
2878 \ctex_set_font_size:Nnn \normalsize { -4 }
2879 {
2880   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
2881   \abovedisplayshortskip \z@ \@plus3\p@
2882   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
2883   \belowdisplayskip \abovedisplayskip
2884   \let\@listi\@listI
2885 }
2886 \ctex_set_font_size:Nnn \small { 5 }
2887 {
2888   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
2889   \abovedisplayshortskip \z@ \@plus3\p@
2890   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
2891   \def\@listi{\leftmargin\leftmargini
2892             \topsep 9\p@ \@plus3\p@ \@minus5\p@
2893             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
2894             \itemsep \parsep}
2895   \belowdisplayskip \abovedisplayskip
2896 }
2897 \ctex_set_font_size:Nnn \footnotesize { -5 }
2898 {
2899   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2900   \abovedisplayshortskip \z@ \@plus3\p@
2901   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2902   \def\@listi{\leftmargin\leftmargini
2903             \topsep 6\p@ \@plus2\p@ \@minus2\p@
2904             \parsep 3\p@ \@plus2\p@ \@minus\p@
2905             \itemsep \parsep}
2906   \belowdisplayskip \abovedisplayskip
2907 }
2908 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
2909 \ctex_set_font_size:Nnn \tiny { -6 } { }
2910 \ctex_set_font_size:Nnn \large { -3 } { }
2911 \ctex_set_font_size:Nnn \Large { -2 } { }
2912 \ctex_set_font_size:Nnn \LARGE { 2 } { }
2913 \ctex_set_font_size:Nnn \huge { -1 } { }
2914 \ctex_set_font_size:Nnn \Huge { 1 } { }
2915 </cs4size>
2916 <ctexsize>\normalsize
2917 <*class|style>

```

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。

```

2918 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
2919 {
2920   \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。

```

2921   \group_begin: \footnotesize \exp_args:NNNo \group_end:
2922   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
2923 }

```

(End definition for `\footnotesep`.)

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

```

2924 \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。

```

2925 \NewDocumentCommand \CTeX { }
2926 { \ifinccsname CTeX \else: C \TeX \fi: }
2927 \AtBeginDocument
2928 {
2929   \cs_if_exist_use:NT \pdfstringdefDisableCommands
2930   { { \tl_set:Nn \CTeX { CTeX } } }
2931 }

```

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

`captiondelimater` 过时选项。

```

2932 \keys_define:nn { ctex }
2933 {
2934   captiondelimater .code:n =
2935   {
2936     \msg_warning:nnn { ctex } { deprecated-option }
2937     { You~can~load~the~package~`caption'~to~get~its~functionality. }
2938   }
2939 }

```

(End definition for `captiondelimater`.)

```

2940 </class|style>

```

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`

```

2941 <*scheme&!generic&chinese>
2942 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
2943 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
2944 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
2945 </scheme&!generic&chinese>

```

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

```

2946 <*class|style>

```

15.12.2 其他兼容性修改

`\end breqn` 宏包对 `\end` 作了如下处理, 然而这个处理并不保险。

```
\def\@tempa#1\endcsname#2\@nil{\def\latex@end##1{#2}}
\expandafter\@tempa\end{#1}\@nil
\def\end#1{\csname end#1\endcsname \latex@end{#1}}%
```

etoolbox 在 `\end` 定义中的 `\csname` 前加入钩子 `\csuse{@end@#1@hook}`。如果 etoolbox 先于 breqn 被载入(这在使用 ctexart 等文档类时几乎是必然的), `\csuse{@end@#1@hook}` 将会被忽略, 即 `\AtEndEnvironment` 失效。如果交换两个宏包的载入顺序, 则 etoolbox 会给出警告: `\AfterEndEnvironment` 失效, 我们打算处理这种情况。

```
2947 \group_begin:
2948 \char_set_catcode_other:N \#
2949 \use:n
2950 {
2951   \group_end:
2952   \ExplSyntaxOff
2953   \ctex_at_end_package:nn { breqn }
2954   {
2955     \ctex_patch_cmd:NnnTF \end { \csname end#1 \endcsname }
2956     {
2957       \csuse { @end@#1@hook }
2958       \csname end#1 \endcsname
2959     } { }
2960     {
2961       \ctex_preto_cmd:NnTF \end { \csuse { @end@#1@hook } }
2962       { } { \ctex_patch_failure:N \end }
2963     }
2964   }
2965   \ExplSyntaxOn
2966 }
```

(End definition for `\end`.)

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。

```
2967 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
2968 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
2969 \msg_new:nnn { ctex } { fontset-unavailable }
2970 { 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。

```
2971 \cs_new_protected_nopar:Npn \ctex_load_fontset:
2972 {
2973   \tl_if_empty:NTF \g__ctex_fontset_tl
2974   {
2975     \ctex_detected_platform:
2976     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2977   }
2978   {
2979     \file_if_exist:NTF { ctex-fontset- \g__ctex_fontset_tl .def }
2980     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
2981     {
2982       \use:x
2983       {
2984         \ctex_detected_platform:
2985         \msg_error:nnxx { ctex } { fontset-not-found }
2986         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
```

```

2987         }
2988         \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2989     }
2990 }
2991 }
2992 \@onlypreamble \ctex_load_fontset:
2993 \msg_new:nnnn { ctex } { fontset-not-found }
2994 {
2995     CTex~fontset~`#1'~could~not~be~found.\\
2996     Fontset~`#2'~will~be~used~instead.
2997 }
2998 { 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。

```

2999 \keys_define:nn { ctex }
3000 {
3001     fontset .code:n =
3002     {
3003         \ctex_if_preamble:TF
3004         {
3005             \str_if_eq_x:nnTF {#1} { none }
3006             { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3007             {
3008                 \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3009                 {
3010                     \tl_gset:Nx \g__ctex_fontset_tl {#1}
3011                     \ctex_load_fontset:
3012                 }
3013                 {
3014                     \msg_error:nnxx { ctex } { fontset-loaded }
3015                     { \g__ctex_fontset_tl } {#1}
3016                 }
3017             }
3018         }
3019         { \msg_error:nn { ctex } { fontset-only-preamble } }
3020     }
3021 }
3022 \msg_new:nnnn { ctex } { fontset-loaded }
3023 {
3024     CTex~fontset~`#1'~has~been~loaded.
3025     \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
3026 }
3027 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3028 \msg_new:nnn { ctex } { fontset-only-preamble }
3029 {
3030     The~`fontset'~option~can~be~used~only~in~preamble.
3031 }

```

(End definition for fontset. This function is documented on page 8.)

载入中文字库。

```

3032 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3033 { \ctex_load_fontset: }

```

15.14 宏包配置文件

15.14.1 ctex.cfg

```

3034 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3035 </class|style>
3036 <*config>

```

```

3037 %%
3038 </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。

```

3039 <*ctexopts>
3040 %%
3041 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3042 </ctexopts>

```

15.15 字体定义文件

15.15.1 传统定义方式

```

3043 <*c19|c70>
3044 %%
3045 %% Chinese characters
3046 %%
3047 <c19>%% character set: GBK (extension of GB 2312)
3048 <c70>%% character set: Unicode
3049 %% font encoding: Unicode
3050 %%
3051 </c19|c70>

3052 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3053 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3054 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3055 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3056 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3057 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

3058 <*rm>
3059 <*c19>
3060 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3061 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3062 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3063 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3064 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3065 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3066 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3067 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3068 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3069 </c19>
3070 <*c70>
3071 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3072 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3073 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3074 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3075 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3076 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3077 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3078 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3079 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3080 </c70>
3081 </rm>

3082 <*sf>
3083 <*c19>
3084 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3085 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3086 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3087 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3088 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3089 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3090 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}

```

```

3091 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3092 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3093 </c19>
3094 <*c70>
3095 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyoud}{\CJKnormal}
3096 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyoud}{\CJKbold}
3097 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyoud}{\CJKbold}
3098 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyoudsl}{\CJKnormal}
3099 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyoudsl}{\CJKbold}
3100 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyoudsl}{\CJKbold}
3101 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyoud}{\CJKnormal}
3102 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyoud}{\CJKbold}
3103 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyoud}{\CJKbold}
3104 </c70>
3105 </sf>
3106 <*tt>
3107 <*c19>
3108 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfss}{\CJKnormal}
3109 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfss}{\CJKbold}
3110 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfss}{\CJKbold}
3111 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfsssl}{\CJKnormal}
3112 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfsssl}{\CJKbold}
3113 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfsssl}{\CJKbold}
3114 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfss}{\CJKnormal}
3115 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfss}{\CJKbold}
3116 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfss}{\CJKbold}
3117 </c19>
3118 <*c70>
3119 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifss}{\CJKnormal}
3120 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifss}{\CJKbold}
3121 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifss}{\CJKbold}
3122 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifsssl}{\CJKnormal}
3123 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifsssl}{\CJKbold}
3124 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifsssl}{\CJKbold}
3125 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifss}{\CJKnormal}
3126 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifss}{\CJKbold}
3127 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifss}{\CJKbold}
3128 </c70>
3129 </tt>
3130 <*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。

```

3131 <*windows>
3132 \file_if_exist:nTF { C:/boot.ini }
3133 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3134 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3135 </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。

```

3136 <*windowsnew, windowsold>
3137 \pdfTeX_if_engine:TF
3138 {
3139   \ctex_zhmap_case:nnn
3140   {
3141     \ctex_punct_set:n { windows }
3142     \setCJKmainfont
3143       [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3144 <*windowsold>
3145     \setCJKsansfont { simhei.ttf }
3146     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }

```

3147 </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。

```

3148 (*windowsnew)
3149     \file_if_exist:nTF { C:/Windows/Fonts/msyh.ttc }
3150     {
3151         \setCJKsansfont [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3152         \setCJKfamilyfont { zhyahei }
3153             [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3154     }
3155     {
3156         \setCJKsansfont [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3157         \setCJKfamilyfont { zhyahei }
3158             [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3159     }
3160     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3161     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3162 </windowsnew>
3163     \setCJKmonofont { simfang.ttf }
3164     \setCJKfamilyfont { zhkai } { simkai.ttf }
3165     \setCJKfamilyfont { zhfs } { simfang.ttf }
3166     \setCJKfamilyfont { zhsong } { simsun.ttc }
3167     \setCJKfamilyfont { zhhei } { simhei.ttf }
3168     \setCJKfamilyfont { zhli } { simli.ttf }
3169     \setCJKfamilyfont { zhyou } { simyou.ttf }
3170     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3171     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3172     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3173     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3174 }
3175 {
3176     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3177     \ctex_punct_set:n { windows }
3178     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3179     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3180     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3181 }
3182 {
3183     \tl_set:Nn \CJKrmdefault { rm }
3184     \tl_set:Nn \CJKsfdefault { sf }
3185     \tl_set:Nn \CJKttdefault { tt }
3186 }
3187 }
3188 {
3189 (*windowsold)
3190     \setCJKmainfont
3191         [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3192     \setCJKsansfont { SimHei }
3193     \setCJKmonofont { FangSong_GB2312 }
3194     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3195     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3196 </windowsold>
3197 (*windowsnew)
3198     \setCJKmainfont
3199         [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3200     \setCJKsansfont
3201         [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3202     \setCJKmonofont { FangSong }
3203     \setCJKfamilyfont { zhkai } { KaiTi }
3204     \setCJKfamilyfont { zhfs } { FangSong }
3205 </windowsnew>
3206     \setCJKfamilyfont { zhsong } { SimSun }
3207     \setCJKfamilyfont { zhhei } { SimHei }
3208     \setCJKfamilyfont { zhli } { LiSu }
3209     \setCJKfamilyfont { zhyou } { YouYuan }
3210 (*windowsnew)
3211     \setCJKfamilyfont { zhyahei }

```

```

3212      [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3213 </windowsnew>
3214   }
3215 </windowsnew,windowsold>

```

15.15.3 ctex-fontset-adobe.def

```

3216 (*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。

```

3217 \pdfutex_if_engine:TF
3218 {
3219   \ctex_if_pdfmode:TF
3220   { \ctex_fontset_error:n { adobe } }
3221   {
3222     \ctex_zhmap_case:nnn
3223     {
3224       \setCJKmainfont
3225       [
3226         cmap = UniGB-UTF16-H ,
3227         BoldFont = AdobeHeitiStd-Regular.otf ,
3228         ItalicFont = AdobeKaitiStd-Regular.otf
3229       ] { AdobeSongStd-Light.otf }
3230       \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3231       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3232       \setCJKfamilyfont { zhsong }
3233       [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3234       \setCJKfamilyfont { zhhei }
3235       [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3236       \setCJKfamilyfont { zhkai }
3237       [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3238       \setCJKfamilyfont { zhfs }
3239       [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3240       \ctex_punct_set:n { adobe }
3241       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3242       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3243       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3244       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3245       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3246     }
3247     {
3248       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3249       \ctex_punct_set:n { adobe }
3250       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3251       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3252       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3253     }
3254     { \ctex_fontset_error:n { adobe } }
3255   }
3256 }
3257 {
3258   \setCJKmainfont
3259   [
3260     BoldFont = AdobeHeitiStd-Regular ,
3261     ItalicFont = AdobeKaitiStd-Regular
3262   ] { AdobeSongStd-Light }
3263   \setCJKsansfont { AdobeHeitiStd-Regular }
3264   \setCJKmonofont { AdobeFangsongStd-Regular }
3265   \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3266   \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3267   \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3268   \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3269 }
3270 </adobe>

```


3271 $\langle *fandol \rangle$

15.15.4 ctex-fontset-fandol.def

