

CT_EX 宏集手册

ctex.org

2015/06/19 v2.1*

简介

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

CT_EX 宏集支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX 和 LuaL^AT_EX 等多种不同的编译方式,并为它们提供了统一的界面。主要功能由宏包 ctex 和中文文档类 ctexart、ctexrep、ctexbook 实现。

目录

第 1 节 介绍	2	第 9 节 实用命令	21
第 2 节 简明教程	3	9.1 字号与间距	21
2.1 CT _E X 宏集的组成	3	9.2 中文数字转换	23
2.2 使用 CT _E X 文档类	3	9.3 杂项	23
2.3 使用 ctex 宏包	3	第 10 节 LuaL ^A T _E X 下的中文支持方式	23
第 3 节 依赖与安装	4	10.1 LuaL ^A T _E X 下替代字体的设置	24
第 4 节 宏包选项与 \ctexset 命令	5	第 11 节 CT _E X 宏集的配置文	25
第 5 节 编译方式、编码与中文字库	6	11.1 修改宏包默认选项	25
5.1 编译方式	6	11.2 宏包载入后的配置	26
5.2 中文编码	6	11.3 配置标题中文翻译	26
5.3 中文字库	6	11.4 自定义字体集	26
第 6 节 排版格式设定	8	第 12 节 对旧版本的兼容性	26
6.1 文档默认字号	8	12.1 CT _E X 0.8a 及以前的版本	26
6.2 章节标题风格	9	12.2 CT _E X 0.9–CT _E X 1.0d	27
6.3 排版方案选项	9	12.3 CT _E X 1.02c 以后的 SVN 开发版	28
第 7 节 文档汉化	11	第 13 节 开发人员	29
7.1 日期汉化	11	第 14 节 参考文献	29
7.2 文档标题汉化	12	第 15 节 代码实现	30
7.3 页面格式设置与汉化	12	第 16 节 版本历史	115
第 8 节 章节标题格式设置	14	第 17 节 代码索引	115

*ctex-kit rev. ca75f0e.

第 1 节 介绍

历史

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

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

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

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

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

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

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

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

关于宏集名字的说明

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

CT_EX 宏集是由 [CT_EX 社区](#) 发起并维护的 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 宏集并非是一事物。

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

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 C_TE_X 宏集组成

为了适应用户不同的需求,我们将 C_TE_X 宏集的主要功能分散在三个中文文档类和四个宏包当中,具体的组成见表 1。

表 1 C_TE_X 宏集的组成

类别	文件	说明
文档类	ctexart.cls	标准文档类 article 的汉化版本,一般适用于短篇幅的文章
	ctexrep.cls	标准文档类 report 的汉化版本,一般适用于中篇幅的报告
	ctexbook.cls	标准文档类 book 的汉化版本,一般适用于长篇幅的书籍
宏包	ctex.sty	提供全部功能,但默认不开启章节标题设置功能,需要使用 heading 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 ctex.sty
	ctexsize.sty	定义和调整中文字号,在 ctex 宏包或 C _T E _X 中文文档类之外单独调用
	ctexheading.sty	提供章节标题设置功能的(见 8 节),在 ctex 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 宏集提供的三个中文文档类。

C_TE_X 宏集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}
```

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

```

\begin{itemize}
  \item 你需要将所有源文件保存为 UTF-8 编码
  \item 你可以使用 XeLaTeX 或 LuaLaTeX 编译
  \item 也可以使用 (pdf)LaTeX 编译
  \item 推荐使用 XeLaTeX 或 LuaLaTeX 编译
\end{itemize}
\end{frame}
\end{document}

```

有些文档类是建立在 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 宏集之前，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 宏集，它们还依赖 – etex 宏包。
- indentfirst 宏包，属于 tools 宏集。
- etoolbox 宏包。
- everyisel 宏包，属于 ms 宏集。
- zhnumber 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- CJK 宏集，它的下划线功能依赖 ulem 宏包。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包。
- ➡ 以上是使用 pdfL^AT_EX 或 L^AT_EX + DVIPDFMx 的编译方式所需要的依赖项，其中 zhm-CJK 是可选的。
- xeCJK 宏包，它还依赖
 - fontspec 宏包，它还依赖
 - * euenc 宏包。
 - * xunicode 宏包，它还依赖 tipa 宏包。
- ➡ 以上是使用 X_YL^AT_EX 编译时的依赖项。
- LuaTeX-ja 宏集，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 依赖的各个宏包和宏集均已被 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 宏集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 T_EX 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。

CT_EX 的正式版本也已被 T_EX Live 和 MiK_TE_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 CT_EX 的 [TDS 安装包](#) 按目录结构将文件复制到 T_EX 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

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

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

另一方面,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 CT_EX 宏集提供的用户接口 \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 CT_EX 宏集的通用控制命令,用来在宏包载入后控制宏包的各项功能。 \ctexset 的参数是一个键值列表,以通用的接口完成各项设置。

\ctexset 的参数是一组由逗号分隔的选项列表,列表中的选项通常是一个 $\langle key \rangle = \langle value \rangle$ 格式的定义。例如设置摘要与参考文献标题名称([7.2 节](#))就可以使用:

例 4

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

\ctexset 采用 L^AT_EX3 风格的键值设置,支持不同类型的选项与层次化的选项设置,相关示例见 [8 节](#)。

第 5 节 编译方式、编码与中文字库

5.1 编译方式

CT_EX 宏集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)。

表 2 CT_EX 宏集的中文支持方式

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

不同的编译方式和中文支持方式会在一定程度上影响 CT_EX 宏集的行为，比如宏包对编码的处理。在用户使用 X_YL_AT_EX 及 LuaL_AT_EX 编译时，CT_EX 宏集使用(且仅能使用)UTF-8 编码；而因为历史原因，在用户使用 L_AT_EX 及 pdfL_AT_EX 编译时，宏包默认使用 GBK 编码。用户需要保证编译方式、源文件编码、宏包编码选项三者一致。关于宏包编码选项，可以参考 5.2 节。

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

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 宏集无法检测用户编写文档时使用的编码格式，因此需要用户自行指定编码。我们建议用户总是使用 UTF-8 编码，并显式指定 UTF8 选项，并使用 X_YL_AT_EX 或 LuaL_AT_EX 编译。

使用 X_YL_AT_EX 或 LuaL_AT_EX 编译时，CT_EX 宏集强制使用 UTF-8 编码，此时 GBK 选项无效。使用 (pdf)L_AT_EX 编译时，CT_EX 宏集默认启用 GBK 选项；不过，用户也可以显式声明 UTF8 选项，使 CT_EX 宏集工作在 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 宏集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。

通常，由 CT_EX 宏集进行的自动配置已经足够使用，无需用户手工干预；但是 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。

zhmap

zhmap = <true|false|zhmCJK>

该选项只能在调用 CT_EX 宏包/文档类时作为选项使用，并且只在使用 pdfL_AT_EX/L_AT_EX 编译时有意义

指定字体映射机制。

true

使用 zhmetrics 宏包，将 CJK 字库通过 \special 命令映射到 .ttf 文件。

false

使用传统的 CJK 字库(Type 1)⁷。

zhmCJK

载入 zhmCJK 宏包⁸，由 zhmCJK 宏包提供从 CJK 字库到 .ttf 的映射。

⁵L_AT_EX、pdfL_AT_EX、X_YL_AT_EX 以及 LuaL_AT_EX。
⁶CT_EX 宏集现在能够识别 Mac OS X 系统以及 Windows 系统。
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。
⁸zhmCJK 宏包基于 zhmetrics 和 CJK 宏包，提供与 xeCJK 宏包类似的用户接口。

表 3 CT_EX 宏集自动配置字体策略

	Mac OS X	Windows New ¹	Windows Old ²	其他
X _E 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。

```
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 CT_EX 宏集加载的字库。

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

CT_EX 预定义了以下六种中文字库。

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

注意:使用 (pdf)L^AT_EX 编译的时候,若设置 zhmap = false(比如需要使用 L^AT_EX + DVIPS 编译),则需要按照传统方式⁹在本地安装好 CJK 字体。

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

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

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

例 5

```
\documentclass[fontset = none]{ctexart}
```

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

```
\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}
```

CT_EX 宏集预定义的中文字库还定义了一些字体命令。

- \songti 宋体,CJK 等价命令 \CJKfamily{zhsong}。
- \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 zihao = {<-4|5|false}
该选项只能在调用 CT_EX 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, 具体情况见表 4。false 禁用本功能。本选项可以用于三个 CT_EX 文档类和 ctex 宏包, 也可以用于 ctexsize 宏包。