```

3272 \pdfTEX_if_engine:TF
3273 {
3274   \ctex_if_pdfmode:TF
3275   { \ctex_fontset_error:n { fandol } }
3276   {
3277     \ctex_zhmap_case:nnn
3278     {
3279       \setCJKmainfont
3280       [
3281         cmap = UniGB-UTF16-H ,
3282         BoldFont = FandolSong-Bold.otf ,
3283         ItalicFont = FandolKai-Regular.otf
3284       ] { FandolSong-Regular.otf }
3285       \setCJKsansfont
3286       [
3287         cmap = UniGB-UTF16-H ,
3288         BoldFont = FandolHei-Bold.otf
3289       ] { FandolHei-Regular.otf }
3290       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3291       \setCJKfamilyfont { zhsong }
3292       [
3293         cmap = UniGB-UTF16-H ,
3294         BoldFont = FandolSong-Bold.otf
3295       ] { FandolSong-Regular.otf }
3296       \setCJKfamilyfont { zhhei }
3297       [
3298         cmap = UniGB-UTF16-H ,
3299         BoldFont = FandolHei-Bold.otf
3300       ] { FandolHei-Regular.otf }
3301       \setCJKfamilyfont { zhfs }
3302       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3303       \setCJKfamilyfont { zhkai }
3304       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
3305       \ctex_punct_set:n { fandol }
3306       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3307       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3308       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3309       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3310       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3311       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
3312     }
3313     {
3314       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
3315       \ctex_punct_set:n { fandol }
3316       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3317       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3318       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3319     }
3320     { \ctex_fontset_error:n { fandol } }
3321   }
3322 }
3323 {
3324   \setCJKmainfont
3325   [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
3326   { FandolSong-Regular }
3327   \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3328   \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
3329   \setCJKfamilyfont { zhsong }
3330   [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
3331   \setCJKfamilyfont { zhhei }
3332   [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3333   \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
3334   \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }

```

```

3335 }
3336 </fandol>
3337 <*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¹⁹:

```

/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 DVIPDFM_x 下,可以通过下述方式使用华文宋体和华文楷体:

```

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

```

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

```

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

```

DVIPDFM_x 不能生成 PDF,报下述错误:

```

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

```

如果将 CMap 改为 UniGB-UTF16-H,错误信息是

```

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

```

¹⁹<http://support.apple.com/kb/HT5944>

在 pdfTeX 下生成的 PDF 只有方框²⁰。华文细黑和华文仿宋的情况类似。

```

3338 \pdfTeX_if_engine:TF
3339 { \ctex_fontset_error:n { mac } }
3340 {
3341   \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
3342   \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
3343   \setCJKmonofont { STFangsong }
3344   \setCJKfamilyfont { zhsong } { STSong }
3345   \setCJKfamilyfont { zhhei } { STHeiti }
3346   \setCJKfamilyfont { zhfs } { STFangsong }
3347   \setCJKfamilyfont { zhkai } { STKaiti }
3348 }
3349 </mac>
3350 <*founder>

```

15.15.6 ctex-fontset-founder.def

```

3351 \pdfTeX_if_engine:TF
3352 {
3353   \ctex_zhmap_case:nnn
3354   {
3355     \setCJKmainfont
3356       [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
3357     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
3358     \setCJKmonofont { FZFSK.TTF }
3359     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
3360     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
3361     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
3362     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
3363     \setCJKfamilyfont { zhli } { FZLSK.TTF }
3364     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
3365     \ctex_punct_set:n { founder }
3366     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3367     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
3368     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3369     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3370     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3371     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
3372     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
3373   }
3374   {
3375     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
3376     \ctex_punct_set:n { founder }
3377     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3378     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3379     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3380   }
3381   { \ctex_fontset_error:n { founder } }
3382 }
3383 {
3384   \setCJKmainfont
3385     [ 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 luaotfload 不支持汉字名称，故使用 Postscript 名字，速度不受影响。

```

3386   \setCJKsansfont [ BoldFont = FZHei-B01 ]
3387   { \xetex_if_engine:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }

```

²⁰<http://www.newsmth.net/bbscon.php?bid=460&id=312640>

```

3388 \setCJKmonofont { FZFangSong-Z02 }
3389 \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
3390 \setCJKfamilyfont { zhhei } { FZHei-B01 }
3391 \setCJKfamilyfont { zhkai } { FZKai-Z03 }
3392 \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
3393 \setCJKfamilyfont { zhli } { FZLiShu-S01 }
3394 \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
3395 }
3396 </founder>
3397 <*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²¹。

```

3398 \pdfTeX_if_engine:TF
3399 {
3400   \ctex_zhmap_case:nnn
3401   {
3402     \setCJKmainfont
3403     [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
3404     \setCJKsansfont { wqy-zenhei.ttc }
3405     \setCJKmonofont { uming.ttc }
3406     \setCJKfamilyfont { zhsong } { uming.ttc }
3407     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
3408     \setCJKfamilyfont { zhkai } { ukai.ttc }
3409     \ctex_punct_set:n { ubuntu }
3410     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3411     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3412     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
3413     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3414     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3415   }
3416   {
3417     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
3418     \ctex_punct_set:n { ubuntu }
3419     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3420     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3421     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3422   }
3423   { \ctex_fontset_error:n { ubuntu } }
3424 }
3425 {
3426   \setCJKmainfont [ ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
3427   \setCJKsansfont { WenQuanYi~Zen~Hei }
3428   \setCJKmonofont { AR~PL~UMing~CN }
3429   \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
3430   \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
3431   \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
3432 }
3433 </ubuntu>

```

15.15.8 中文字体命令

```

3434 <!*windows>
3435 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }

```

²¹<http://www.freedesktop.org/wiki/Software/CJKUnifonts/Download/>

```

3436 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
3437 \lubuntu\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
3438 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
3439 \*windowsnew|windowsold|founder)
3440 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
3441 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
3442 \</windowsnew|windowsold|founder)
3443 \<windowsnew)\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
3444 \</!windows)
3445 \</fontset)
3446 \< *zhmap)

```

15.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

3447 \begingroup\catcode61\catcode48\catcode32=10\relax%
3448 \catcode 35=6 % #
3449 \catcode123=1 % {
3450 \catcode125=2 % }
3451 \toks0{\endlinechar=\the\endlinechar\relax}%
3452 \toks2{\endlinechar=13}%
3453 \def\x#1 #2 {%
3454   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
3455   \toks2\expandafter{\the\toks2 \catcode#1=#2}}%
3456 \x 13 5 % carriage return
3457 \x 32 10 % space
3458 \x 35 6 % #
3459 \x 40 12 % (
3460 \x 41 12 % )
3461 \x 45 12 % -
3462 \x 46 12 % .
3463 \x 47 12 % /
3464 \x 58 12 % :
3465 \x 60 12 % <
3466 \x 61 12 % =
3467 \x 64 11 % @
3468 \x 91 12 % [
3469 \x 93 12 % ]
3470 \x 123 1 % {
3471 \x 125 2 % }
3472 \edef\x#1{\endgroup%
3473   \edef\noexpand#1{%
3474     \the\toks0 %
3475     \let\noexpand\noexpand\noexpand#1%
3476     \noexpand\noexpand\noexpand\undefined%
3477     \noexpand\noexpand\noexpand\endinput}%
3478     \the\toks2}%
3479 \expandafter\x\csname ctex@zhmap@endinput\endcsname
3480 \input ifpdf.sty\relax

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

3481 \begingroup
3482 \expandafter\ifx\csname ProvidesFile\endcsname\relax
3483   \long\def\x#1\ProvidesFile#2[#3]{%
3484     #1%
3485     \immediate\write-1{File: #2 #3}%
3486     \expandafter\xdef\csname ver@#2\endcsname{#3}}
3487   \expandafter\x%
3488 \fi
3489 \endgroup

```

15.15.9.1 zhwindowsfonts.tex

```

3490 (*windows)
3491 \ProvidesFile{zhwindowsfonts.tex}%
3492 [2014/06/03 v2.0 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
3493
3494 \ifpdf
3495 \pdfmapline{=gbk@UGBK@ <simsum.ttc}
3496 \pdfmapline{=gbksong@UGBK@ <simsum.ttc}
3497 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
3498 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
3499 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
3500 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
3501 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
3502
3503 \pdfmapline{=cyberb@Unicode@ <simsum.ttc}
3504 \pdfmapline{=unisong@Unicode@ <simsum.ttc}
3505 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
3506 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
3507 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
3508 \pdfmapline{=unili@Unicode@ <simli.ttf}
3509 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
3510
3511 \pdfmapline{=gbksongsl@UGBK@ <simsum.ttc}
3512 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
3513 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
3514 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
3515 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
3516 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
3517
3518 \pdfmapline{=unisongsl@Unicode@ <simsum.ttc}
3519 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
3520 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
3521 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
3522 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
3523 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
3524
3525 \else
3526 \special{pdf:mapline gbk@UGBK@ unicode :0:simsum.ttc -v 50}
3527 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsum.ttc -v 50}
3528 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
3529 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
3530 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
3531 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
3532 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
3533
3534 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsum.ttc -v 50}
3535 \special{pdf:mapline unisong@Unicode@ unicode :0:simsum.ttc -v 50}
3536 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
3537 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
3538 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
3539 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
3540 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
3541
3542 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsum.ttc -s .167 -v 50}
3543 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
3544 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
3545 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
3546 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
3547 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
3548
3549 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsum.ttc -s .167 -v 50}
3550 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
3551 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
3552 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
3553 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
3554 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
3555
3556 \end{windows}

```

15.15.9.2 zhadobefonts.tex

```

3557 (*adobe)
3558 \ProvidesFile{zhadobefonts.tex}%
3559 [2014/06/03 v2.0 Adobe font map loader for DVIPDFMx (CTEX)]
3560
3561 \ifpdf
3562 %% pdfTeX does not support OTF fonts
3563
3564 \else
3565 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3566 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3567 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3568 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3569 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3570
3571 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3572 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3573 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3574 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3575 \special{pdf:mapline unifss@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3576
3577 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3578 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3579 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3580 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3581
3582 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3583 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3584 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3585 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3586
3587 (/adobe)

```

15.15.9.3 zhfandolfonts.tex

```

3588 (*fandol)
3589 \ProvidesFile{zhfandolfonts.tex}%
3590 [2014/06/03 v2.0 Fandol font map loader for DVIPDFMx (CTEX)]
3591
3592 \ifpdf
3593 %% pdfTeX does not support OTF fonts
3594
3595 \else
3596 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3597 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3598 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
3599 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
3600 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
3601
3602 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3603 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3604 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
3605 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
3606 \special{pdf:mapline unifss@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
3607
3608 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3609 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3610 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3611 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3612
3613 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3614 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3615 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3616 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3617
3618 (/fandol)

```

15.15.9.4 zhfounderfonts.tex

```

3619 (*founder)
3620 \ProvidesFile{zhfounderfonts.tex}%
3621 [2014/06/03 v2.0 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
3622
3623 \ifpdf
3624 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
3625 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
3626 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
3627 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
3628 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
3629 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
3630 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
3631
3632 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
3633 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
3634 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
3635 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
3636 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
3637 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
3638 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
3639
3640 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
3641 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
3642 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
3643 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
3644 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
3645 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
3646
3647 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
3648 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
3649 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
3650 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
3651 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
3652 \pdfmapline{=uniyoussl@Unicode@ <FZY1K.TTF}
3653
3654 \else
3655 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
3656 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
3657 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
3658 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
3659 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
3660 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
3661 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
3662
3663 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
3664 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
3665 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
3666 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
3667 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
3668 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
3669 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
3670
3671 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
3672 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
3673 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
3674 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
3675 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
3676 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
3677
3678 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
3679 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
3680 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
3681 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
3682 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
3683 \special{pdf:mapline uniyoussl@Unicode@ unicode FZY1K.TTF -s .167}
3684

```



```
3685 </founder>
```

15.15.9.5 zhubuntufonts.tex

```
3686 <*ubuntu>
3687 \ProvidesFile{zhubuntufonts.tex}%
3688 [2014/06/03 v2.0 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
3689
3690 \ifpdf
3691 \pdfmapline{=gbk@UGBK@ <uming.ttc}
3692 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
3693 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
3694 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
3695 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
3696 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
3697
3698 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
3699 \pdfmapline{=unisong@Unicode@ <uming.ttc}
3700 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
3701 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
3702 \pdfmapline{=unifs@Unicode@ <uming.ttc}
3703 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
3704
3705 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
3706 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
3707 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
3708 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
3709 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
3710
3711 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
3712 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
3713 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
3714 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
3715 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
3716
3717 \else
3718 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
3719 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
3720 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
3721 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
3722 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
3723
3724 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
3725 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
3726 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
3727 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
3728 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
3729
3730 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
3731 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
3732 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
3733 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
3734
3735 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
3736 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
3737 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
3738 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
3739
3740 </ubuntu>
3741 \fi
3742
3743 \ctex@zhmap@endinput
3744 </zhmap>
```

15.15.10 制作 spa 文件

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

```

3745 {*spa}
3746 {*macro}
3747 \input expl3-generic %
3748 \ExplSyntaxOn
3749 \xetex_if_engine:F
3750 {
3751   \msg_new:nn { ctex } { xetex }
3752     { XeTeX~is~required~to~compile~this~document! }
3753   \msg_fatal:nn { ctex } { xetex }
3754 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

3755 \seq_new:N \c__ctex_punct_seq
3756 \seq_gset_from_clist:Nn \c__ctex_punct_seq
3757 {
3758     "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
3759     "3008 , "300A , "3016 , "3010 ,
3760     "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
3761     "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
3762     "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
3763 }

```

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

```

3764 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
3765 {
3766     \iow_open:Nn \g__ctex_spa_iow {#1}
3767     \clist_map_inline:nn {#2}
3768     { \__ctex_write_family:nn ##1 }
3769     \iow_close:N \g__ctex_spa_iow
3770 }
3771 \iow_new:N \g__ctex_spa_iow
3772 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