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

表 4 标准字体命令与字号的对应

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

10pt 该选项只能在调用 C_TE_X 宏包/文档类时作为选项使用
 11pt
 12pt

C_TE_X 文档类是在 L_AT_EX 标准文档类之上开发的。因此,除了可以使用 C_TE_X 宏包定义的字号选项之外,还可以使用标准文档类的同类选项(10pt、11pt 和 12pt)。在使用这些来自标准文档类的选项的时候,C_TE_X 文档类的字号选项会被抑制。亦即,在 zihao 选项之后设置 10pt 选项,zihao 选项将不再起作用。

标准文档类的其他选项在 C_TE_X 文档类中依旧有效。例如,设置纸张大小和方向的 a4paper 和 landscape,设置单双面的 oneside 和 twoside 等。C_TE_X 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

C_TE_X 宏集提供了一套用于修改文档章节标题格式的接口。该选项用于选择是否启用该功能。详细的设置方法请参见 8 节和 7.3 节。

C_TE_X 宏集提供的三个文档类总是启用该功能。如果在 ctex.sty 下启用该选项,将会检查当前是否使用 L_AT_EX 标准文档类。若然,则该选项将会使得 ctex.sty 宏包的行为和 C_TE_X 宏集提供的三个中文文档类完全一致;若不然,则会根据 \chapter 是否有定义来使用 ctexbook 或者 ctexart 的标题设置。

sub3section 该选项只能在调用 C_TE_X 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 C_TE_X 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 C_TE_X 宏包/文档类时作为选项使用

选择文章的排版方案,预设有 chinese 和 plain 两种方案。

chinese 调整默认字号为五号字,并调整行距为 1.3;汉化文档中的标题名字(如“图”、“表”、“目录”和“参考文献”,见 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)。当 ctex 宏包与标准文档类及其衍生文档类联用时,会载入 indentfirst 宏包,以实现章节标题后的段首缩进。

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。

¹¹使用 C_TE_X 文档类,或者使用 ctex 宏包且开启该选项时。

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

`punct` `punct = <quanjiao|banjiao|kaiming|CCT|plain>`
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;
`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 $(\text{pdf})\text{\LaTeX}$ 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。本选项的初始值与 `scheme` 有关。

`scheme = chinese` 初始值为 1.3, 即 1.3 倍行距。此时, 相邻两行的基线(`\baselineskip`)距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。

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

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。

¹³ \LaTeX 将单个换行视作一个空格。

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。

<数值或带单位的数值>

用于设置段首缩进的长度。如果不带单位,则默认单位是单个汉字字宽 `\ccwd`; 如果带单位,则使用该单位。`true`等价于设置 `autoindent = 2`。`false`

禁用自动调整功能,可以设置固定长度的段首缩进。如设置每段缩进 40 点:

例 8

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

linestretch

`linestretch = {数值或长度}`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

`CTeX` 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 6 月 19 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 6 月 19 日”。使用阿拉伯数字和汉字的日期格式。

`big`

效果为“二〇一五年六月十九日”。使用全汉字的日期格式。

`old`

效果为“June 19, 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}
```

`CTeX` 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。

<code>contentsname</code>	<code>contentsname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
---------------------------	--

设置目录标题名 `\contentsname`。中文默认为“目录”。

<code>listfigurename</code>	<code>listfigurename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
-----------------------------	--

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

<code>listtablename</code>	<code>listtablename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
----------------------------	---

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

<code>figurename</code>	<code>figurename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
-------------------------	--

设置图片环境标题名 `\figurename`。中文默认为“图”。

<code>tablename</code>	<code>tablename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
------------------------	---

设置表格环境标题名 `\tablename`。中文默认为“表”。

<code>abstractname</code>	<code>abstractname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
---------------------------	--

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

<code>indexname</code>	<code>indexname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
------------------------	---

设置索引标题名 `\indexname`。中文默认为“索引”。

<code>appendixname</code>	<code>appendixname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
---------------------------	--

设置附录标题名 `\appendixname`。中文默认为“附录”。

<code>bibname</code>	<code>bibname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置
----------------------	---

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

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,CT_EX 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。

CT_EX 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。CT_EX 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 1 节——天地玄黄

— 1 —

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。

第 7.3 节和本节介绍的功能已经被提取到 ctexheading 宏包之中,可以在 ctex 宏包和 ctexart 等文档类之外独立使用。各选项的默认值与 scheme = plain 时的情形相同。

章节标题的格式选项是分层设置的。顶层的选项是章节标题名称,次一级的选项是章节标题的格式。章节标题名包括 part, chapter, section, subsection, subsubsection, paragraph, subparagraph;而可用的格式包括 name, number, format, nameformat, numberformat, aftername, titleformat, before skip, after skip, 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。注意斜线前后都不要有空格。

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	scheme = plain	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 L^AT_EX 辅助文件中的是 number 选项的定义。

但是, number 选项不会影响计数器本身的输出。即设置 section/number 不会影响 \thesection 的定义。(但该选项会影响 \CTEXthesection 的定义, 见后。)

表 6 number 选项的默认设置

标题名	scheme = chinese	scheme = plain	原 \the(标题) 等价定义
part	\chinese{part}	\thepart	\Roman{part}
chapter	\chinese{chapter}	\thechapter	\arabic{chapter}
section	同右	\thesection	\arabic{section}
subsection	同右	\thesubsection	\thesection.\arabic{subsection}
subsubsection	同右	\thesubsubsection	\thesubsection.\arabic{subsubsection}
paragraph	同右	\theparagraph	\thesubsubsection.\arabic{paragraph}
subparagraph	同右	\thesubparagraph	\theparagraph.\arabic{subparagraph}

\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 C_TE_X 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	scheme = plain
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。

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

```
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。

`numberformat` 选项的最后一个格式命令可以是一个有一个参数的命令。这个命令的参数用于接受编号数字。

例如,我们可以使用 `numberformat` 特别强调章标题中的数字:

表 8 nameformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
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	{}

例 16

```
\ctexset{
  chapter/number = \arabic{chapter},
  chapter/numberformat = \color{blue}\zihao{0}\itshape,
}
```

上面的代码在 scheme = chinese 时可以做出类似这样的章标题效果：

第 4 章

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

aftername = {<代码>}
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。
aftername+ 选项用于在原有代码后面附加代码。

表 9 aftername 选项的默认设置

标题名	scheme = chinese	scheme = plain
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	scheme = plain
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/aftertitle
chapter/aftertitle
section/aftertitle
subsection/aftertitle
subsubsection/aftertitle
paragraph/aftertitle
subparagraph/aftertitle
.../aftertitle+

aftertitle = {<代码>}
aftertitle+= {<代码>}
该选项只能用 \ctexset 来设置

aftertitle 选项的参数 <代码> 将被插入到章节标题内容之后。可用于控制标题内容结束后是否另起一段。

aftertitle+ 选项用于在原有代码后面附加代码。

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

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 aftertitle 选项的默认值。

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

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

beforeskip 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 beforeskip 选项的默认值。

表 11 beforeskip 选项的默认设置

标题名	scheme = chinese	scheme = plain
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 aftertitle 选项的默认设置

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

表 13 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

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。
```

表 14 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

表 15 pagestyle 选项的默认设置

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

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)。

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 16 appendix/name 选项的默认设置

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

appendix/number

number = {{数字输出命令}}
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。
我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

表 17 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}

例 19

```
\ctexset {
  chapter = {
    format      = \Huge\bfseries,
    nameformat  = \hrule height 1bp \relax\bigskip\hfill\chapternamebox,
    number      = \arabic{chapter},
    aftername   = \par\medskip,
    titleformat = {},
    aftertitle  = \bigskip\hrule height 2bp \relax
  }
}
\newcommand\chapternamebox[1]{%
  \parbox{\ccwd}{\linespread{1}\selectfont\centering #1}}
.....
\chapter{熟悉 \LaTeX}
```

本例的设置效果大致如下：

第
1
章

熟悉 \LaTeX

第 9 节 实用命令

9.1 字号与间距

\backslash zihao

\backslash 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 18 所示。使用 \backslash 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。

\backslash ziju

\backslash ziju{<中文字符宽度的倍数>}

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

这个命令会影响 \backslash ccwd 的值，但不会影响英文字距。

\backslash ccwd

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

\backslash CTEXsetfont

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

表 18 中文字号

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

9.2 中文数字转换

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

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

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

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

以中文格式输出数字。这里的数字可以是整数、小数和分数。

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

将阿拉伯数字转换为中文数字串。

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

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

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

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

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

9.3 杂项

$\backslash\text{CTeX}$	用于显示 $\text{CT}_{\text{E}}\text{X}$ 标志。
-------------------------	---

第 10 节 $\text{Lua}\text{\LaTeX}$ 下的中文支持方式

在 $\text{Lua}\text{\LaTeX}$ 下, $\text{CT}_{\text{E}}\text{X}$ 宏集依赖 `LuaTeX-ja` 宏包来完成中文支持。该宏包是日本 $\text{T}_{\text{E}}\text{X}$ 社区的北川弘典、前田一贵、八登崇之等人开发的,设计目的主要是在 $\text{Lua}\text{\LaTeX}$ 引擎下实现日本 $\text{pT}_{\text{E}}\text{X}$ 引擎的(大部分)功能。它为了兼容 $\text{p}\text{\LaTeX}$ 的使用习惯,对 $\text{\LaTeX}_{2_{\text{e}}}$ 的 `NFSS` 作了不少修改和扩充。这对于简体中文用户来说不是必要的,因而 $\text{CT}_{\text{E}}\text{X}$ 禁用了它在 \LaTeX 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-ja` 宏包开始支持竖排,但 $\text{CT}_{\text{E}}\text{X}$ 暂不支持竖排。

10.1 Lua_{TeX} 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 20