```

(End definition for \ctex_make_spa:nn.)

```

3773 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
3774 {
3775   \group_begin:
3776     \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
3777     \l__ctex_punct_font
3778     \clist_clear:N \l__ctex_punct_bounds_clist
3779     \seq_map_inline:Nn \c__ctex_punct_seq
3780       { \exp_args:No \__ctex_save_bounds:n { \int_use:N \XeTeXcharglyph ##1 } }
3781     \iow_now:Nx \g__ctex_spa_iow
3782     {
3783       \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

3784         { \l__ctex_punct_bounds_clist , , , }
3785     }
3786     \group_end:
3787 }
3788 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
3789 {

```

```

3790 \clist_put_right:Nx \l__ctex_punct_bounds_clist
3791 {
3792   \__ctex_calc_bounds:nn { \c_one } {#1} ,
3793   \__ctex_calc_bounds:nn { \c_three } {#1}
3794 }
3795 }
3796 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1em 的比值的一百倍。

```

3797 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
3798 {
3799   \fp_eval:n
3800   {
3801     round
3802     (
3803       \dim_to_decimal_in_unit:nn
3804       { 100 \XeTeXglyphbounds #1 ~ #2 }
3805       { 1 em }
3806     )
3807   }
3808 }
3809 \ExplSyntaxOff
3810 \</macro>

```

下面是 CTeX 定义的一些字体。

```

3811 <*make>
3812 \input ctexpamacro %
3813
3814 \MAKESPA {ctexpunct.spa}
3815 {
3816   {adobezhhsong}      {AdobeSongStd-Light} ,
3817   {adobezhhei}        {AdobeHeitiStd-Regular} ,
3818   {adobezhkai}        {AdobeKaitiStd-Regular} ,
3819   {adobezhfs}         {AdobeFangsongStd-Regular} ,
3820   {fandolzhhsong}     {FandolSong} ,
3821   {fandolzhsongb}     {FandolSong-Bold} ,
3822   {fandolzhhei}       {FandolHei} ,
3823   {fandolzhheib}      {FandolHei-Bold} ,
3824   {fandolzhkai}       {FandolKai} ,
3825   {fandolzhfs}        {FandolFang} ,
3826   {founderzhhsong}    {FZShuSong-Z01} ,
3827   {founderzhsongb}    {FZXiaoBiaoSong-B05} ,
3828   {founderzhhei}      {FZHei-B01} ,
3829   {founderzhheil}     {FZXiHeiI-Z08} ,
3830   {founderzhkai}      {FZKai-Z03} ,
3831   {founderzhfs}       {FZFangSong-Z02} ,
3832   {founderzhli}       {FZLiShu-S01} ,
3833   {founderzhyou}      {FZXiYuan-M01} ,
3834   {founderzhyoub}     {FZZhunYuan-M02} ,
3835   {ubuntuzhsong}      {AR PL UMinG CN} ,
3836   {ubuntuzhhei}       {WenQuanYi Zen Hei} ,
3837   {ubuntuzhkai}       {AR PL UKai CN} ,
3838   {windowszhhsong}    {SimSun} ,
3839   {windowszhhei}      {SimHei} ,
3840   {windowszhkai}      {KaiTi} ,
3841   {windowszhfs}       {FangSong} ,
3842   {windowszhli}       {LiSu} ,
3843   {windowszhyou}      {YouYuan} ,
3844   {windowszhyawei}     {Microsoft YaHei} ,
3845   {windowszhyawei}     {Microsoft YaHei Bold}
3846 }
3847
3848 \primitive\end
3849 </make>
3850 </spa>

```

第 16 节 版本历史

v2.0

General: c5size, cs4size 是过时选项。 36

captiondelimiter 是过时选项。 94

fancyhdr 成为过时选项, 原选项功能总是打开。 39

fntef 成为过时选项, 原选项功能总是打开。 39

hyperref 成为过时选项, 原选项功能总是打开。 39

indent, noindent 是过时选项。 36

nofonts, adobefonts, winfonts 是过时选项。 37

nopunct 是过时选项。 38

nospace 是过时选项。 38

nozhmap 是过时选项。 38

punct 选项可以设置标点格式。 38

ctex 宏包新增 heading 选项。 38

\CTEXindent, \CTEXnoindent 是过时命令。 75

\CTEXsetup, \CTEXoptions 是过时命令。 15, 41

\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; CTEXfilltwosides 是过时环境。 69

标题设置新增 pagestyle 选项。 79

调整 \footnotesep 的大小, 以适合行距的变化。 94

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。 76

将中文字号功能提取到可以独立使用的 ctexsize。 90

默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。 69

删除 c19gsn.fd 和 c19gkai.fd。 1

通过 LuaTeX-ja 宏包支持 Lua \LaTeX 。 46

新增 autoindent 选项。 36

新增 fontset 选项。 37

新增 linespread 选项。 36

新增 scheme 选项, 并将 cap 和 nocap 列为过时选项 . . . 39

新增 zhmCJK 支持选项。 38

新增 zihao 选项。 36

新增统一设置接口 \ctexset。 41

应用 \LaTeX 3 重新整理代码。 1

中文字号不再采用近似值。 91

自动检测操作系统, 载入对应的字体配置。 66

\CJK@surr: 解决与 \nouppercase 的冲突。 42

\end: 解决 etoolbox 与 breqn 关于 \end 的冲突。 95

v2.0.1

General: 修复 10pt、11pt 等选项无效的问题。 39

第 17 节 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

_ 2458, 2460, 2473, 2474, 2476

\# 160, 2432, 2948

.../format+ 17

.../nameformat+ 17

.../numberformat+ 18

.../titleformat+ 18

\ / 104, 921, 1284, 1285

\< 103

\\ 13, 14, 31, 104, 181, 222, 224, 226, 491, 681, 1079, 1161, 2058, 2680, 2681, 2751, 2752, 2995, 3025

\{ 103, 1090, 1091

\} 1090, 1091

10pt 9, 39

11pt 9, 39

12pt 9, 39

A

\A 105, 106

\abovedisplayshortskip 2842, 2850, 2861, 2881, 2889, 2900

\abovedisplayskip 2841, 2844, 2849, 2856, 2860, 2867, 2880, 2883, 2888, 2895, 2899, 2906

\abstractname 2049

abstractname 13

\addCJKfontfeature 1161, 1184

\addCJKfontfeatures 56, 1179, 1184

\addcontentsline 2185, 2186, 2188, 2210, 2211, 2213, 2255, 2259, 2262, 2337, 2351

\addtocontents 2265, 2266

\addvspace 2171, 2178, 2265, 2266

adobefonts 27

AlternateFont 24, 58

\appendix 83, 2407, 2408

appendix/name 21

appendix/number 21

\appendixname 2051, 2672

appendixname 13

\apptocmd 156

\AtBeginDocument 584, 1579, 2927

\AtBeginDvi 527

\AtBeginShipoutFirst 529, 1691

\AtBeginUTFCommand 725

\AtEndOfClass 212

\AtEndOfPackage 213

\AtEndPreamble 419, 595, 1611

\AtEndUTFCommand 731

autoindent 12, 36, 72

B

\baselinestretch 1927

\begingroup	2329, 2364, 3447, 3481	\char_value_catcode:n	79, 557, 558
\belowdisplayshortskip	2843, 2851, 2862, 2882, 2890, 2901	CharRange	24, 58
\belowdisplayskip	2844, 2856, 2867, 2883, 2895, 2906	\Chinese	28, 2014
\bfdefault	942, 948, 1266, 1273	\chinese	23, 76, 2013, 2014, 2528, 2563
\bfseries	2194, 2197, 2220, 2223, 2232, 2244, 2287, 2292, 2310, 2519, 2524, 2531, 2537, 2541, 2544, 2552, 2559, 2566, 2577, 2581, 2588, 2601, 2614, 2620	\CJK	1720
\bibname	2053, 2058, 2061, 2062	\CJKbold	3067, 3068, 3078, 3079, 3085, 3086, 3088, 3089, 3091, 3092, 3096, 3097, 3099, 3100, 3102, 3103, 3109, 3110, 3112, 3113, 3115, 3116, 3120, 3121, 3123, 3124, 3126, 3127
bibname	13	\CJKfamily	54, 537, 539, 541, 543, 1016, 1038, 1217, 1219, 1221, 1223, 1369, 3435, 3436, 3437, 3438, 3440, 3441, 3443
bool commands:		\CJKfamilydefault	532, 533, 543, 598, 1212, 1213, 1223, 1230, 1232, 1236, 1243, 1256, 1591, 1595, 1603, 1617, 1619
\bool_gset_false:N	341, 346	\CJKfilltwosides	1728
\bool_gset_true:N	336	\CJKfontspec	56, 1174
\bool_if:NT	1119, 2009	\CJKglue	1754, 1765, 1779, 1788
\bool_if:NTF	76, 2684	\CJKhook	44, 613
\bool_if:nTF	112	\CJKnormal	3060, 3061, 3062, 3063, 3064, 3065, 3066, 3071, 3072, 3073, 3074, 3075, 3076, 3077, 3084, 3087, 3090, 3095, 3098, 3101, 3108, 3111, 3114, 3119, 3122, 3125
\bool_new:N	208, 1156	\CJKpunctmapfamily	646, 647, 648, 649, 650, 651, 655, 656, 657, 658, 659, 660
\bool_set_false:N	1131	\CJKrmdefault	524, 533, 534, 537, 1086, 1097, 1187, 1213, 1214, 1217, 1232, 1235, 1236, 1599, 3170, 3172, 3173, 3178, 3179, 3180, 3183, 3241, 3244, 3245, 3250, 3251, 3252, 3306, 3309, 3310, 3316, 3317, 3318, 3366, 3369, 3370, 3377, 3378, 3379, 3410, 3413, 3414, 3419, 3420, 3421
\bool_set_true:N	1149	\CJKsfdefault	525, 535, 539, 1087, 1098, 1193, 1215, 1219, 1600, 3146, 3160, 3161, 3184, 3242, 3307, 3311, 3367, 3371, 3411
box commands:		\CJKtilde	599
\box_ht:N	2922	\CJKttdefault	526, 536, 541, 1088, 1099, 1198, 1216, 1221, 1601, 3171, 3185, 3243, 3308, 3368, 3412
\box_new:N	27	clearalternatefont	25, 62
\box_wd:N	1755, 2428	clist commands:	
		\clist_clear:N	963, 3778
C		\clist_concat:NNN	985
c5size	27	\clist_const:Nn	626
cap	27	\clist_gput_right:Nn	440, 461, 463
captiondelimiter	28, 94	\clist_gput_right:No	452
\catcode	893, 3447, 3448, 3449, 3450, 3454, 3455	\clist_gset:Nn	1204
CCT	27	\clist_if_empty:NT	1291, 1547
CCTfont	27	\clist_if_empty:NTF	1307
\ccwd	23, 69, 189, 261, 1755, 1759, 1761, 1791, 1792, 1878, 1916, 1935, 1936, 1943, 1970, 1971, 1972, 1975, 1977, 1978, 1989, 2000, 2001, 2942, 2943, 2944	\clist_map_break:n	2689
\centering	2217, 2242, 2529, 2536, 2564, 2581	\clist_map_function:nN	1498, 1512
\chapter	2693	\clist_map_function:xN	1464, 1466
chapter/aftername	18	\clist_map_inline:Nn	616, 1400
chapter/afterskip	20	\clist_map_inline:nn	425, 638, 1505, 1711, 1738, 2381, 2686, 2769, 2804, 3767
chapter/beforeskip	19	\clist_new:N	215, 982, 1205, 1342, 3796
chapter/format	17	\clist_put_left:Nx	986
chapter/indent	20	\clist_put_right:Nn	1148
chapter/name	15	\clist_put_right:Nx	1388, 3790
chapter/nameformat	17	\clist_set:Nn	1325, 1326
chapter/number	16	\clist_set:No	967
chapter/numberformat	18	\contentsname	2044
chapter/pagestyle	20		
chapter/titleformat	18		
\chaptermark	2264		
\chaptername	2555		
char commands:			
\char_set_catcode:nn	79, 557, 558		
\char_set_catcode_active:N	1284		
\char_set_catcode_letter:n	73		
\char_set_catcode_math_superscript:n	552		
\char_set_catcode_other:N	106, 160, 197, 198, 2432, 2948		
\char_set_catcode_other:n	551, 734		
\char_set_lccode:nn	103, 104, 105, 738, 1285		

- contentsname 13
- cs commands:
 - \cs:w 2496
 - \cs_end: 2496
 - \cs_generate_variant:Nn 59, 65, 99, 153, 931, 1050, 1064, 1137, 1157, 1158, 1321, 1358, 1460, 1498, 1506
 - \cs_gset_eq:cc 1124, 1127
 - \cs_gset_eq:NN 170, 234, 235, 236, 337, 342, 347, 399, 401, 435, 439, 794, 798, 2733
 - \cs_gset_nopar:Npx 795
 - \cs_gset_protected_nopar:cpx 1012, 2385
 - \cs_gset_protected_nopar:Npn 742, 2408
 - \cs_gset_protected_nopar:Npx 833, 1362, 1456
 - \cs_if_exist:cF 789, 831, 869
 - \cs_if_exist:cTF 2483
 - \cs_if_exist:Nf 801, 1361, 1453, 1812, 2695
 - \cs_if_exist:NTF 2693
 - \cs_if_exist_p:c 112
 - \cs_if_exist_use:c 1034, 1035
 - \cs_if_exist_use:cF 772
 - \cs_if_exist_use:cT 62, 1475
 - \cs_if_exist_use:cTF 1056
 - \cs_if_exist_use:NT 2929
 - \cs_if_free:cF 618, 633, 1121
 - \cs_if_free:cTF 1265
 - \cs_if_free:Nf 756, 1905
 - \cs_if_free:NTF 1891
 - \cs_new:Npn 151, 184, 191, 202, 959, 1037
 - \cs_new_eq:cc 1747
 - \cs_new_eq:cN 483, 484, 485, 496, 1733
 - \cs_new_eq:Nc 1799
 - \cs_new_eq:NN 34, 36, 38, 207, 210, 211, 229, 603, 664, 810, 835, 1184, 1190, 1613, 1734, 1735, 1788, 1798, 2013, 2014, 2015, 2379, 2407, 2501, 2696, 3772
 - \cs_new_nopar:Npn 850, 1029, 1082, 1093, 3797
 - \cs_new_protected:Npn 48, 54, 68, 100, 110, 123, 154, 155, 156, 157, 163, 177, 735, 960, 979, 1289, 1377, 1391, 1542, 1793, 2144, 2422, 2824, 2830, 3773
 - \cs_new_protected_nopar:cpx 1717
 - \cs_new_protected_nopar:Npn 60, 83, 85, 212, 213, 522, 547, 564, 605, 611, 614, 631, 636, 644, 653, 662, 770, 779, 792, 797, 799, 811, 825, 852, 867, 880, 924, 937, 983, 991, 1010, 1030, 1040, 1065, 1103, 1115, 1117, 1139, 1141, 1226, 1251, 1260, 1305, 1316, 1343, 1352, 1359, 1365, 1385, 1398, 1417, 1431, 1438, 1450, 1470, 1486, 1504, 1507, 1509, 1522, 1534, 1568, 1584, 1589, 1622, 1640, 1659, 1670, 1674, 1724, 1730, 1751, 1762, 1790, 1890, 1892, 1917, 1952, 1962, 1967, 2081, 2107, 2377, 2479, 2488, 2493, 2742, 2757, 2789, 2794, 2967, 2971, 3764, 3788
 - \cs_new_protected_nopar:Npx 1170
 - \cs_set_eq:cc 620, 634
 - \cs_set_eq:cN 1806
 - \cs_set_eq:Nc 2378
 - \cs_set_eq:NN 172, 262, 267, 272, 544, 562, 582, 608, 612, 757, 804, 805, 957, 1016, 1224, 1361, 1369, 1379, 1380, 1381, 1453, 1454, 1455, 1594, 2020, 2025, 2028, 2033
 - \cs_set_nopar:cpx 1748
 - \cs_set_protected_nopar:Npn 607, 1765, 2831
 - \cs_set_protected_nopar:Npx 1795
 - \cs_to_str:N 101, 1169
 - \cs_undefine:c 63, 995, 996, 1000, 1001
 - \cs_undefine:N 173, 1382
- cs4size 27
- csname 897, 899, 901, 917, 2318, 2319, 2334, 2336, 2339, 2341, 2342, 2349, 2350, 2353, 2355, 2356, 2955, 2958, 3479, 3482, 3486
- csuse 2957, 2961
- CTeX 23, 94, 2925, 2930
- ctex commands:
 - \ctex_add_hook:cn 31, 52
 - \ctex_add_hook:Nn 31, 54, 59, 613
 - \ctex_add_to_selectfont:n 71, 1793, 1818, 1904
 - \ctex_appto_cmd:NnTF 33, 156, 1581
 - \ctex_assign_heading_name:nn 78, 2113, 2141, 2404
 - __ctex_assign_heading_name:nnn 78, 2143, 2144
 - \ctex_at_end:n 35, 212, 213, 369, 1788, 3034
 - \ctex_at_end_package:nn 31, 31, 48, 528, 754, 759, 1685, 1687, 2442, 2463, 2507, 2953
 - \ctex_auto_ignorespaces: 44, 603, 612, 1835
 - \l__ctex_autoindent_tl 35, 209, 261, 266, 271, 1878, 1882, 1885, 1954, 1957
 - __ctex_calc_bounds:nn 3792, 3793, 3797
 - __ctex_ccglue: 1779, 1788
 - \l__ctex_ccglue_skip 1766, 1770, 1773, 1775, 1783, 1928, 1939, 1948, 1984, 1992
 - \ctex_CJK_input:n 42, 547, 562
 - \c__ctex_class_tl 90, 92, 466, 471, 476, 2689, 2691, 2697, 2699, 2700, 2702
 - \ctex_declare_math_sizes:nnnn 91, 2789, 2823
 - __ctex_def_heading_keys:n 78, 2107, 2155
 - \ctex_default_pt:n 34, 187, 191
 - __ctex_default_pt:w 193, 202
 - \ctex_detected_platform: .. 66, 1622, 1640, 2975, 2984
 - \l__ctex_encoding_tl 34, 204, 205, 297, 298, 515, 518, 590, 598, 1681, 1863, 1867, 2011, 2728
 - \c__ctex_engine_file_tl 1646, 1656
 - \l__ctex_family_default_init_tl 66, 1591, 1612, 1614, 1619
 - __ctex_family_default_wrap:n 1594, 1613, 1616
 - \ctex_file_input:n 32, 83, 455, 665, 1864, 1869, 1870, 2702, 2729, 2730, 2833, 2835, 2976, 2980, 2988, 3034, 3133, 3134
 - \ctex_file_wrapper:nnn 31, 68, 84, 87, 549
 - __ctex_fix_varioref_label:n 85
 - \ctex_fix_varioref_label:n 2491, 2493
 - \g__ctex_font_size_flag 35, 229, 234, 235, 236, 435, 439, 460, 2732, 2733, 2832
 - \c__ctex_font_size_prop 91, 2744, 2755, 2761, 2799, 2826
 - \c__ctex_font_size_seq 2753, 2756, 2767

`\l__ctex_font_size_tl` 2744, 2745, 2791, 2792, 2826, 2827
`\ctex_fontset_error:n`
 95, 2967, 3220, 3254, 3275, 3320, 3339, 3381, 3423
`\g__ctex_fontset_tl` ... 301, 1624, 1652, 1662, 1663,
 2973, 2976, 2979, 2980, 2986, 2988, 3008, 3010, 3015, 3032
`__ctex_get_font_sizes:Nn` 91, 2791, 2794
`\l__ctex_heading_bool` 380, 2684
`\c__ctex_headings_seq` 77, 2074, 2075, 2152, 2490
`\ctex_hypersetup:n`
 67, 67, 1670, 1674, 1677, 1680, 1683, 1698, 1701
`\ctex_if_autoindent_touched:F`
 35, 210, 262, 267, 272, 2736
`\ctex_if_ccglue_touched:` 1776
`\ctex_if_ccglue_touched:TF` 70, 1923
`\ctex_if_ccglue_touched_p:` 70
`\ctex_if_macosx:TF` 67, 1647, 1648, 1650, 1659
`\ctex_if_pdfmode:TF` 30, 36, 38, 1684, 3219, 3274
`\ctex_if_preamble:` 40
`\ctex_if_preamble:TF` 31, 3003
`\ctex_ignorespaces_case:N` .. 44, 605, 1832, 1835, 1838
`\l__ctex_indent_bool` 2009
`__ctex_initial_heading:n` 77, 2081, 2154
`\l__ctex_line_spread_fp` 93, 254, 2734, 2918, 2920
`\l__ctex_line_stretch_tl` 1910, 1915, 1916, 1920
`\ctex_load_fontset:` 95, 2971, 2992, 3011, 3033
`\ctex_load_zhmap:n` 531
`\ctex_load_zhmap:nnnn`
 42, 522, 3176, 3248, 3314, 3375, 3417