```
\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 CT_EX 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

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

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

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

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 21

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

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

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

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 \LaTeX 语法。

例 22

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

例 23

```
% 略复杂的 ctex.cfg 内容示例: 禁止段末孤字成行。
% 在使用 XeTeX 编译时, 打开 xeCJK 的 CheckSingle 选项。
\xetex_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 \CTeX 宏集需要同时支持 GBK 和 UTF-8 两种编码, 因此对标题的中文翻译写在两个配置文件当中: `ctex-name-gbk.cfg` 和 `ctex-name-utf8.cfg`。两个文件的设置相同, 只是编码不同。

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

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 \CTeX 宏集的配置文件, 也应该与其他本地配置文件一起保存在本地 TDS 目录树下, 并可以配合 `ctexopts.cfg` 等配置文件使用。

第 12 节 对旧版本的兼容性

12.1 \CTeX 0.8a 及以前的版本

在 `ctex-kit` 项目成立之前, \CTeX 宏包的最后一个版本是 \CTeX 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。

<hr/> cs4size c5size <hr/>	分别相当于 zihao=-4 和 zihao=5, 过时选项。
<hr/> CCT CCTfont <hr/>	相关选项已删除。
<hr/> indent noindent <hr/>	indent 什么也不做, ctex 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开; C _T E _X 文档类的相关功能由章节 beforeskip 选项的值来确定。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。
<hr/> zhmap nozhmap <hr/>	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。
<hr/> winfonts adobefonts nofonts <hr/>	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)。
<hr/> punct nopunct <hr/>	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。
<hr/> cap nocap <hr/>	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。
<hr/> space nospace <hr/>	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 <code>\usepackage</code> 载入 fancyhdr 宏包。</p> <p><code>fancyhdr</code> 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 <code>\usepackage</code> 载入 hyperref 宏包。</p> <p><code>hyperref</code> 选项过时, 因兼容性保留, 功能是在导言区末尾载入 hyperref 宏包。</p>
<u>fntef</u>	<p>旧版本的 <code>fntef</code> 选项用于统一 CCTfntef 与 CJKfntef 的界面, 新版本 CT_EX 宏集不再支持 CCT, 而是直接载入 CJKfntef 或 xeCJKfntef 宏包并禁用其彩色设置。该选项是过时选项。</p>
<u>\CTEXunderdot \CTEXunderline \CTEXunderdblline \CTEXunderwave \CTEXsout \CTEXxout \CTEXfilltwosides</u>	<p>在调用 <code>fntef</code> 宏包选项的同时, 旧版本 CT_EX 宏包由于需要支持 CCT 系统, 会将以 <code>\CJK</code> 开头的 <code>\CJKunderline</code> 等宏换名为以 <code>\CTEX</code> 开头的 <code>\CTEXunderline</code> 等宏。此功能在新版本的 CT_EX 宏集中已失去意义。</p> <p>此外, 在 pdfT_EX 引擎下, 用于设置格式的 <code>\CJKunderdotbasesep</code> 等宏也被换名为 <code>\CTEXunderdotbasesep</code> 等宏。</p> <p>在新版本中, 上述由 <code>fntef</code> 衍生的相关宏都成为过时命令。</p>
<u>\CTEXindent</u>	更新 <code>\ccwd</code> 宽度后设置 <code>\parindent=2\ccwd</code> 。过时命令。
<u>\CTEXnoindent</u>	设置 <code>\parindent=0pt</code> 。过时命令。
<u>\CTEXsetup</u>	<p><code>\CTEXsetup[⟨选项⟩]{⟨标题⟩}</code></p> <p>相当于设置了 <code>\ctexset{⟨标题⟩ = {⟨选项⟩}}</code>。过时命令。</p>
<u>\CTEXoptions</u>	<p><code>\CTEXoptions[⟨选项⟩]</code></p> <p>相当于设置了 <code>\ctexset{⟨选项⟩}</code>。过时命令。</p>
<u>\Chinese</u>	<p><code>\Chinese{⟨counter⟩}</code></p> <p>新版宏集中 <code>\chinese</code> 统一了旧版本中 <code>\chinese</code> 和 <code>\Chinese</code> 的功能。因此, 该命令已过时。</p>
<u>captiondelimter</u>	<p>原为 <code>\CTEXoptions</code> 命令的选项, 用于控制 <code>\caption</code> 编号后面的标点。此选项已过时, 并在新版本的 CT_EX 宏包中失效。</p> <p>可以使用 caption 宏包的 <code>labelsep</code> 选项来完成同样的功能。</p>

例 24

```
% 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>`

ctexcap 是默认打开 heading 选项的 ctex。

2 `<*ctexcap>`

3 `\PassOptionsToPackage { heading = true } { ctexcap }`

4 `\RequirePackageWithOptions { ctex }`

5 `</ctexcap>`

宏包载入检查。

6 `<*class|ctex>`

7 `\tl_const:Nx \c__ctex_version_tl`

8 `{ \cs_if_exist_use:cF { ver@ \@currname . \@currentx } { 9999/99/99 } }`

9 `<*class>`

10 `\cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl`

11 `\cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl`

12 `\cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl`

13 `\cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl`

14 `</class>`

15 `<*ctex>`

16 `\msg_new:nnnn { ctex } { subpackage-loaded }`

17 `{ Package~\#1'~can~not~be~loaded~with~`ctex'. }`

18 `{`

19 `~\#1'~is~actually~a~part~of~`ctex'.\\`

20 `It~is~not~necessary~to~load~it~separately.`

21 `}`

22 `\@ifpackagelated { ctexsize }`

23 `{ \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }`

24 `{ \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }`

25 `\@ifpackagelated { ctexheading }`

26 `{ \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }`

27 `{ \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }`

28 `</ctex>`

29 `</class|ctex>`

30 `<*class|style>`

31 `\RequirePackage { xparse , l3keys2e }`

32 `</class|style>`

检查 expl3 和 l3keys2e 的版本。

33 `<*class>`

34 `\msg_new:nnnn { ctex } { l3-too-old }`

35 `{ Support~package~\#1'~too~old. }`

36 `{`

37 `Please~update~an~up~to~date~version~of~the~bundles\\`

38 `\l3kernel'~and~\l3packages'\\`

39 `using~your~TeX~package~manager~or~from~CTAN.`

40 `}`

41 `\@ifpackagelater { expl3 } { 2014/07/20 } { }`

42 `{ \msg_error:nnn { ctex } { l3-too-old } { expl3 } }`

43 `\@ifpackagelater { l3keys2e } { 2014/05/05 } { }`

44 `{ \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }`

45 `</class>`

46 `<*class|ctex|ctexheading>`

47 `\RequirePackage { etoolbox }`

48 `<!ctexheading>\RequirePackage { fix-cm , everyrel }`

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`

49 `\tl_new:N \l__ctex_tmp_tl`

50 `\int_new:N \l__ctex_tmp_int`

51 `<!ctexheading>\dim_new:N \l__ctex_tmp_dim`

```
52 \box_new:N \l__ctex_tmp_box
```

(End definition for `\l__ctex_tmp_tl` and others.)