`\l__ctex_ltj_add_alternate_bool` 1119, 1131, 1149, 1156
`\ctex_ltj_add_font_features:n` 56, 1139, 1157
`\ctex_ltj_add_font_features:nn` 56, 1141, 1158
`\ctex_ltj_add_font_features:x` 1181
`\ctex_ltj_add_font_features:xn` 1140
`__ctex_ltj_alternate_cs:n`
 996, 999, 1000, 1001, 1034, 1035, 1037, 1122,
 1125, 1126, 1128, 1129, 1311, 1348, 1410, 1435, 1475, 1492
`\l__ctex_ltj_alternate_family_tl` 1373, 1376
`\l__ctex_ltj_alternate_options_clist`
 1326, 1327, 1338, 1342
`\l__ctex_ltj_alternate_seq`
 964, 1298, 1299, 1304, 1310, 1318, 1566
`\l__ctex_ltj_base_CJKfamily_tl`
 965, 981, 1032, 1410, 1435
`\l__ctex_ltj_base_family_tl` 62,
 997, 1027, 1033, 1354, 1395, 1454, 1473, 1478, 1479, 1491
`__ctex_ltj_change_encoding:` 52, 937, 1015, 1368
`\l__ctex_ltj_char_range_clist` 963,
 1280, 1291, 1307, 1325, 1337, 1349, 1372, 1394, 1545, 1547
`\ctex_ltj_char_range_key:nn` 64, 1539, 1542
`__ctex_ltj_char_range_parse_feature:w`
 64, 1550, 1554
`\g__ctex_ltj_char_range_prop` ... 63, 1402, 1513, 1518
`\l__ctex_ltj_char_range_tl`
 1402, 1405, 1511, 1513, 1515, 1517, 1524
`__ctex_ltj_check_family:n` 53, 991, 1309
`\ctex_ltj_clear_alternate_font:n` 1464, 1470
`\l__ctex_ltj_current_font_tl`
 48, 772, 776, 777, 789, 831, 836
`\l__ctex_ltj_current_shape_tl` 854, 859, 864, 865
`\ctex_ltj_declare_alternate_shape:nnnnn`
 61, 1371, 1391, 1393
`\ctex_ltj_declare_char_range:n` 1504, 1506
`__ctex_ltj_declare_char_range:nn` 1505, 1507
`\ctex_ltj_declare_char_range:nn` 63, 1508, 1509
`\ctex_ltj_declare_char_range:x` 1501
`\ctex_ltj_def_char_range_key:n` 64, 1514, 1534
`\g__ctex_ltj_default_features_clist` . 985, 1204, 1205
`\ctex_ltj_ensure_default_family:` ... 57, 1226, 1608
`\ctex_ltj_extract_font:` 49, 804, 811
`__ctex_ltj_family_csname:n` ... 995, 1012, 1029, 1056
`\g__ctex_ltj_family_font_name_prop`
 53, 970, 989, 993, 1067, 1143, 1228, 1240
`\g__ctex_ltj_family_font_options_prop`
 53, 971, 990, 1146
`\ctex_ltj_family_if_exist:nTF` 54
`\ctex_ltj_family_if_exist:xN` 1051
`\ctex_ltj_family_if_exist:xNF` 1230
`\ctex_ltj_family_if_exist:xNT` 1256
`\ctex_ltj_family_if_exist:xNTF` 1042, 1235, 1253, 1473
`\g__ctex_ltj_family_int` . 1109, 1111, 1387, 1388, 1390
`\g__ctex_ltj_family_name_prop`
 53, 988, 997, 1019, 1053, 1354
`\l__ctex_ltj_family_tl`
 1044, 1049, 1106, 1107, 1122, 1126, 1129, 1140, 1467, 1468
`__ctex_ltj_family_unknown_warning:n`
 54, 1047, 1065, 1483
`\l__ctex_ltj_font_options_clist` 967,
 972, 973, 974, 982, 1018, 1147, 1148, 1151, 1345, 1346, 1350
`\ctex_ltj_fontspec:nn` 55, 1103, 1116
`__ctex_ltj_fontspec:nnn` 1117, 1137
`__ctex_ltj_fontspec:xnn` 1110
`\ctex_ltj_fontspec:xx` 1115, 1150, 1176
`\g__ctex_ltj_fontspec_family_tl`
 1017, 1020, 1021, 1022, 1028, 1033, 1058
`\g__ctex_ltj_fontspec_prop` 1105, 1133, 1138
`__ctex_ltj_gset_family_cs:nn` 53, 975, 1010
`\ctex_ltj_if_alternate_shape_exist:n` 845
`\ctex_ltj_if_alternate_shape_exist:nF` 827
`\ctex_ltj_if_alternate_shape_exist:nT` 814, 854
`\ctex_ltj_if_alternate_shape_exist:nTF` 50
`\l__ctex_ltj_jfm_tl` 52, 927, 928, 930, 941, 943, 986
`\c__ctex_ltj_math_fam_int` 949, 950
`\c__ctex_ltj_math_family_tl`
 1262, 1264, 1266, 1269, 1273
`\c__ctex_ltj_math_tl`
 944, 945, 947, 949, 1202, 1253, 1263, 1268, 1272
`__ctex_ltj_msg_def_family_map:n` 1080, 1082
`__ctex_ltj_msg_family_map:n` 1009, 1079, 1093
`__ctex_ltj_noembed_wrap:n` 957, 959
`__ctex_ltj_patch_external_font:w` 50, 817, 850
`\ctex_ltj_pickup_font:` 48, 785, 799, 810, 874
`__ctex_ltj_pop_fontname:` 788, 797, 876

- _ctex_ltj_push_fontname:n 784, 792, 873
- \ctex_ltj_reset_altername_font:n 1466, 1486
- \g_ctex_ltj_reset_altername_prop
..... 1002, 1477, 1489, 1497
- \ctex_ltj_save_altername_family:Nnnn
..... 60, 1336, 1356, 1359
- \ctex_ltj_save_altername_seq:cn 1311
- \ctex_ltj_save_altername_seq:Nn 59, 1316, 1321
- \ctex_ltj_save_altername_seq:Nnnwn 59
- \ctex_ltj_save_altername_seq:Nnnwnw 1319, 1322
- _ctex_ltj_save_altername_shape:cn 1409, 1434
- _ctex_ltj_save_altername_shape:Nnn 62, 1450, 1460
- \ctex_ltj_save_char_range:n 63, 1512, 1519
- \ctex_ltj_save_char_range:nn 1521, 1522
- _ctex_ltj_save_fontname:w 794, 798
- \ctex_ltj_select_altername_font: 50, 852, 1821
- \ctex_ltj_select_font: 48, 770, 1820
- _ctex_ltj_select_font_aux: 48, 773, 779, 790
- \ctex_ltj_set_altername_family:coonn 1347
- _ctex_ltj_set_altername_family:n 1023, 1030
- \ctex_ltj_set_altername_family:nn .. 60, 1314, 1343
- \ctex_ltj_set_altername_family:nnF ... 59, 968, 1305
- \ctex_ltj_set_altername_family:nnn
..... 60, 1355, 1363, 1365
- \ctex_ltj_set_altername_family:Nnnnn ... 1352, 1358
- \ctex_ltj_set_altername_seq:n 58, 1278, 1289
- \ctex_ltj_set_altername_shape:n ... 1419, 1431, 1442
- \ctex_ltj_set_altername_shape:nnN .. 62, 1404, 1438
- \ctex_ltj_set_altername_shape:nnn .. 61, 1407, 1414
- \ctex_ltj_set_altername_shape:nnnn 1416, 1417
- \ctex_ltj_set_altername_shape:Nnnnnnn 61, 1394, 1398
- \ctex_ltj_set_family:nnn 52, 960, 980, 1134
- \ctex_ltj_set_family:xxx
..... 979, 1165, 1172, 1187, 1193, 1198, 1202
- \ctex_ltj_subst_font: 49, 805, 825
- \ctex_ltj_swap_cs:NN 60, 1370, 1377
- \ctex_ltj_switch_family:n 1040, 1050, 1135, 1171
- \ctex_ltj_switch_family:x 1039, 1107
- _ctex_ltj_tmp:w 1379, 1381, 1382
- \l_ctex_ltj_tmp_seq 1295, 1296, 1299, 1303
- \l_ctex_ltj_tmp_tl 966, 967, 993, 1004, 1007,
1144, 1152, 1168, 1171, 1172, 1253, 1254, 1256, 1257,
1293, 1294, 1295, 1329, 1331, 1332, 1339, 1549, 1557, 1566
- \l_ctex_ltj_tmpa_tl 1230, 1235
- _ctex_ltj_um_char:Nw 735, 757
- \g_ctex_ltj_unknown_family_seq ... 1069, 1071, 1076
- _ctex_ltj_update_family_uid:N . 61, 973, 1345, 1385
- \ctex_ltj_update_mathfont: 58, 1248, 1251
- \ctex_ltj_update_mathfont:n 1254, 1257, 1260
- \ctex_ltj_update_xkanjiskip: 65, 1568, 1822
- _ctex_ltj_use_global_options:N 974, 983, 1327, 1346
- \l_ctex_ltj_xkanjiskip_skip
..... 65, 1571, 1573, 1574, 1577, 1578
- \ctex_lua_now_x:n 30, 34,
728, 746, 821, 847, 851, 856, 861, 883, 936, 1433, 1457, 1626
- \c_ctex_macosx_file_tl 1661, 1665
- \ctex_make_spa:nn 110, 3764, 3772
- \ctex_mono_jfm:n 52, 924, 931
- \ctex_mono_jfm:o 932, 1855
- \ctex_package_end_hook:n 31, 60, 65
- \ctex_package_end_hook:o 31, 67
- \ctex_parse_name:NN 32, 100, 154, 155, 156
- \ctex_parse_name:NNn 110, 153
- _ctex_parse_name:nNNnN 117, 123
- \ctex_parse_name:NNx 101
- _ctex_parse_name:w 136, 142, 151
- _ctex_patch_cmd:Nnn 161, 163
- \ctex_patch_cmd:Nnn 33, 157, 2451,
2453, 2457, 2460, 2466, 2468, 2472, 2475, 2942, 2943, 2944
- \ctex_patch_cmd:NnnTF 33, 154, 168, 2955
- \ctex_patch_failure:N 175, 177, 538, 540, 542,
545, 1218, 1220, 1222, 1225, 1582, 1808, 2438, 2445, 2962
- \ctex_plane_to_utfxvibe:Nn 42, 564, 582
- \ctex_preto_cmd:NnTF 33, 155, 537, 539,
541, 543, 1217, 1219, 1221, 1223, 1803, 2437, 2444, 2961
- \l_ctex_punct_bounds_clist ... 3778, 3784, 3790, 3796
- \c_ctex_punct_family_clist 616, 626
- \l_ctex_punct_font 3776, 3777
- \ctex_punct_map_bfseries:nn
..... 45, 636, 3161, 3173, 3179, 3245,
3251, 3310, 3311, 3317, 3370, 3371, 3372, 3378, 3414, 3420
- \ctex_punct_map_family:nn ... 45, 631, 3146, 3160,
3170, 3171, 3178, 3241, 3242, 3243, 3250, 3306, 3307,
3308, 3316, 3366, 3367, 3368, 3377, 3410, 3411, 3412, 3419
- \ctex_punct_map_itshape:nn 45, 653,
3172, 3180, 3244, 3252, 3309, 3318, 3369, 3379, 3413, 3421
- \ctex_punct_map_series:nnn 640, 641, 644
- \c_ctex_punct_seq 3755, 3756, 3779
- \ctex_punct_set:n 44, 614,
3141, 3177, 3240, 3249, 3305, 3315, 3365, 3376, 3409, 3418
- \ctex_punct_space:nn 45, 662, 664
- \l_ctex_punct_tl
..... 356, 586, 587, 673, 932, 1852, 1853, 1854, 1855
- _ctex_save_bounds:n 3780, 3788
- _ctex_save_font_size:nn 91, 2757, 2788
- _ctex_save_selectfont: 1799, 1806
- \ctex_scheme_input:n 32, 85, 99
- \ctex_scheme_input:o 2509, 2704
- \l_ctex_scheme_tl 381, 2509, 2704
- \g_ctex_section_depth_flag
..... 34, 207, 399, 401, 2623, 2643, 2663
- \ctex_select_size: .. 1879, 1886, 1889, 1890, 1911, 1965
- \ctex_set_default_ccwd:Nn ... 34, 184, 271, 1885, 1910
- \ctex_set_font_size:Nnn 92, 2824, 2839,
2847, 2858, 2869, 2870, 2871, 2872, 2873, 2874, 2875,
2878, 2886, 2897, 2908, 2909, 2910, 2911, 2912, 2913, 2914
- _ctex_set_font_size:nnNn 2827, 2830
- \ctex_set_ignorespaces: 44, 607, 609, 611, 613
- \g_ctex_spa_iow 3766, 3769, 3771, 3781
- \g_ctex_std_options_clist
..... 35, 215, 440, 452, 461, 463, 467, 472, 477
- _ctex_tmp:w 170, 172, 173

<code>\l__ctex_tmp_box</code>	27, 30, 1754, 1755, 2424, 2428
<code>\l__ctex_tmp_dim</code>	26, 30, 1920, 1921, 1926, 1929, 1935, 1974, 1976, 1978, 1987, 1989
<code>\l__ctex_tmp_int</code>	25, 30, 567, 569, 570, 572, 576, 578, 1934, 1936, 1937, 1943, 1944, 1976, 1977, 1978, 1981, 1982, 1987, 1989, 2380, 2384, 2388
<code>\l__ctex_tmp_tl</code>	24, 30, 424, 431, 446, 447, 481, 483, 484, 485, 496, 566, 568, 2109, 2151, 2157, 2158, 2799, 2800
<code>\ctex_update_ccglue:</code>	70, 1762, 1930, 1949, 1993
<code>\ctex_update_ccwd:</code>	69, 1751, 1924, 2001
<code>\ctex_update_default_family:</code>	65, 1589, 1611
<code>\ctex_update_em_unit:</code>	70, 1790, 1919, 1969
<code>\ctex_update_parindent:</code>	74, 1897, 1952, 1994
<code>\ctex_update_size:</code>	73, 1891, 1892, 1905
<code>\ctex_update_stretch:</code>	73, 1896, 1917
<code>\ctex_update_ziju:</code>	74, 1899, 1967
<code>\ctex_varioref_hook:</code>	85, 2488, 2499, 2507
<code>__ctex_write_family:nn</code>	3768, 3773
<code>\c__ctex_zero_tl</code>	1894, 1901, 1903
<code>\ctex_zhmap_case:nnn</code>	35, 211, 337, 342, 347, 3139, 3222, 3277, 3353, 3400
<code>\g__ctex_zhmCJK_bool</code>	35, 208, 336, 341, 346, 514, 589
<code>\ctex_zihao:n</code>	90, 2741, 2742
<code>\ctex_ziju:n</code>	1961, 1962
<code>\l__ctex_ziju_dim</code> 1970, 1971, 1975, 1986, 1988, 1992, 1996	
<code>\l__ctex_ziju_tl</code>	1894, 1902, 1903, 1964, 1970
<code>\ctexbibname</code>	2059, 2068
<code>\CTEXcounter</code>	2015
<code>\CTEXdigits</code>	23, 76, 2018
<code>\CTEXfilltwosides</code>	28, 1724, 1734
<code>\CTEXindent</code>	28, 75, 1997, 1999
<code>\CTEXnoindent</code>	28, 75, 2003, 2005
<code>\CTEXnumber</code>	23, 76, 2016
<code>\CTEXoptions</code>	28, 41, 506, 508
<code>\ctexset</code>	6, 41, 498, 503, 509
<code>\CTEXsetfont</code>	23, 73, 1889
<code>\CTEXsetup</code>	28, 41, 500, 502
<code>\CTEXsout</code>	28
<code>\ctexspadef</code>	45, 664, 3783
<code>\CTEXthechapter</code>	16, 2254, 2257, 2458, 2473, 2474
<code>\CTEXtheparagraph</code>	16
<code>\CTEXthepart</code>	16, 2186, 2211
<code>\CTEXthesection</code>	16, 2451, 2460, 2467, 2476
<code>\CTEXthesubparagraph</code>	16
<code>\CTEXthesubsection</code>	16, 2453, 2469
<code>\CTEXthesubsubsection</code>	16
<code>\CTEXunderdblline</code>	28
<code>\CTEXunderdot</code>	28
<code>\CTEXunderline</code>	28
<code>\CTEXunderwave</code>	28
<code>\CTEXxout</code>	28
<code>\CurrentOption</code>	452
D	
<code>declarecharrange</code>	24, 63
<code>\DeclareFontEncoding</code>	934
<code>\DeclareFontFamily</code>	939, 3052, 3053, 3054, 3055, 3056, 3057
<code>\DeclareFontShape</code>	940, 942, 3060, 3061, 3062, 3063, 3064, 3065, 3066, 3067, 3068, 3071, 3072, 3073, 3074, 3075, 3076, 3077, 3078, 3079, 3084, 3085, 3086, 3087, 3088, 3089, 3090, 3091, 3092, 3095, 3096, 3097, 3098, 3099, 3100, 3101, 3102, 3103, 3108, 3109, 3110, 3111, 3112, 3113, 3114, 3115, 3116, 3119, 3120, 3121, 3122, 3123, 3124, 3125, 3126, 3127
<code>\DeclareFontSubstitution</code>	935
<code>\DeclareMathSizes</code>	2792
<code>\DeclareSymbolFont</code>	945, 1263
<code>\def</code>	890, 896, 909, 915, 2182, 2206, 2228, 2240, 2249, 2273, 2297, 2316, 2320, 2346, 2361, 2372, 2373, 2852, 2863, 2891, 2902, 3453, 3483
<code>\defaultCJKfontfeatures</code>	57, 1203, 1211
dim commands:	
<code>\dim_abs:n</code>	1988
<code>\dim_add:Nn</code>	1971
<code>\dim_compare:nNnF</code>	1956, 1979
<code>\dim_compare:nNnTF</code>	186, 1921, 1972
<code>\dim_eval:n</code>	1941, 1987
<code>\dim_max:nn</code>	2427
<code>\dim_min:nn</code>	1988
<code>\dim_new:N</code>	26, 1761, 1996
<code>\dim_set:Nn</code>	1755, 1791, 1792, 1920, 1926, 1957, 1970, 1974, 2001, 2425, 2922
<code>\dim_sub:Nn</code>	1978
<code>\dim_to_decimal:n</code>	34, 2763, 2764, 2801
<code>\dim_to_decimal_in_unit:nn</code>	3803
<code>\dim_use:N</code>	194, 2922
<code>\dim_zero:N</code>	2007
<code>\directlua</code>	711, 712, 891
E	
<code>\E</code>	106
<code>\edef</code>	903, 3472, 3473
<code>\else</code>	912, 917, 919, 2175, 2187, 2212, 2258, 2261, 2269, 2279, 2303, 2323, 2338, 2345, 2352, 2371, 3525, 3564, 3595, 3654, 3717
else commands:	
<code>\else:</code>	37, 44, 517, 848, 1780, 2926
<code>\end</code>	95, 2955, 2961, 2962, 3848
<code>\endCJKfilltwosides</code>	1730
<code>\endcsname</code>	897, 899, 901, 917, 2318, 2319, 2334, 2336, 2339, 2341, 2342, 2349, 2350, 2353, 2355, 2356, 2955, 2958, 3479, 3482, 3486
<code>\endCTEXfilltwosides</code>	1730, 1735
<code>\endgroup</code>	2335, 2370, 3472, 3489
<code>\endinput</code>	3477
<code>\endlinechar</code>	3451, 3452
etex commands:	
<code>\etex_dimexpr:D</code>	194, 1935
<code>\etex_glueexpr:D</code>	1927
<code>\EverySelectfont</code>	1812, 1815

exp commands:

\exp_after:wN	193, 801, 817, 819, 1654, 1656, 1657, 2745, 2792, 2827
\exp_args:cc	2484
\exp_args:Nc	126
\exp_args:Nnc	85
\exp_args:NNf	1557
\exp_args:NNNo	2921
\exp_args:Nno	2496
\exp_args:No	568, 3780
\exp_args:Nx	369, 595, 669, 1961, 2741
\exp_last_unbraced:Nf	136
\exp_not:c	118, 836, 1719, 1721, 1748, 2087, 2088, 2089, 2093, 2094, 2095, 2096, 2099, 2100, 2103, 2104, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2124, 2126, 2128, 2130, 2132, 2389, 2390, 2391, 2394
\exp_not:N	118, 597, 598, 599, 763, 764, 766, 872, 1019, 1020, 1021, 1022, 1336, 1457, 1599, 1600, 1601, 1616, 1720, 2095, 2387, 2394, 2761, 2986
\exp_not:n	75, 134, 1016, 1017, 1594
\exp_not:o	446, 765, 767, 1018, 1151, 1152, 1337, 1338, 1339, 1363, 1457, 1617, 1796, 2157, 2485
\expandafter	917, 918, 920, 3454, 3455, 3479, 3482, 3486, 3487
\ExplSyntaxOff	72, 78, 167, 693, 889, 1802, 2436, 2952, 3809
\ExplSyntaxOn	77, 722, 923, 1810, 2448, 2965, 3748

F

\familydefault	1597
fancyhdr	28, 39
\fangsong	3437
\fi	912, 918, 919, 920, 2169, 2177, 2189, 2196, 2214, 2222, 2260, 2263, 2272, 2281, 2289, 2290, 2305, 2326, 2343, 2357, 2359, 2374, 2473, 3488, 3741

fi commands:

\fi:	39, 46, 464, 546, 583, 601, 848, 1780, 2454, 2461, 2836, 2926
\figurename	2047
figurename	13

file commands:

\g_file_current_name_tl	685, 1657
\file_if_exist:nTF	1642, 1644, 1646, 1661, 2979, 3132, 3149
\file_if_exist_input:nF	92
\file_input:n	84, 93, 95, 527, 529, 555

five commands:

\c_five	2764
fntef	28, 39
\font	3052, 3053, 3054, 3055, 3056, 3057

font@name commands:

\font@name:	906
fontset	8, 37, 96
\fontsize	2745

fontspec commands:

\g_fontspec_encoding_tl	938
\l_fontspec_family_tl	1396
_fontspec_namewrap:n	52, 957

\fontspec_set_family:Nnn	1017, 1373
\footnotesep	94, 2922
\footnotesize	2858, 2897, 2921

four commands:

\c_four	401, 576, 578
---------------	---------------

fp commands:

\fp_compare:nNnF	2918
\fp_compare:nNnT	2734
\fp_eval:n	1964, 3799
\fp_use:N	1901, 2920

G

GBK	7, 37
\gdef	894, 2412, 2413, 2414, 2417, 2418, 2419
\globaljfont	819

group commands:

\group_begin:	102, 114, 159, 166, 196, 727, 733, 737, 781, 803, 829, 871, 962, 1014, 1283, 1367, 1452, 1472, 1488, 1593, 2098, 2431, 2921, 2947, 3775
\group_end:	109, 125, 165, 171, 175, 201, 731, 741, 753, 786, 807, 842, 875, 977, 1024, 1288, 1374, 1458, 1484, 1495, 1605, 2101, 2435, 2921, 2951, 3786
\group_insert_after:N	839, 840

H

hbox commands:

\hbox_set:Nn	1754, 2424
heading	10, 38
\heiti	3436
\hskip	2331, 2347, 2366, 2372, 2373, 2467, 2469
\hspace	2185, 2186, 2210, 2211, 2257
\Huge	2223, 2244, 2292, 2310, 2541, 2559, 2875, 2914
\huge	2197, 2220, 2232, 2287, 2524, 2537, 2544, 2552, 2566, 2874, 2913
hyperref	28, 39
\hypersetup	1671
\hyphenchar	3052, 3053, 3054, 3055, 3056, 3057

I

if commands:

\if_bool:N	514, 589
\if_case:w	460, 2832
\if_cs_exist:N	563
\if_meaning:w	42, 1779
\ifdim	2173, 2277, 2301, 2328, 2363
\ifincsname	2926
\ifmmode	915
\IfNoValueF	504, 510
\IfNoValueTF	1169, 1421, 1526, 1559, 1561, 2147
\ifnum	2183, 2193, 2208, 2219, 2250, 2285, 2321, 2338, 2352
\ifpdf	35, 3494, 3561, 3592, 3623, 3690
\ifx	917, 3482
\immediate	3485
indent	27, 36
\indexname	2050
indexname	13
\input	3480, 3747, 3812

int commands:		linespread	11, 36
\int_compare:nNnF	1981, 2732	linestretch	12, 73
\int_compare:nNnTF	569, 1937, 2623, 2643, 2663	\linewidth	1935, 1943, 1975
\int_const:Nn	949	\lishu	3440
\int_div_truncate:nn	73, 576	\listfigurename	2045
\int_eval:n	1422, 1424, 1425, 1527, 1529, 1530	listfigurename	13
\int_from_hex:n	568	\listtablename	2046
\int_gincr:N	1109, 1387	listtablename	13
\int_incr:N	2384	\LoadClass	468, 473, 478
\int_mod:nn	578	\long	3483
\int_new:N	25, 1390	\lstlistingname	764, 765
\int_set:Nn	559, 567, 1934, 1976	\lstlistlistingname	766, 767
\int_set_eq:NN	553	\ltjdefcharrange	694, 695, 696, 698, 702, 703, 706, 709
\int_sub:Nn	572, 1982	LTJFONTUID	61
\int_to_Hex:n	570, 575, 577	\ltjgetparameter	1571, 1759, 1783
\int_use:N	559, 1111, 1388, 2388, 3780	\ltjsetkanjiskip	1773
\int_zero:N	2380	\ltjsetmathletter	751
\interlinepenalty	2190, 2216, 2229, 2241, 2291, 2309, 2332, 2333, 2367, 2368	\ltjsetparameter	710, 716, 1585
iow commands:		\ltjsetxkanjiskip	1574
\iow_close:N	3769	lua commands:	
\iow_new:N	3771	\lua_now_x:n	30
\iow_now:Nx	3781	luatex commands:	
\iow_open:Nn	3766	\luatex_directlua:D	34
\itemsep	2855, 2866, 2894, 2905	\luatexluaescapestring	847, 859, 886, 892, 1412, 1427, 1428, 1445, 1446, 1455
J		M	
\jfam	919, 950	\MAKESPA	3772, 3814
K		\markboth	2199, 2215
\kaishu	3438	\mathgroup	919
kernel commands:		max commands:	
\l_kernel_expl_bool	76	\c_max_dim	1921
keys commands:		\mddefault	935, 940, 946, 1264, 1269
\l_keys_choice_tl	370	minus commands:	
\keys_define:nn	230, 446, 449, 954, 1276, 1384, 1461, 1499, 1538, 1827, 1848, 1872, 1906, 2021, 2042, 2062, 2065, 2068, 2157, 2160, 2400, 2402, 2932, 2999	\c_minus_one	229, 553, 2732
\keys_if_exist:nnF	1536	mode commands:	
\keys_if_exist:nnT	2719, 2722	\mode_if_math:TF	744
\l_keys_key_tl	218, 220, 222	msg commands:	
\keys_set:nn	26, 244, 250, 283, 294, 309, 318, 327, 353, 363, 370, 377, 388, 394, 499, 504, 510, 1716, 2512, 2549, 2571, 2585, 2598, 2611, 2617, 2625, 2637, 2645, 2657, 2664, 2665, 2666, 2707, 2721, 2723, 2731, 2735, 2737, 3041	\msg_critical:nnn	2968
\keys_set_known:nnN	966	\msg_critical:nnx	685
L		\msg_error:nn	495, 3019
\labelformat	2496	\msg_error:nnn	18, 20, 676, 689, 692, 2746, 2828
\LARGE	2873, 2912	\msg_error:nnx	2037
\Large	2194, 2519, 2531, 2577, 2581, 2872, 2911	\msg_error:nnxx	2985, 3014
\large	2588, 2871, 2910	\msg_fatal:nn	3753
\leavevmode	2169	\msg_new:nn	3751
\leftmargin	2852, 2863, 2891, 2902	\msg_new:nnn	179, 217, 219, 221, 223, 225, 679, 1008, 1077, 1159, 1707, 2056, 2678, 2969, 3028
\leftmargini	2852, 2863, 2891, 2902	\msg_new:nnnn	10, 28, 488, 2039, 2748, 2993, 3022
\let	895, 898, 900, 911, 912, 921, 2322, 2845, 2884, 3475	\msg_warning:nn	1154, 1709, 1845, 2067
\linespread	2920	\msg_warning:nnn	242, 248, 276, 288, 304, 313, 322, 351, 361, 375, 386, 392, 406, 411, 417, 1072, 2936, 3006
		\msg_warning:nnnn	502, 508, 1719, 1726, 1999, 2005
		\msg_warning:nnx	178, 2700
		\msg_warning:nnxx	1004

N

nan commands:

\c_nan_fp	255
\newCJKfontfamily	56, 1166
\NewDocumentCommand	498, 500, 506, 1038, 1164, 1166, 1174, 1179, 1185, 1191, 1196, 1201, 1203, 1322, 1414, 1519, 1554, 1889, 1960, 1997, 2003, 2016, 2018, 2141, 2740, 2925, 3435, 3436, 3437, 3438, 3440, 3441, 3443
\newfontfeature	951, 952, 953
nil commands:	
\q_nil	735
\nobreak	2194, 2200, 2234, 2287, 2292, 2293, 2310, 2311, 2523, 2558
\nobreakspace	2194, 2220
nocap	27
\noexpand	3473, 3475, 3476, 3477
nofonts	27
noindent	27
nopunct	27
\normalem	1704
\normalfont	543, 544, 545, 1188, 1194, 1199, 1223, 1224, 1225, 1247, 2191, 2192, 2217, 2218, 2230, 2231, 2242, 2243, 2283, 2284, 2307, 2308, 2394
\normalsize	2601, 2614, 2620, 2839, 2878, 2916, 2924
nospace	27
nozhmap	27
\numberline	84, 2256, 2257, 2339, 2340, 2353, 2354, 2437, 2438, 2444, 2445

O

\O	106
one commands:	
\c_one	235, 1415, 1520, 1989, 2142, 3792
or commands:	
\or:	462, 2834

P

\P	106, 197
\pagestyle	2727
pagestyle	79
\par	2170, 2194, 2199, 2220, 2223, 2224, 2232, 2233, 2244, 2245, 2287, 2292, 2293, 2310, 2311, 2523, 2538, 2558
paragraph/aftername	18
paragraph/afterskip	20
paragraph/beforeskip	19
paragraph/format	17
paragraph/indent	20
paragraph/name	15
paragraph/nameformat	17
paragraph/number	16
paragraph/numberformat	18
paragraph/titleformat	18
\parindent	1956, 1957, 1979, 2000, 2001, 2006, 2007, 2191, 2192, 2230, 2231, 2283, 2284, 2307, 2308, 2665
\parsep	2854, 2855, 2865, 2866, 2893, 2894, 2904, 2905
\part	79, 2168
part/aftername	18

part/afterskip	20
part/beforeskip	19
part/format	17
part/indent	20
part/name	15
part/nameformat	17
part/number	16
part/numberformat	18
part/pagestyle	20
part/titleformat	18
\partname	2194, 2220, 2515
\PassOptionsToClass	467, 472, 477
\PassOptionsToPackage	3, 515, 1675, 2011
\patchcmd	154
\pdfmapline	3495, 3496, 3497, 3498, 3499, 3500, 3501, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3511, 3512, 3513, 3514, 3515, 3516, 3518, 3519, 3520, 3521, 3522, 3523, 3624, 3625, 3626, 3627, 3628, 3629, 3630, 3632, 3633, 3634, 3635, 3636, 3637, 3638, 3640, 3641, 3642, 3643, 3644, 3645, 3647, 3648, 3649, 3650, 3651, 3652, 3691, 3692, 3693, 3694, 3695, 3696, 3698, 3699, 3700, 3701, 3702, 3703, 3705, 3706, 3707, 3708, 3709, 3711, 3712, 3713, 3714, 3715
\pdfstringdefDisableCommands	2929
pdftex commands:	
\pdfTeX_if_engine:TF	206, 1861, 3137, 3217, 3272, 3338, 3351, 3398
\pretocmd	155
prg commands:	
\prg_do_nothing:	1361, 1453, 1735, 1798, 1832
\prg_new_conditional:Npnn	40, 845, 1776
\prg_new_protected_conditional:Npnn	1051
\prg_return_false:	43, 848, 1061, 1780, 1784
\prg_return_true:	45, 848, 1054, 1059, 1780, 1784
\primitive	3848
\ProcessKeysOptions	456
prop commands:	
\prop_get:NnN	1146
\prop_get:NnNT	1354
\prop_get:NnNTF	1064, 1105, 1143, 1402, 2744, 2799, 2826
\prop_get:NxNTF	1053
\prop_gpop:NnNT	993, 997, 1489
\prop_gput:Nnn	970, 1133, 2761
\prop_gput:Nno	971, 1019, 1477, 1513
\prop_gremove:Nn	1002
\prop_if_empty:NF	1067, 1228
\prop_map_break:n	1242
\prop_map_inline:Nn	1240
\prop_new:N	988, 989, 990, 1138, 1497, 1518, 2755
\protect	130, 131, 2256, 2257, 2265, 2266, 2339, 2340, 2353, 2354
\ProvidesFile	3483, 3491, 3558, 3589, 3620, 3687
punct	10, 27, 38, 72
\punctstyle	587, 1853

Q

\quad 2318, 2458, 2460, 2467, 2469,
2473, 2474, 2476, 2530, 2565, 2574, 2589, 2602, 2615, 2621
\quotation 94, 2944

R

\raggedright 2191, 2230, 2283, 2307, 2522, 2557
\refname 2052, 2058, 2064, 2065
\refstepcounter 85, 2184, 2209, 2252, 2324, 2501, 2502
\relax 894, 915, 918, 919,
2172, 2208, 2219, 2276, 2300, 2325, 2327, 2331, 2347,
2362, 2372, 2373, 2467, 2469, 3447, 3451, 3454, 3480, 3482
\renewcommand 2168
\RenewDocumentCommand 2502
\RequirePackage .. 7, 23, 413, 419, 516, 519, 520, 521, 668,
687, 690, 724, 732, 763, 1685, 1694, 1704, 1705, 2010, 2012
\RequirePackageWithOptions 4
resetalternatfont 25, 62
reverse commands:
 \reverse_if:N 589
\rmdefault 1599
\rmfamily 537, 538, 1217, 1218

S

\S 106
scan commands:
 \scan_stop: ... 62, 194, 819, 1454, 1455, 1927, 1935, 3776
scheme 10, 39
scriptfont@name commands:
 \scriptfont@name: 905
\scriptsize 2869, 2908
\secdef 2179
section/aftername 18
section/afterskip 20
section/beforeskip 19
section/format 17
section/indent 20
section/name 15
section/nameformat 17
section/number 16
section/numberformat 18
section/titleformat 18
\selectfont .. 1045, 1480, 1493, 1803, 1808, 1824, 1891, 2745
seq commands:
 \seq_clear:N 964
 \seq_concat:NNN 1298
 \seq_gput_right:Nn 1071, 2767
 \seq_gset_from_clist:Nn 2075, 3756
 \seq_if_empty:Nf 1310
 \seq_if_in:NnF 1069
 \seq_map_inline:Nn 1318, 2152, 2490, 3779
 \seq_new:N 1076, 1303, 1304, 2074, 2756, 3755
 \seq_put_right:No 1566
 \seq_set_filter:NNn 1296
 \seq_set_split:NnV 1295
 \seq_use:Nnnn 2753

\setCJKfamilyfont 56, 1090, 1164,
3152, 3157, 3164, 3165, 3166, 3167, 3168, 3169, 3194,
3195, 3203, 3204, 3206, 3207, 3208, 3209, 3211, 3232,
3234, 3236, 3238, 3265, 3266, 3267, 3268, 3291, 3296,
3301, 3303, 3329, 3331, 3333, 3334, 3344, 3345, 3346,
3347, 3359, 3360, 3361, 3362, 3363, 3364, 3389, 3390,
3391, 3392, 3393, 3394, 3406, 3407, 3408, 3429, 3430, 3431
\setCJKmainfont .. 57, 1086, 1185, 1190, 1206, 3142, 3190,
3198, 3224, 3258, 3279, 3324, 3341, 3355, 3384, 3402, 3426
\setCJKmathfont 57, 1201, 1209
\setCJKmonofont 57, 1088, 1196, 1208, 3163, 3193,
3202, 3231, 3264, 3290, 3328, 3343, 3358, 3388, 3405, 3428
\setCJKromanfont 1190, 1210
\setCJKsansfont
..... 57, 1087, 1191, 1207, 3145, 3151, 3156, 3192,
3200, 3230, 3263, 3285, 3327, 3342, 3357, 3386, 3404, 3427
\SetSymbolFont 947, 1268, 1272
\sfdefault 1600
\sffamily 539, 540, 1219, 1220
six commands:
 \c_six 2764
skip commands:
 \skip_horizontal:N 1766, 1770
 \skip_if_eq:nnT 1570
 \skip_if_eq:nnTF 1783
 \skip_new:N 1577, 1775
 \skip_set:Nn ... 1573, 1578, 1759, 1928, 1939, 1984, 1992
 \skip_zero:N 1948
\small 2847, 2886
\songti 3435
\space 2253, 2287, 2515, 2555, 2672
space 11, 27, 38, 71
\special 1692, 3526, 3527, 3528, 3529, 3530, 3531,
3532, 3534, 3535, 3536, 3537, 3538, 3539, 3540, 3542,
3543, 3544, 3545, 3546, 3547, 3549, 3550, 3551, 3552,
3553, 3554, 3565, 3566, 3567, 3568, 3569, 3571, 3572,
3573, 3574, 3575, 3577, 3578, 3579, 3580, 3582, 3583,
3584, 3585, 3596, 3597, 3598, 3599, 3600, 3602, 3603,
3604, 3605, 3606, 3608, 3609, 3610, 3611, 3613, 3614,
3615, 3616, 3655, 3656, 3657, 3658, 3659, 3660, 3661,
3663, 3664, 3665, 3666, 3667, 3668, 3669, 3671, 3672,
3673, 3674, 3675, 3676, 3678, 3679, 3680, 3681, 3682,
3683, 3718, 3719, 3720, 3721, 3722, 3724, 3725, 3726,
3727, 3728, 3730, 3731, 3732, 3733, 3735, 3736, 3737, 3738
\SplitArgument 1415, 1520, 2142
stop commands:
 \q_stop ... 137, 142, 151, 194, 202, 1319, 1323, 1550, 1555
str commands:
 \str_case:nnTF 128
 \str_case:onF 1597
 \str_case:onTF 142
 \str_case_x:nnF 1084, 1095
 \str_if_eq:nnF 3025