`\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。

```
53 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
54 {
55   \ifpackageloaded {#1}
56     {#2}
57     { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
58 }
```

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

`\ctex_gadd_hook:Nn` 给钩子附加内容。

`\ctex_gadd_hook:cn`

```
59 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
60 {
61   \tl_if_exist:NF #1 { \tl_new:N #1 }
62   \tl_gput_right:Nn #1 {#2}
63 }
64 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }
```

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

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

`\ctex_package_end_hook:o`

```
65 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
66 {
67   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
68   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
69 }
70 \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。

```
71 \tl_put_left:Nn \@popfilename
72 {
73   \cs_if_eq:NNT \@currentx \pkgextension
74   { \ctex_package_end_hook:o { \@currname } }
75 }
```

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

```
76 \cs_new_protected:Npn \ctex_file_wrapper:nnn #1#2#3
77 {
78   \use:x
79   {
80     \ExplSyntaxOff
81     \char_set_catcode_letter:n { 64 }
82     #1
83     \exp_not:n {#2}
84     \bool_if:NTF \l__kernel_expl_bool
85     { \ExplSyntaxOn }
86     { \ExplSyntaxOff }
87     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
88     #3
89   }
90 }
```

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

`\ctex_file_input:n` 输入文件。

```
91 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
92 { \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。

```
93 \cs_new_protected_nopar:Npn \ctex_scheme_input:n #1
94 {
95   \ctex_file_wrapper:nnn
96   { }
97   {
98     \tl_if_exist:NTF \c__ctex_class_tl
99     {
100       \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
101       { \file_input:n { ctex-scheme- #1 .def } }
102     }
103     { \file_input:n { ctex-scheme- #1 .def } }
104   }
105   { }
106 }
107 \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`。

```
108 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
109 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
110 \group_begin:
111 \char_set_lccode:nn { \< } { \{ }
112 \char_set_lccode:nn { \ / } { \ \ }
113 \char_set_lccode:nn { \A } { \t }
114 \tl_map_function:nN { \A \E \S \O \P } \char_set_catcode_other:N
115 \tex_lowercase:D
116 {
117   \group_end:
118   \cs_new_protected:Npn \ctex_parse_name:NNn #1#2#3
119   {
120     \bool_if:nTF { \cs_if_exist_p:c { #3 ~ } || \cs_if_exist_p:c { /#3 } }
121     {
122       \group_begin:
123       \use:x
124       {
125         \group_end:
126         \__ctex_parse_name:nnNNn { \token_get_replacement_spec:N #2 }
127         \exp_not:N #2 \exp_not:c { #3 ~ } \exp_not:c { /#3 }
```

```

128         } {#3} #1
129     }
130     { #1#2 }
131 }
132 \cs_new_protected:Npn \__ctex_parse_name:nnNNnN #1#2#3#4#5#6
133 {
134     \exp_args:Nc #6
135     {
136         \str_case:nnTF {#1}
137         {
138             { \protect #3 } { }
139             { \x@protect #2 \protect #3 } { }
140         }
141         {
142             \str_if_eq_x:nnTF { \exp_not:n { /@protected@ #3 /#3 } }
143             {
144                 \exp_last_unbraced:Nf \__ctex_parse_name:w
145                 \token_get_replacement_spec:N #3 AESAOPA ~ < \q_stop
146             }
147             { /#5 ~ } { #5 ~ }
148         }
149         {
150             \str_case:onTF { \__ctex_parse_name:w #1 AESAOPA ~ < \q_stop }
151             {
152                 { /@protected@ #2 #4 } { }
153                 { /@ #4 } { }
154             }
155             { /#5 } {#5}
156         }
157     }
158 }
159 \cs_new:Npn \__ctex_parse_name:w #1 AESAOPA ~ #2 < #3 \q_stop { #1#2 }
160 }
161 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

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

```

\ctex_patch_cmd:NnnTF 在打补丁前先解析实际名字。
\ctex_preto_cmd:NnTF
\ctex_appto_cmd:NnTF
162 \cs_new_protected_nopar:Npn \ctex_patch_cmd:NnnTF { \ctex_parse_name:NN \patchcmd }
163 \cs_new_protected_nopar:Npn \ctex_preto_cmd:NnTF { \ctex_parse_name:NN \pretocmd }
164 \cs_new_protected_nopar: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 L^AT_EX3 语法。

```

165 \cs_new_protected:Npn \ctex_patch_cmd:Nnn
166 {
167     \group_begin:
168     \char_set_catcode_other:N \#
169     \__ctex_patch_cmd:Nnn
170 }
171 \cs_new_protected:Npn \__ctex_patch_cmd:Nnn #1#2#3
172 {
173     \group_end:
174     \group_begin:
175     \ExplSyntaxOff
176     \ctex_patch_cmd:NnnTF #1 {#2} {#3}
177     {
178         \cs_gset_eq:NN \__ctex_tmp:w #1
179         \group_end:
180         \cs_set_eq:NN #1 \__ctex_tmp:w
181         \cs_undefine:N \__ctex_tmp:w
182     }

```

```

183     { \group_end: \ctex_patch_failure:N #1 }
184   }
185   \cs_new_protected:Npn \ctex_patch_failure:N #1
186   { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
187   \msg_new:nnn { ctex } { patch-failure }
188   {
189     Patching~command~`#1'~failed.\\
190     ctex~may~not~work~as~expected.
191   }

```

(End definition for \ctex_patch_cmd:Nnn.)

\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。

```

192 \cs_new_eq:NN \g__ctex_section_depth_flag \c_two

```

(End definition for \g__ctex_section_depth_flag.)

```

193 </class|ctex|ctexheading>
194 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

195 \msg_new:nnnn { ctex } { package-too-old }
196 { Support~package~`#1'~too~old. }
197 {
198   Please~update~an~up~to~date~version~of~the~package~`#1'\\
199   using~your~TeX~package~manager~or~from~CTAN.
200 }

```

\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。

```

201 \cs_new_eq:NN \ctex_lua_now_x:n \lua_now_x:n

```

(End definition for \ctex_lua_now_x:n.)

\ctex_if_pdfmode:TF ifpdf 宏包的简单实现。

```

202 \cs_if_exist:NTF \pdfTeX_pdfoutput:D
203 {
204   \int_compare:nNnTF \pdfTeX_pdfoutput:D > \c_zero
205   {
206     \cs_new_eq:NN \ifctexpdf \tex_iftrue:D
207     \cs_new_eq:NN \ctex_if_pdfmode:TF \use_i:nn
208   }
209 }
210 { \use:n }
211 {
212   \cs_new_eq:NN \ifctexpdf \tex_iffalse:D
213   \cs_new_eq:NN \ctex_if_pdfmode:TF \use_ii:nn
214 }

```

(End definition for \ctex_if_pdfmode:TF.)

\ctex_if_preamble:TF 测试是否在 L^AT_EX_{2_ε} 的导言区。在宏包内部初始为真, 在 \AfterEndPreamble 钩子中再设置为假。注意, \AfterEndPreamble 在 \AtBeginDocument 之后执行, 可以与 \@onlypreamble 的行为一致。

```

215 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
216 \AfterEndPreamble { \cs_set_eq:NN \ctex_if_preamble:TF \use_ii:nn }

```

(End definition for \ctex_if_preamble:TF.)

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

```

217 \cs_new:Npn \ctex_set_default_ccwd:Nn #1#2
218 {
219   \dim_compare:nNnTF
220     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
221     { \tl_set:Nn #1 {#2} }
222     { \tl_set:Nn #1 { #2 \ccwd } }
223 }

```

(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`。

```

224 \cs_new:Npn \ctex_default_pt:n #1
225 {
226   \exp_after:wN \__ctex_default_pt:w
227   \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
228 }
229 \group_begin:
230   \char_set_catcode_other:N \P
231   \char_set_catcode_other:N \T
232   \tex_lowercase:D
233   {
234     \group_end:
235     \cs_new:Npn \__ctex_default_pt:w #1 PT #2 \q_stop { #1 PT }
236   }

```

(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。

```

237 \tl_new:N \l__ctex_encoding_tl
238 \tl_set:Nx \l__ctex_encoding_tl
239 { \pdfTeX_if_engine:TF { GBK } { UTF8 } }

```

(End definition for `\l__ctex_encoding_tl`.)

`\g__ctex_zhmCJK_bool` 是否使用 `zhmCJK` 宏包。

```

240 \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。

```

241 \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。

```

242 \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`。

```

243 \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。

```

244 <class>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
245 <ctex>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfPackage }

```

(End definition for \ctex_at_end:n.)

\g__ctex_std_options_clist 保存传递给标准文档类的选项。

```
246 < *class>
247 \clist_new:N \g__ctex_std_options_clist
248 < /class>
```

(End definition for \g__ctex_std_options_clist.)

对无效选项给出警告。

```
249 \msg_new:nnn { ctex } { invalid-option }
250 { Option~`~\l_keys_key_tl'~is~invalid~in~current~mode. }
251 \msg_new:nnn { ctex } { invalid-value }
252 { Value~`~#1'~is~invalid~for~the~key~`~\l_keys_key_tl'. }
```

对过时选项或命令给出警告。

```
253 \msg_new:nnn { ctex } { deprecated-option }
254 { Option~`~\l_keys_key_tl'~is~deprecated.\\ #1 }
255 \msg_new:nnn { ctex } { deprecated-command }
256 { Command~ #1 is~ deprecated.\\ #2 }
257 \msg_new:nnn { ctex } { deprecated-environment }
258 { Environment~`~#1'~is~deprecated.\\ #2 }
259 < /class|ctex>
```

\g__ctex_font_size_flag 0 表示修改默认字体大小为五号, 1 为小四号, 其它值则不作修改。

```
260 < *class|ctex|ctexsize>
261 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one
262 < /class|ctex|ctexsize>
```

(End definition for \g__ctex_font_size_flag.)

15.2 宏包选项

```
263 < *class|style>
264 \keys_define:nn { ctex / option }
265 {
266 < /class|style>
```

zihao

```
267 < *class|ctex|ctexsize>
268 zihao .choice: ,
269 zihao .default:n = { 5 } ,
270 zihao / 5 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero } ,
271 zihao / -4 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_one } ,
272 zihao / false .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
273 < ctexsize> }
274 < /class|ctex|ctexsize>
275 < *class|ctex>
276 c5size .code:n =
277 {
278 \msg_warning:nnn { ctex } { deprecated-option }
279 { Option~`~zihao=5'~is~set. }
280 \keys_set:nn { ctex / option } { zihao = 5 }
281 } ,
282 cs4size .code:n =
283 {
284 \msg_warning:nnn { ctex } { deprecated-option }
285 { Option~`~zihao=-4'~is~set. }
286 \keys_set:nn { ctex / option } { zihao = -4 }
287 } ,
288 c5size .value_forbidden: ,
289 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。

```
290 linespread .fp_set:N = \l__ctex_line_spread_fp ,
291 linespread .initial:n = { \c_nan_fp } ,
292 linespread .value_required: ,
```

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

autoindent 自动调整段落的首行缩进功能。

```
293 autoindent .choice: ,
294 autoindent .default:n = { true } ,
295 autoindent / true .code:n =
296 {
297   \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
298   \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
299 } ,
300 autoindent / false .code:n =
301 {
302   \tl_clear:N \l__ctex_autoindent_tl
303   \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
304 } ,
305 autoindent / unknown .code:n =
306 {
307   \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
308   \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
309 } ,
```

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

indent 仅为兼容性保留, 已过时。

```
310 indent .code:n =
311 {
312   \msg_warning:nnn { ctex } { deprecated-option }
313   {
314     The~ functionality~ has~ been~ removed.~
315     It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
316     options.~
317   }
318 } ,
319 indent .value_forbidden: ,
320 noindent .code:n =
321 {
322   \msg_warning:nnn { ctex } { deprecated-option }
323   {
324     The~ functionality~ has~ been~ removed.~
325     It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
326     options.~
327   }
328 } ,
329 noindent .value_forbidden: ,
```

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

GBK

UTF8

```
330 GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
331 UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
332 GBK .value_forbidden: ,
333 UTF8 .value_forbidden: ,
```

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

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

```

334     fontset      .tl_gset:N = \g__ctex_fontset_tl ,
335     nofonts      .code:n =
336     {
337         \msg_warning:nnn { ctex } { deprecated-option }
338         {
339             Option~`fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
340             fontset~ option.
341         }
342         \keys_set:nn { ctex / option } { fontset = none }
343     } ,
344     adobefonts    .code:n =
345     {
346         \msg_warning:nnn { ctex } { deprecated-option }
347         {
348             Option~`fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
349             fontset~ option.
350         }
351         \keys_set:nn { ctex / option } { fontset = adobe }
352     } ,
353     winfonts      .code:n =
354     {
355         \msg_warning:nnn { ctex } { deprecated-option }
356         {
357             Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
358             fontset~ option.
359         }
360         \keys_set:nn { ctex / option } { fontset = windows }
361     } ,
362     nofonts      .value_forbidden: ,
363     winfonts      .value_forbidden: ,
364     adobefonts    .value_forbidden: ,

```

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

zhmap

```

365     zhmap .choice: ,
366     zhmap .default:n = { true } ,
367     zhmap / zhmCJK .code:n =
368     {
369         \bool_gset_true:N \g__ctex_zhmCJK_bool
370         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
371     } ,
372     zhmap / true .code:n =
373     {
374         \bool_gset_false:N \g__ctex_zhmCJK_bool
375         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
376     } ,
377     zhmap / false .code:n =
378     {
379         \bool_gset_false:N \g__ctex_zhmCJK_bool
380         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
381     } ,
382     nozhmap .code:n =
383     {
384         \msg_warning:nnn { ctex } { deprecated-option }
385         { Option~`zhmap=false'~ is~ set. }
386         \keys_set:nn { ctex / option } { zhmap = false }
387     } ,
388     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。

```

389     punct .tl_set:N = \l__ctex_punct_tl ,
390     punct .default:n = { quanjiao } ,
391     punct .initial:n = { quanjiao } ,
392     nopunct .code:n =
393     {
394         \msg_warning:nnn { ctex } { deprecated-option }
395         { Option~`punct=plain'~ is~ set. }
396         \keys_set:nn { ctex / option } { punct = plain }
397     } ,
398     nopunct .value_forbidden: ,

```

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

space

```

399     space .choices:nn =
400     { true , auto , false }
401     {
402         \exp_args:Nx \ctex_at_end:n
403         { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
404     } ,
405     space .default:n = { true } ,
406     nospace .code:n =
407     {
408         \msg_warning:nnn { ctex } { deprecated-option }
409         { Option~`space=false'~ is~ set. }
410         \keys_set:nn { ctex / option } { space = false }
411     } ,
412     nospace .value_forbidden: ,

```

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

heading

```

413     heading .bool_set:N = \l__ctex_heading_bool ,

```

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

```

414 </class>ctex>
415 <*class>ctex|ctexheading>

```

sub3section

sub4section

```

416     sub3section .code:n =
417     { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
418     sub4section .code:n =
419     { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
420     sub3section .value_forbidden: ,
421     sub4section .value_forbidden: ,

```

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

scheme

```

422     scheme .tl_set:N = \l__ctex_scheme_tl ,
423 <*ctexheading>
424     scheme .default:n = { plain } ,
425     scheme .initial:n = { plain }
426 }
427 </ctexheading>
428 <!*ctexheading>
429     scheme .default:n = { chinese } ,
430     scheme .initial:n = { chinese } ,
431 </!ctexheading>

```

```

432 </class>ctex<ctexheading>
433 <*class>ctex>

434 cap .code:n =
435 {
436   \msg_warning:nnn { ctex } { deprecated-option }
437   { Option~`scheme = chinese' ~ is~ set. }
438   \keys_set:nn { ctex / option } { scheme = chinese }
439 } ,
440 nocap .code:n =
441 {
442   \msg_warning:nnn { ctex } { deprecated-option }
443   { Option~`scheme = plain' ~ is~ set. }
444   \keys_set:nn { ctex / option } { scheme = plain }
445 } ,
446 cap .value_forbidden: ,
447 nocap .value_forbidden: ,

```

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

fntef 这些都是过时的宏包兼容选项，原选项功能总是打开的。

```

448 fntef .code:n =
449 {
450   \msg_warning:nnn { ctex } { deprecated-option }
451   { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
452 } ,
453 fancyhdr .code:n =
454 {
455   \msg_warning:nnn { ctex } { deprecated-option }
456   { `fancyhdr'~ package~ is~ loaded. }
457   \RequirePackage { fancyhdr }
458 } ,
459 hyperref .code:n =
460 {
461   \msg_warning:nnn { ctex } { deprecated-option }
462   { `hyperref'~ package~ will~ be~ loaded. }
463   \ctex_at_end:n
464   {
465     \cs_if_exist:NF \hypersetup
466     { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
467   }
468   \AtEndPreamble { \RequirePackage { hyperref } }
469 } ,
470 }

```

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

```

471 </class>ctex>
472 <*class>ctex<ctexsize>

```

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

```

12pt
473 \tl_clear:N \l__ctex_tmp_tl
474 \clist_map_inline:nn
475 {
476   10pt , 11pt , 12pt ,
477   8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
478 }
479 {
480   \tl_put_right:Nn \l__ctex_tmp_tl
481   {
482     #1 .code:n =
483     { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,

```

```

485 </!class>
486 <*class>
487     {
488         \cs_gset_eq:Nn \g__ctex_font_size_flag \c_two
489         \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
490     } ,
491 </class>
492     #1 .value_forbidden: ,
493 }
494 }
495 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
496 \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。

```

497 <*class>
498 \keys_define:nn { ctex / option }
499 {
500     unknown .code:n =
501     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
502 }
503 </class>
504 <!ctexsize>\ctex_file_input:n { ctexopts.cfg }
505 </class|ctex|ctexsize>
506 <*class|style>
507 \ProcessKeysOptions { ctex / option }
508 </class|style>
509 <*class>

```

五号字使用标准文档类的 10pt 字体大小设置,小四号字则使用 12pt。

```

510 \if_case:w \g__ctex_font_size_flag
511     \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
512 \or:
513     \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
514 \fi:

```

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

```

515 <*article>
516 \tl_const:Nn \c__ctex_class_tl { article }
517 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
518 \LoadClass { article }
519 </article>
520 <*book>
521 \tl_const:Nn \c__ctex_class_tl { book }
522 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
523 \LoadClass { book }
524 </book>
525 <*report>
526 \tl_const:Nn \c__ctex_class_tl { report }
527 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
528 \LoadClass { report }
529 </report>
530 </class>

```

15.3 用户设置接口

\ctexset

```

531 <*class|ctex|ctexheading>
532 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
533 </class|ctex|ctexheading>

```

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

`\CTEXsetup` 过时命令。
`\CTEXoptions`

```

534 (*class|ctex)
535 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
536 {
537   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
538   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
539   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
540 }
541 \NewDocumentCommand \CTEXoptions { +o }
542 {
543   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
544   { \ctexset~ {~ #1~ }~ is~ set. }
545   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
546 }
547 </class|ctex>

```

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

15.4 特定引擎支持与设置

15.4.1 `ctex-engine-pdfTeX.def`

548 `<*pdfTeX>`

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

```

549 \seq_new:N \c__ctex_cmap_encoding_seq
550 \seq_set_from_clist:Nn \c__ctex_cmap_encoding_seq
551 { C19 , C10 , C00 , C09 , C40 , C60 }

```

(End definition for `\c__ctex_cmap_encoding_seq`.)

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

```

552 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2
553 {
554   \cs_if_free:cF { #1 + #2 }
555   {
556     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
557     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
558   }
559 }
560 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
561 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn

```

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

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

```

562 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
563 {
564   \cs_if_free:NF \CJK@plane
565   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
566 }
567 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
568 {
569   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
570   #1
571 }
572 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
573 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
574 {

```

```

575 \tex_lowercase:D { \tl_set:Nx \l__ctex_tmp_tl { #2 \CJK@plane } }
576 \tex_immediate:D \pdfobj stream ~ file { \l__ctex_tmp_tl .cmap }
577 \cs_new_protected_nopar:Npx #1
578 {
579     \exp_not:N \pdffontattr \exp_not:N \font
580     { /ToUnicode ~ \int_use:N \pdflastobj \c_space_tl 0 ~ R }
581 }
582 }

```

(End definition for \ctex_add_cmap:n.)

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

```

583 \group_begin:
584 \char_set_catcode_other:N \#
585 \ctex_if_pdfmode:TF
586 {
587     \group_end:
588     \ExplSyntaxOff
589     \ctex_appto_cmd:NnTF \DeclareFontFamily { \CTEX@Family@CMap {#1} {#2} }
590     {
591         \ctex_at_end_package:nn { cmap }
592         { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
593     }
594     { \ctex_patch_failure:N \DeclareFontFamily }
595     \ExplSyntaxOn
596 }
597 { \group_end: }

```

(End definition for \DeclareFontFamily.)

首先检查选项,决定是否载入 zhmCJK 宏包。

```

598 \if_bool:N \g__ctex_zhmCJK_bool
599 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
600 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

601 \else:
602     \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
603     { \RequirePackage { CJK } }
604     { \RequirePackage { CJKutf8 } }
605     \RequirePackage { CJKpunct , CJKspace }

```

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

```

606 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
607 {
608     \tl_set:Nn \CJKrmdefault {#1}
609     \tl_set:Nn \CJKsfdefault {#2}
610     \tl_set:Nn \CJKttdefault {#3}
611     \AtBeginDvi { \file_input:n {#4} }
612     \ctex_at_end_package:nn { atbegshi }
613     { \AtBeginShipoutFirst { \file_input:n {#4} } }
614 }
615 \onlypreamble \ctex_load_zhmap:n

```

(End definition for \ctex_load_zhmap:nnnn.)

```

616 \tl_if_exist:NF \CJKfamilydefault
617 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
618 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
619 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
620 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
621 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
622 { \ctex_patch_failure:N \rmfamily }
623 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }

```

```

624 { \ctex_patch_failure:N \sffamily }
625 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
626 { \ctex_patch_failure:N \ttfamily }
627 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
628 { \cs_set_eq:NN \reset@font \normalfont }
629 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```
630 \fi:
```

`\ctex_CJK_input:n` 和 `\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。

```

631 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
632 {
633   \ctex_file_wrapper:nnn
634   {
635     \char_set_catcode_other:n { 60 } % <
636     \char_set_catcode_math_superscript:n { 94 } % ^
637     \int_set_eq:NN \tex_endlinechar:D \c_minus_one
638   }
639   { \file_input:n {#1} }
640   {
641     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
642     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
643     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
644   }
645 }
646 \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` 和 `\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。

```

647 \if_cs_exist:N \CJK@surr
648 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
649 {
650   \tl_set:Nx \l__ctex_tmp_tl {#2}
651   \int_set:Nn \l__ctex_tmp_int
652   { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
653   \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
654   { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
655   {
656     \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
657     \tl_gset:Nx #1
658     {
659       \int_to_Hex:n
660       { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
661       \int_to_Hex:n
662       { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
663     }
664   }
665 }
666 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
667 \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`。

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

```

668 \AtBeginDocument
669 {
670   \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
671   { \punctstyle { \l__ctex_punct_tl } }
672 }

```

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。

```

673 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
674 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
675 {
676   \CJK@makeActive
677   \CJK@loadBinding { standard }
678 }

```

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。

```

679 \exp_args:Nx \AtEndPreamble
680 {
681   \exp_not:N \CJK@envStart
682   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
683   \exp_not:N \CJKtilde
684 }

```

`zhmCJK` 判断结束。

```

685 \fi:

```

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

```

686 <@=>
687 \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。

```

688 <@=-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。

```
689 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
690 {
691   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
692     { \cs_set_eq:NN \CJK@ignorespaces #1 }
693   \ctex_set_ignorespaces:
694 }
695 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
696 { \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。

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

(End definition for \CJKhook.)

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

```
698 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
699 {
700   \clist_map_inline:Nn \c__ctex_punct_family_clist
701   {
702     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
703     {
704       \cs_set_eq:cc
705       { CJKpunct@ ##1 @spaces }
706       { c__ctex_ #1 ##1 _punct_spaces_tl }
707     }
708   }
709 }
710 \clist_const:Nn \c__ctex_punct_family_clist
711 {
712   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
713   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
714 }
```

(End definition for \ctex_punct_set:n.)

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

```
715 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
716 {
717   \cs_if_free:cF { CJKpunct@ #2 @spaces }
718   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
719 }
```

(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。

```
720 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
721 {
722   \clist_map_inline:nn {#1}
723   {
724     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
725     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
726   }
727 }
```

```

728 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
729 {
730   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
731   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
732   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
733   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
734   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
735   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
736 }

```

(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。

```

737 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
738 {
739   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
740   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
741   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
742   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
743   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
744   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
745 }

```

(End definition for \ctex_punct_map_itshape:nn.)

\ctex_punct_space:nn 定义标点的边界信息。

\ctexspadef

```

746 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
747 { \tl_const:cn { c_--ctex_ #1 _punct_spaces_tl } {#2} }
748 \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。

```

749 \ctex_file_input:n { ctexspa.def }

750 </pdfTeX>

```

15.4.2 ctex-engine-xetex.def

```

751 <*xetex>
752 \RequirePackage { xeCJK }
753 \exp_args:Nx \xeCJKsetup
754 {
755   LoadFandol = false ,
756   AutoFakeBold = true ,
757   PunctStyle = \l__ctex_punct_tl
758 }

```

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。

```

759 \@ifpackagelater { fontspec } { 2014/05/25 } { }
760 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
761 </xetex>

```

15.4.3 ctex-engine-luatex.def

```

762 <*luatex>

```

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

```

763 \msg_new:nnn { ctex } { luatexja-loaded }

```

```

764 {
765   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
766   Loading~file~`#1'~will~abort!
767 }
768 \@ifpackageloaded { luatexja }
769 { \msg_critical:nx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
770 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
771 \RequirePackage { luatexja }
772 \@ifpackagelater { luatexja-core } { 2015/02/22 } { }
773 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
774 \RequirePackage { fontspec }
775 \@ifpackagelater { fontspec } { 2014/05/25 } { }
776 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

15.4.3.1 LuaTeX-jā 的默认设置