 \str_if_eq:nnTF 926
 \str_if_eq:onF 590, 3032
 \str_if_eq:onTF 518, 1681, 1689, 2728, 3008

\str_if_eq_x:nnF	586
\str_if_eq_x:nnTF	134, 1232, 3005
\string	897, 899, 901, 910, 918
\strutbox	2922
sub3section	10, 39
sub4section	10, 39
subparagraph/aftername	18
subparagraph/afterskip	20
subparagraph/beforeskip	19
subparagraph/format	17
subparagraph/indent	20
subparagraph/name	15
subparagraph/nameformat	17
subparagraph/number	16
subparagraph/numberformat	18
subparagraph/titleformat	18
subsection/aftername	18
subsection/afterskip	20
subsection/beforeskip	19
subsection/format	17
subsection/indent	20
subsection/name	15
subsection/nameformat	17
subsection/number	16
subsection/numberformat	18
subsection/titleformat	18
subsubsection/aftername	18
subsubsection/afterskip	20
subsubsection/beforeskip	19
subsubsection/format	17
subsubsection/indent	20
subsubsection/name	15
subsubsection/nameformat	17
subsubsection/number	16
subsubsection/numberformat	18
subsubsection/titleformat	18
T	
\T	198
\t	105
\tablename	2048
tablename	13
\TeX	2926
T _E X and L ^A T _E X ₂ _ε commands:	
\@@italiccorr	921
\@par	2332, 2334, 2367, 2369
\@afterheading	2203, 2237, 2271
\@afterindentfalse	2174, 2278, 2302
\@afterindenttrue	2176, 2280, 2304
\@Alph	2668, 2673
\@chapapp	2253, 2287, 2458, 2473, 2474
\@chapter	80, 2249
\@currentlabel	2481, 2485
\@currext	481
\@currname	67, 481
\@defaultunits	34
\@empty	85, 2322
\@endpart	2225, 2246
\@EverySelectfont@Init	1805
\@hangfrom	2331, 2366
\@ifclassloaded	2688
\@ifpackagelater	17, 19, 675, 688, 691
\@ifpackageloaded	50, 494, 684, 1668, 2439, 2498
\@ifundefined	2317, 2340, 2354
\@listI	2845, 2884
\@listi	2845, 2852, 2863, 2884, 2891, 2902
\@M	2190, 2216, 2229, 2241, 2291, 2309, 2332, 2333, 2367, 2368
\@makechapterhead	81, 2268, 2270, 2273
\@makeschapterhead	81, 2297
\@minus	2578, 2582, 2592, 2595, 2605, 2608, 2628, 2631, 2639, 2648, 2651, 2659, 2841, 2843, 2849, 2851, 2853, 2854, 2860, 2862, 2864, 2865, 2880, 2882, 2888, 2890, 2892, 2893, 2899, 2901, 2903, 2904
\@nil	872
\@notprerr	31, 31, 42
\@onlypreamble	31, 31, 31, 42, 531, 1206, 1207, 1208, 1209, 1210, 1211, 2992
\@part	79, 2179, 2182, 2206
\@pkgextension	483, 484, 485, 496, 686
\@plus	2575, 2578, 2582, 2590, 2592, 2595, 2603, 2605, 2608, 2628, 2631, 2633, 2639, 2648, 2651, 2653, 2659, 2841, 2842, 2843, 2849, 2850, 2851, 2853, 2854, 2860, 2861, 2862, 2864, 2865, 2880, 2881, 2882, 2888, 2889, 2890, 2892, 2893, 2899, 2900, 2901, 2903, 2904
\@popfilename	31, 66
\@secCNTformat	82, 2316, 2325
\@sect	82, 2320
\@setfontsize	2831
\@spart	80, 2179, 2228, 2240
\@sssect	83, 83, 2361
\@startsection	83, 83, 86, 2387
\@svsec	2322, 2325, 2331, 2348, 2349
\@svsechd	2346, 2372, 2373
\@tempdima	2425, 2427
\@tempskipa	2172, 2173, 2174, 2178, 2276, 2277, 2278, 2282, 2300, 2301, 2302, 2306, 2327, 2328, 2362, 2363
\@topnewpage	2268
\@xsect	2360, 2375
\abstractname	13
\addCJKfontfeature	62
\AfterEndEnvironment	95
\appendixname	13, 21, 21, 21
\AtBeginDocument	43
\AtEndEnvironment	95
\AtEndOfClass	35
\AtEndOfPackage	35
\AtEndOfPackageFile*	31
\baselineskip	11, 12, 65, 73, 73
\bfseries	45
\bibname	13
\c@chapter	2673

\c@secnumdepth	2183, 2193, 2208, 2219, 2250, 2285, 2321, 2338, 2352
\c@section	2668
\caption	28
\catcode	31, 31, 42, 42, 105
\ccwd ..	12, 12, 12, 23, 23, 23, 28, 34, 69, 70, 71, 73, 73, 73, 74
\chapter	10, 15, 20, 21, 21, 80
\chaptermark	14
\chaptername	16
\char	47
\CheckCommand	71
\Chinese	28, 28
\chinese	23, 23, 28, 28
\CJK@ignorespaces	44, 603
\CJK@encoding	52, 52, 53, 61, 778, 782, 866, 933, 934, 936, 938, 939, 940, 942, 946, 948, 1263, 1266, 1268, 1272, 1412, 1427, 1428, 1445, 1446
\CJK@envStart	43, 43, 43, 597
\CJK@family	48, 48, 773, 775, 778, 783, 830, 866, 1042, 1479, 1489, 1491, 1824, 1825
\CJK@ignorespaces	44, 44, 608, 612
\CJK@input	42, 43, 562
\CJK@loadBinding	42, 43, 593
\CJK@loadEncoding	42
\CJK@makeActive	43, 592
\CJK@plane	42, 71, 1812
\CJK@surr	42, 42, 42, 563, 582
\CJK@upperReset	43
\CJKfamily	53
\CJKfamilydefault	57, 65, 65, 65, 66, 66
\CJKglue	73, 73, 94
\CJKhook	44, 44
\CJKrmdefault	42
\CJKsymbol	71
\CJKunderdotbasesep	28
\CJKunderline	28
\clearpage	44
\contentsname	13, 13
\CS	49, 49
\csname	94, 95
\CTEX@appendixnumber	2405, 2413, 2418
\CTEX@chapter@afterskip	2295, 2313
\CTEX@chapter@beforeskip	2276, 2300
\CTEX@chapter@format	2284, 2308
\CTEX@chapter@indent	2284, 2308
\CTEX@chapter@pagestyle	2163, 2274, 2298
\CTEX@chapter@titleformat	2293, 2311
\CTEX@chaptername	2288
\CTEX@part@afterskip	2202, 2236
\CTEX@part@beforeskip	2172
\CTEX@part@format	2192, 2218, 2231, 2243
\CTEX@part@indent	2192, 2231
\CTEX@part@pagestyle	2162, 2207
\CTEX@part@titleformat	2198, 2224, 2233, 2245
\CTEX@partname	2195, 2221
\CTEX@postappendix	2399, 2414, 2419
\CTEX@postchapter	2419
\CTEX@postsection	2414
\CTEX@preappendix	2398, 2412, 2417
\CTEX@prechapter	2417
\CTEX@presection	2412
\CTEX@save@appendix	2407, 2410
\CTEX@save@refstepcounter	2501, 2504
\CTEX@selectfont@hook	71, 1795, 1796, 1798, 1803, 1812, 1815
\CTEX@set@titleformat@n	83, 2377, 2393
\CTEX@setcurrentlabel@n	2479, 2505
\CTEX@thechapter	2418
\CTEX@thesection	2413
\CTEX@titleformat@n ..	83, 83, 83, 2369, 2373, 2378, 2379
\CTEX@toc@width@n	2422, 2437, 2444
\CTEX@todayold	2020, 2025
\CTEX@verbatim@font@hook	1581, 1584
\ctex@zhmap@endinput	3743
\CTEXdigits	23, 23
\CTEXnumber	23, 23, 23, 23, 23
\CTEXoptions	28, 28
\ctexset	6, 6, 6, 6, 6, 6, 6, 8, 8, 10, 11, 12, 12, 12, 12, 13, 13, 13, 13, 13, 13, 13, 13, 15, 15, 16, 16, 17, 17, 18, 18, 18, 19, 20, 20, 20, 21, 21, 24, 25, 27, 28, 28, 96
\CTEXsetup	28
\CTEXthechapter	14, 14, 14, 16
\CTEXthesection	14, 14, 16, 85
\CTEXunderdotbasesep	28
\CTEXunderline	28
\curr@fontshape	784, 814, 823, 827, 873
\DeclareAlternateKanjiFont	61
\DeclareFontFamily	49
\DeclareFontShape	49
\DeclareFontShape@	61, 1371
\DeclareRobustCommand	32
\define@newfont	48, 806
\do@subst@correction	48, 49, 805
\end	95, 95
\EverySelectfont	71
\external@font	816, 817, 819
\extract@font	48, 804
\f@baselineskip	1927
\f@encoding	782, 822, 917
\f@family	48, 783, 822, 830
\f@series	778, 866
\f@shape	778, 866
\f@size	778, 784, 861, 869, 872, 873, 1755, 1791, 2428
\false	9
\fangsong	9, 9
\figurename	13
\font	49, 49
\font@name	48, 49, 787, 794, 795, 798, 801, 819, 820, 837, 897, 898, 899, 900, 901
\fontcharwd	70, 70
\footnotesep	93, 94, 94
\footnotesize	94

\get@external@font	813	\normalsize	9, 9, 90
\getanddefine@fonts	895, 909	\nouppercase	42
\globaljfont	49	\p@	1755, 1791, 2220, 2265, 2266, 2275, 2287, 2294, 2299, 2312, 2428, 2538, 2553, 2558, 2560, 2567, 2841, 2842, 2843, 2849, 2850, 2851, 2853, 2854, 2860, 2861, 2862, 2864, 2865, 2880, 2881, 2882, 2888, 2889, 2890, 2892, 2893, 2899, 2900, 2901, 2903, 2904
\heiti	9	\pagestyle	10
\Hy@driver	1689	\paragraph	10, 10, 10, 10, 10, 19, 20, 34, 34
\Hy@unicodetrue	68	\parindent	12, 23, 73, 75, 94
\HyPsd@ConvertToUnicode	68	\part	15, 20, 80
\HyPsd@LoadUnicode	68	\partname	16
\HyPsd@pdfencoding	68, 68	\PassOptionsToClass	40
\HyPsd@ToBigChars	68	\PassOptionsToPackage	67
\if@mainmatter	2251, 2286, 2473, 2695, 2696	\patchcmd	32
\if@noskipsec	2169	\pdfmapline	100, 102
\if@twocolumn	2267	\pdfstringdef	68
\if@twoside	2452, 2459	\pickup@font	48
\ifin@	911, 919	\pickup@jfont	810, 898, 900, 902
\ifpdf	30	\ProcessKeysOptions	26
\indexname	13	\protected	47
\itdefault	49	\protected@edef	2017, 2019, 2325, 2481
\itshape	45	\ProvidesFile	105
\kaishu	9	\ps@fancy	85, 2466, 2468, 2472, 2475
\labelformat	85, 85	\ps@headings	84, 2451, 2453, 2457, 2460
\leftmark	14	\refname	13
\linewidth	73, 73	\refstepcounter	85
\lishu	9, 9	\relax	42, 73
\listfigurename	13	\reset@font	544, 1224
\listtablename	13	\rightmark	14
\ltj@@al@getanddefine@fonts	895, 912	\rmfamily	46
\ltj@@does@alt@set	50	\roman	23
\ltj@@getjfontnumber	51, 882	\scantokens	32
\ltj@@IsFontJapanese	52, 890, 894, 910	\scriptfont@name	900
\ltj@@ja@getanddefine@fonts	896, 911	\section	10, 10, 15, 15, 21, 21
\ltj@@mathJapaneseFonts	894, 918	\sectionmark	14, 14
\ltj@@set@stackfont	904, 905, 906	\selectfont	50, 65, 71, 71, 71, 73
\ltj@allalchar	729, 747	\setCJKfamilyfont	24, 24, 24
\ltj@curjfnt	49, 71	\setCJKmainfont	24
\ltj@pickup@altfont@auxy	50, 867	\sf@size	899
\ltj@pickup@altfont@copy	51, 880	\ssfamilly	46
\ltj@setpar@global	903	\size@update	73, 73, 1891, 1905
\ltj@temp	911, 912, 913	\sldefault	49
\ltj@tempcnta	910, 918	\songti	8
\ltj@tempcntc	51, 886	\special	8, 67
\ltjalchar	47	\split@name	872
\ltjsetparameter	65	\SplitArgument	78
\m@ne	2183, 2193, 2250, 2285, 3052, 3053, 3054, 3055, 3056, 3057	\ssf@size	901
\markboth	14	\strutbox	94
\markright	14	\subparagraph	10, 10, 10, 10, 19, 20, 34
\math@bgroup	916	\subst@correction	833, 840
\math@egroup	920	\tablename	13
\math@fonts	903	\textfont@name	898
\mathbb	94	\tf@size	897
\maxdimen	12, 73	\thechapter	14, 16, 21
\meaning	32	\thesection	14, 16, 16, 21, 85
\newcommand	32		
\newrobustcmd	32		
\nfss@catcodes	52		

<code>\thispagestyle</code>	80	<code>\tl_if_blank:nTF</code> 1328, 1424, 1425, 1529, 1530, 1544, 1562	
<code>\today</code>	12, 12	<code>\tl_if_blank_p:n</code>	1297
<code>\ttfamily</code>	46	<code>\tl_if_empty:cT</code>	2495
<code>\uppercase</code>	42, 42	<code>\tl_if_empty:Nf</code>	773, 1954
<code>\use@mathgroup</code>	915	<code>\tl_if_empty:NTF</code>	2095, 2973
<code>\usepackage</code>	28, 28	<code>\tl_if_eq:NNT</code>	1591
<code>\verbatim@font</code>	1581, 1582	<code>\tl_if_eq:NNTF</code>	1894
<code>\wrong@fontshape</code>	48	<code>\tl_if_exist:Nf</code>	
<code>\x@protect</code>	131	56, 532, 534, 535, 536, 1212, 1214, 1215, 1216, 2691
<code>\XeTeXglyphbounds</code>	110	<code>\tl_if_exist:NTF</code>	90, 2061, 2064
<code>\yahei</code>	9, 9	<code>\tl_map_function:nN</code>	106
<code>\youyuan</code>	9, 9	<code>\tl_map_inline:Nn</code>	1440
<code>\z@</code>	716, 719, 2173, 2191, 2230, 2277, 2283,	<code>\tl_map_inline:nn</code>	2797
	2301, 2307, 2328, 2363, 2842, 2850, 2861, 2881, 2889, 2900	<code>\tl_new:c</code>	2083, 2084
<code>\zhdigits</code>	23	<code>\tl_new:N</code>	24, 56,
<code>\zhnumber</code>	23, 23, 23, 23		204, 209, 534, 535, 536, 775, 776, 864, 930, 981, 1007,
<code>\zihao</code>	21, 21		1027, 1028, 1049, 1376, 1517, 1612, 1902, 1915, 2398, 2399
<code>\ziju</code>	23	<code>\tl_put_left:Nn</code>	66, 1805
<code>\zw</code>	65, 65, 70	<code>\tl_put_right:Nn</code> 431, 1557, 2124, 2126, 2128, 2130, 2132	
tex commands:		<code>\tl_put_right:Nx</code>	1524, 2109, 2800, 2801
<code>\tex_divide:D</code>	73, 1936, 1977	<code>\tl_replace_all:Nnn</code>	1294, 1332
<code>\tex_endlinechar:D</code>	553, 559	<code>\tl_set:cn</code>	2146, 2149
<code>\tex_font:D</code>	3776	<code>\tl_set:Nn</code>	188,
<code>\tex_iftrue:D</code>	2696		189, 261, 297, 298, 524, 525, 526, 764, 766, 777, 865,
<code>\tex_ignorespaces:D</code> .	1039, 1177, 1182, 1838, 1961, 2741		927, 928, 965, 1032, 1044, 1293, 1329, 1331, 1372,
<code>\tex_lowercase:D</code>	107, 199, 739, 1286		1545, 1549, 1824, 1867, 1878, 1916, 2930, 3183, 3184, 3185
<code>\tex_uppercase:D</code>	1654	<code>\tl_set:Nx</code> ...	205, 566, 816, 1168, 1614, 1852, 1863, 1964
textfont@name commands:		<code>\tl_set_eq:Nc</code>	481
<code>\textfont@name:</code>	904	<code>\tl_set_eq:NN</code>	
<code>\the</code>	886, 3451, 3454, 3455, 3474, 3478	782, 783, 830, 938, 1033, 1058, 1479, 1491, 1903
<code>\thechapter</code>	2253, 2256, 2287, 2458, 2473, 2474, 2556	<code>\tl_trim_spaces:n</code>	1508
<code>\theparagraph</code>	2613	<code>\today</code>	2020, 2025, 2028, 2033
<code>\thepart</code>	2185, 2194, 2210, 2220, 2516	today	12, 76
<code>\thesection</code>	2451, 2460, 2467, 2476, 2573	token commands:	
<code>\thesubparagraph</code>	2619	<code>\token_get_replacement_spec:N</code>	117, 137
<code>\thesubsection</code>	2453, 2469, 2587	<code>\token_to_str:N</code>	178, 1086, 1087,
<code>\thesubsubsection</code>	2600		1088, 1090, 1097, 1098, 1099, 1161, 2058, 2059, 2749, 3783
<code>\thispagestyle</code>	2207, 2274, 2298	<code>\toks</code>	3451, 3452, 3454, 3455, 3474, 3478
three commands:		<code>\topsep</code>	2853, 2864, 2892, 2903
<code>\c_three</code>	399, 1981, 2643, 3793	<code>\TrimSpaces</code>	500
<code>\tiny</code>	2870, 2909	<code>\ttdefault</code>	1601
tl commands:		<code>\ttfamily</code>	541, 542, 1221, 1222
<code>\tl_clear:c</code>	1715, 2148	two commands:	
<code>\tl_clear:N</code>		<code>\c_two</code>	207, 236, 435, 439, 1982, 2428, 2623, 2663
..	266, 424, 447, 1511, 1515, 1825, 1882, 2151, 2158, 2796	<code>\c_two_hundred_fifty_six</code>	569, 572
<code>\tl_const:cn</code>	663, 686	<code>\typeout</code>	2253, 2254
<code>\tl_const:cx</code>	2085, 2091		
<code>\tl_const:Nn</code>	466, 471, 476, 533, 933,		
	944, 1213, 1214, 1215, 1216, 1656, 1665, 2689, 2697, 2699		
<code>\tl_const:Nx</code>	1262, 1901		
<code>\tl_gput_right:Nn</code>	57		
<code>\tl_gset:Nn</code>	1236, 1652, 1662, 1663		
<code>\tl_gset:Nx</code>	570, 573, 1595, 1624, 3010		
<code>\tl_gset_eq:NN</code>	1021, 1619		
<code>\tl_gset_rescan:Nnn</code>	1243		
<code>\tl_head:N</code>	2800		

U