```

777 \ExplSyntaxOff

```

以下设置抄录自 `l1tjdefs.sty`。

```

778 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
779 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
780 \ltjdefcharrange{3}{%
781   "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
782 \ltjdefcharrange{4}{%
783   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
784   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
785   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
786 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
787 \ltjdefcharrange{6}{%
788   "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
789   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
790 \ltjdefcharrange{7}{
791   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
792   "AC00-"D7FF}
793 \ltjdefcharrange{8}{ "A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
794 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, +8}}
795 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 `ltj-latex.sty`。

```

796 \directlua{
797   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
798   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
799 }
800 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .4pt,
801   xkanjiskip=.25\zw plus 1pt minus 1pt,
802   autospacing, autoxspacing, jacharrange={-1},
803   yjabaselineshift=\z@, yjabaselineshift=\z@,
804   jcharwidowpenalty=500, differentjfm=paverage
805 }
806 \ExplSyntaxOn

```

15.4.3.2 LuaTeX-jā 的补丁

```

807 <@@=ctex_l1tj>

```

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

```

808 \RequirePackage { xunicode-addon }
809 \AtBeginUTFCommand
810 {
811   \group_begin:
812   \ctex_lua_now_x:n { tex.globaldefs = 0 }
813   \ltj@allalchar
814 }

```

```
815 \AtEndUTFCommand { \group_end: }
```

对 fontspec 沿用 LuaTeX-ja 的补丁。

```
816 \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。

```
817 \group_begin:
818 \char_set_catcode_other:n { \c_zero }
819 \cs_new_protected:Npn \__ctex_ltj_um_char:Nw #1 = #2 \q_nil
820 {
821   \group_begin:
822   \char_set_lccode:nn { \c_zero } {#2}
823   \tex_lowercase:D
824   {
825     \group_end:
826     \cs_gset_protected_nopar:Npn #1
827     {
828       \mode_if_math:TF { ^^@ }
829       { {
830         \ctex_lua_now_x:n { tex.globaldefs = 0 }
831         \ltj@allalchar ^^@
832       } }
833     }
834   }
835   \ltjsetmathletter {#2}
836 }
837 \group_end:
838 \ctex_at_end_package:nn { unicode-math }
839 {
840   \cs_if_free:NF \um_cs_set_eq_active_char:Nw
841   { \cs_set_eq:NN \um_cs_set_eq_active_char:Nw \__ctex_ltj_um_char:Nw }
842 }
```

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

```
843 \ctex_at_end_package:nn { listings }
844 {
845   \use:x
846   {
847     \exp_not:N \RequirePackage { lltjp-listings }
848     \tl_set:Nn \exp_not:N \lstlistingname
849     { \exp_not:o { \lstlistingname } }
850     \tl_set:Nn \exp_not:N \lstlistlistingname
851     { \exp_not:o { \lstlistlistingname } }
852   }
853 }
```

15.4.3.3 字体切换方式

\ctex_ltj_select_font: \CJK@family 保存的是当前 CJK 实际的字体族名,如果为空表示没有设置过字体。

```
\CJK@family
854 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
855 {
856   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
857   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
858 }
859 \tl_new:N \CJK@family
860 \tl_new:N \l__ctex_ltj_current_font_tl
861 \tl_set:Nn \l__ctex_ltj_current_font_tl
862 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }
```

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

__ctex_ltj_select_font_aux: 使用 \pickup@font 取得字体名称前,总需要先设置 \font@name。在这里将 \f@family 换成

CJK 字体族,并确保编码正确。

```

863 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
864 {
865   \group_begin:
866   \tl_set_eq:NN \f@encoding \CJK@encoding
867   \tl_set_eq:NN \f@family \CJK@family
868   \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
869   \ctex_ltj_pickup_font:
870   \group_end:
871   \font@name
872   \__ctex_ltj_pop_fontname:

```

当字形未定义的时候, NFSS 就会启动替换机制(\wrong@fontshape)。第一次启动后, \l__ctex_ltj_current_font_tl 还是没有定义。为此,我们再次选择字体,确保它有定义和指向正确的 font.id。这对 AlternateFont 的设置特别重要。

```

873   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
874   { \__ctex_ltj_select_font_aux: }
875 }
876 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
877 {
878   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
879   \cs_gset_nopar:Npx \font@name {#1}
880 }
881 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
882 { \cs_gset_eq:NN \font@name \__ctex_ltj_save_fontname:w }

```

(End definition for __ctex_ltj_select_font_aux:.)

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

```

883 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
884 {
885   \exp_after:wN \cs_if_exist:NF \font@name
886   {
887     \group_begin:
888     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
889     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
890     \define@newfont
891     \group_end:
892   }
893 }
894 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

(End definition for \ctex_ltj_pickup_font:.)

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

```

%%%%\jfont\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 ',
  cstemp, '\\endcsname{\\ltj@cur'..
  (jfm_dir == 'yoko' and 'j' or 't') .. 'fnt', fn, '\\relax}')

```

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

```

895 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
896 {

```

```

897 \get@external@font
898 \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
899 {
900   \tl_set:Nx \external@font
901   { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
902 }
903 \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。

```

904 \font@name
905 \ctex_lua_now_x:n { font.current(tex.getattribute('ltj@curjfont')) }
906 \use:c { \f@encoding + \f@family }
907 \use:c { \curr@fontshape }
908 }

```

(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。

```

909 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
910 {
911   \ctex_ltj_if_alterate_shape_exist:nF { \curr@fontshape }
912   {
913     \group_begin:
914     \tl_set_eq:NN \CJK@family \f@family
915     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
916     {
917       \cs_gset_protected_nopar:Npx \subst@correction
918       {
919         \cs_new_eq:NN
920         \exp_not:c { \l__ctex_ltj_current_font_tl }
921         \font@name
922       }
923       \group_insert_after:N \group_insert_after:N
924       \group_insert_after:N \subst@correction
925     }
926     \group_end:
927   }
928 }

```

(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。

```

929 \prg_new_conditional:Npnn \ctex_ltj_if_alterate_shape_exist:n #1 { T , F , TF }
930 {
931   \ctex_lua_now_x:n { luatexja.jfont.does_alt_set ('\\luatexluaescapestring {#1}') }
932   \prg_return_true: \else: \prg_return_false: \fi:
933 }

```

(End definition for \ctex_ltj_if_altername_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。

```
934 \cs_new_nopar:Npn \_ctex_ltj_patch_external_font:w #1 ~ at
935   { #1 \ctex_lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }
```

(End definition for _ctex_ltj_patch_external_font:w.)

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

```
936 \cs_new_protected_nopar:Npn \ctex_ltj_select_altername_font:
937   {
938     \ctex_ltj_if_altername_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
939     {
940       \ctex_lua_now_x:n
941       {
942         luatexja.jfont.output_alt_font_cmd
943         ('y', '\luatexluaescapestring { \l__ctex_ltj_current_shape_tl }')
944       }
945       \ctex_lua_now_x:n { luatexja.jfont.pickup_alt_font_a ('f@size') }
946     }
947   }
948   \tl_new:N \l__ctex_ltj_current_shape_tl
949   \tl_set:Nn \l__ctex_ltj_current_shape_tl
950   { \CJK@encoding / \CJK@family / \f@series / \f@shape }
```

(End definition for \ctex_ltj_select_altername_font:.)

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

```
951 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
952   {
953     \cs_if_exist:cF { #1/f@size }
954     {
955       \group_begin:
956       \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
957       \_ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
958       \ctex_ltj_pickup_font:
959       \group_end:
960       \_ctex_ltj_pop_fontname:
961     }
962   }
```

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

963 <@@=)

\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。

```
964 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
965   {
966     \ltj@@getjfontnumber #1
967     \ctex_lua_now_x:n
968     {
969       luatexja.jfont.pickup_alt_font_b
970       ( \the\ltj@tempcntc, '\luatexluaescapestring {#2}' )
```

```

971     }
972 }

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

973 \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。

```

974 \def\ltj@@IsFontJapanese#1{%
975   \directlua{luatexja.jfont.is_kenc(string.match(
976     '\luatexluaescapestring{#1}', '[~/]+'))}}
977 {\catcode`M=12%
978 \gdef\ltj@@mathJapaneseFonts#1M#2#3\relax{\ltj@@IsFontJapanese{#3}}
979 \let\ltj@@al@getanddefine@fonts=\getanddefine@fonts
980 \def\ltj@@ja@getanddefine@fonts#1#2{%
981   \xdef\font@name{\csname \string#2/\tf@size\endcsname}%
982   \pickup@jfont\let\textfont@name\font@name
983   \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
984   \pickup@jfont\let\scriptfont@name\font@name
985   \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
986   \pickup@jfont
987   \edef\math@fonts{\math@fonts\ltj@setpar@global%
988     \ltj@@set@stackfont#1,\textfont@name:{MJT}}%
989   \ltj@@set@stackfont#1,\scriptfont@name:{MJS}}%
990   \ltj@@set@stackfont#1,\font@name:{MJSS}}%
991 }%
992 }
993 \def\getanddefine@fonts#1#2{%
994   \ltj@tempcnta=#1\ltj@@IsFontJapanese{\string#2}%
995   \ifin@let\ltj@temp=\ltj@@ja@getanddefine@fonts%
996   \else \let\ltj@temp=\ltj@@al@getanddefine@fonts\fi
997   \ltj@temp{#1}{#2}%
998 }
999 \def\use@mathgroup#1#2{\relax\ifmmode
1000   \math@bgroup
1001   \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
1002     #1\fi\ltj@tempcnta=#2 \expandafter\ltj@@mathJapaneseFonts\string#1\relax%
1003     \ifin@jfam#2\relax\else\mathgroup#2\relax\fi
1004   \expandafter\math@egroup\fi}%

1005 \let\@@italiccorr=\/

1006 <@@=ctex_ltj>

1007 \ExplSyntaxOn