```
um commands:
    \um_cs_set_eq_active_char:Nw . . . . . 756, 757
\undefined . . . . . 3476
\updefault . . . 935, 940, 942, 946, 948, 1264, 1266, 1269, 1273
use commands:
    \use:c . . . . . 784, 822, 823, 873, 949, 999, 1492
    \use:n . 210, 1233, 1237, 1613, 1733, 1800, 2379, 2433, 2949
```

`\use:x` 70,
 115, 446, 761, 872, 980, 1116, 1334, 1508, 2157, 2759, 2982
`\use_i:nn` 36
`\use_i:nnn` 337
`\use_ii:nn` 38
`\use_ii:nnn` 211, 342
`\use_iii:nnn` 347
`\use_none:n` .. 262, 267, 272, 1016, 1236, 1369, 1734, 2015
 UTF8 7, 37

V

`\verse` 94, 2942, 2943
`\vskip` 2201, 2202,
 2220, 2235, 2236, 2287, 2294, 2295, 2312, 2313, 2538, 2558
`\vspace` 2275, 2282, 2299, 2306

W

`winfonts` 27
`\write` 3485

X

`\x` 3453, 3456,
 3457, 3458, 3459, 3460, 3461, 3462, 3463, 3464, 3465,
 3466, 3467, 3468, 3469, 3470, 3471, 3472, 3479, 3483, 3487
`\xdef` 897, 899, 901, 3486
`\xeCJKsetup` 669, 1770, 1833, 1836, 1839, 1854

xetex commands:

`\xetex_if_engine:F` 3749
`\xetex_if_engine:TF` 1868, 3387
`\XeTeXcharglyph` 3780
`\XeTeXglyphbounds` 3804

Y

`\yahei` 3443
`\youyuan` 3441

Z

zero commands:

`\c_zero` 234, 734, 738, 1937, 2733
`\c_zero_dim` 1929, 1941, 1956, 1972, 1979, 2135, 2664
`\c_zero_fp` 1901, 2734, 2918
`\c_zero_skip` 2133, 2134
`\zhdigits` 23, 2019
`zhmap` 8, 27, 38
`\zhnum` 2013
`\zhnumber` 23, 2017
`\zhnumsetup` 2029, 2034
`\zhtoday` 2028, 2033
`\zihao` 21, 90, 2740, 2749
`zihao` 9, 36
`\ziju` 23, 74, 1960
`\zw` 717, 1573, 1578, 1759, 1792