```

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

```

1008 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
1009 {
1010   \str_if_eq:nnTF {#1} { plain }
1011   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
1012   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
1013 }
1014 \tl_new:N \l__ctex_ltj_jfm_tl
1015 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
1016 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

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

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

```

1017 \tl_const:Nn \CJK@encoding { LTJY3 }
1018 \DeclareFontEncoding { \CJK@encoding } { } { }
1019 \DeclareFontSubstitution { LTJY3 } { song } { \mddefault } { \updefault }
1020 \ctex_lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
1021 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
1022   { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
1023 \DeclareFontFamily { \CJK@encoding } { song } { }
1024 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
1025   { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1026 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
1027   { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1028 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
1029 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
1030   { \CJK@encoding } { song } { \mddefault } { \updefault }
1031 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
1032   { \CJK@encoding } { song } { \bfdefault } { \updefault }
1033 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
1034 \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。

```

1035 \newfontfeature { CID }      { cid = #1 }
1036 \newfontfeature { JFM }      { jfm = #1 }
1037 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

1038 \keys_define:nn { fontspec-preparse-external }
1039   {
1040     NoEmbed .code:n =
1041       { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
1042   }
1043 \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。

```

1044 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1045   {
1046     \group_begin:
1047     \clist_clear:N \l__ctex_ltj_char_range_clist
1048     \seq_clear:N \l__ctex_ltj_alternate_seq
1049     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1050     \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1051     \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1052     \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1053     {
1054       \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1055       \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1056         {#1} { \l__ctex_ltj_font_options_clist }
1057       \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1058       \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1059       \__ctex_ltj_gset_family_cs:nn {#1} {#3}
1060     }
1061     \group_end:
1062   }
1063 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1064   { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1065 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1066 \clist_new:N \l__ctex_ltj_font_options_clist
1067 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
1068   {
1069     \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1

```

```

1070 \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1071 }

```

(End definition for \ctex_ltj_set_family:nnn.)

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

```

1072 \prop_new:N \g__ctex_ltj_family_name_prop
1073 \prop_new:N \g__ctex_ltj_family_font_name_prop
1074 \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。

```

1075 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
1076 {
1077   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1078   {
1079     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1080     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1081     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1082     {
1083       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1084       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1085       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1086       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1087     }
1088     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1089   }
1090 }
1091 \tl_new:N \l__ctex_ltj_tmp_tl
1092 \msg_new:nnn { ctex } { redefine-family }
1093 { 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。

```

1094 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1095 {
1096   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }
1097   {
1098     \group_begin:
1099     \__ctex_ltj_change_encoding:
1100     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1101     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1102     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1103     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1104     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1105     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1106     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1107     \__ctex_ltj_set_alternate_family:n {#1}
1108     \group_end:
1109   }
1110 }
1111 \tl_new:N \l__ctex_ltj_base_family_tl
1112 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1113 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1114 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1115 {
1116   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1117   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1118   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }

```

```

1119 \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n {#1} }
1120 }
1121 \cs_new:Npn \__ctex_ltj_altername_cs:n #1 { ctex_ltj/altername_family/#1 }

(End definition for \__ctex_ltj_gset_family_cs:nn.)

```

\CJKfamily 切换字体。

```

1122 \NewDocumentCommand \CJKfamily { m }
1123 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1124 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1125 {
1126   \ctex_ltj_family_if_exist:NNTF {#1} \CJK@family
1127   {
1128     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1129     \selectfont
1130   }
1131   { \__ctex_ltj_family_unknown_warning:n {#1} }
1132 }
1133 \tl_new:N \l_ctex_ltj_family_tl
1134 \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。

```

1135 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1136 {
1137   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1138   { \prg_return_true: }
1139   {
1140     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1141     {
1142       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1143       \prg_return_true:
1144     }
1145     { \prg_return_false: }
1146   }
1147 }
1148 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

(End definition for \ctex_ltj_family_if_exist:nTF.)

```

__ctex_ltj_family_unknown_warning:n

```

1149 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1150 {
1151   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1152   {
1153     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1154     {
1155       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1156       \msg_warning:nnn { ctex } { family-unknown } {#1}
1157     }
1158   }
1159 }
1160 \seq_new:N \g__ctex_ltj_unknown_family_seq
1161 \msg_new:nnn { ctex } { family-unknown }
1162 {
1163   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1164   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1165 }
1166 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1167 {
1168   \str_case_x:nnF {#1}
1169   {

```

```

1170         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1171         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1172         \CJKttdefault { \token_to_str:N \setCJKmonofont }
1173     }
1174     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1175     [...] \{...\}
1176 }
1177 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1178 {
1179     \str_case:x:nnF {#1}
1180     {
1181         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1182         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1183         \CJKttdefault { \token_to_str:N \CJKttdefault }
1184     }
1185     {#1}
1186 }

```

(End definition for __ctex_ltj_family_unknown_warning:n.)

\ctex_ltj_fontspec:nn

```

1187 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1188 {
1189     \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1190     { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1191     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1192     {
1193         \int_gincr:N \g__ctex_ltj_family_int
1194         \__ctex_ltj_fontspec:xnn
1195         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1196         {#1} {#2}
1197     }
1198 }
1199 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1200 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1201 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1202 {
1203     \bool_if:NT \l__ctex_ltj_add_alterate_bool
1204     {
1205         \cs_if_free:cF
1206         { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1207         {
1208             \cs_gset_eq:cc
1209             { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1210             { \__ctex_ltj_alterate_cs:n { reset / \l_ctex_ltj_family_tl } }
1211             \cs_gset_eq:cc
1212             { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1213             { \__ctex_ltj_alterate_cs:n { clear / \l_ctex_ltj_family_tl } }
1214         }
1215         \bool_set_false:N \l__ctex_ltj_add_alterate_bool
1216     }
1217     \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1218     \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1219     \ctex_ltj_switch_family:n {#1}
1220 }
1221 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1222 \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

```

1223 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1224 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1225 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1226 {

```

```

1227 \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1228 {#1} \l__ctex_ltj_tmp_tl
1229 {
1230   \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1231   {#1} \l__ctex_ltj_font_options_clist
1232   \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1233   \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1234   \ctex_ltj_fontspec:xx
1235   { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1236   { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1237 }
1238 { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1239 }
1240 \bool_new:N \l__ctex_ltj_add_alterate_bool
1241 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1242 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1243 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1244 {
1245   \token_to_str:N \addCJKfontfeature (s)~ignored.\
1246   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1247 }

```

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

```

\setCJKfamilyfont
\newCJKfontfamily
\CJKfontspec
\addCJKfontfeatures
1248 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
1249 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1250 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
1251 {
1252   \tl_set:Nx \l__ctex_ltj_tmp_tl
1253   { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1254   \cs_new_protected_nopar:Npx #2
1255   { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1256   \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1257 }
1258 \NewDocumentCommand \CJKfontspec { O { } m }
1259 {
1260   \ctex_ltj_fontspec:xx {#1} {#2}
1261   \tex_ignorespaces:D
1262 }
1263 \NewDocumentCommand \addCJKfontfeatures { m }
1264 {
1265   \ctex_ltj_add_font_features:x {#1}
1266   \tex_ignorespaces:D
1267 }
1268 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

(End definition for \setCJKfamilyfont and others.)

```

\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1269 \NewDocumentCommand \setCJKmainfont { O { } m }
1270 {
1271   \ctex_ltj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1272   \normalfont
1273 }
1274 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1275 \NewDocumentCommand \setCJKsansfont { O { } m }
1276 {
1277   \ctex_ltj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1278   \normalfont
1279 }
1280 \NewDocumentCommand \setCJKmonofont { O { } m }
1281 {
1282   \ctex_ltj_set_family:xxx { \CJKttdefault } {#1} {#2}
1283   \normalfont

```

```

1284 }
1285 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1286 { \ctex_ltj_set_family:xxx { \c__ctex_ltj_math_tl } {#1} {#2} }
1287 \NewDocumentCommand \defaultCJKfontfeatures { m }
1288 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1289 \clist_new:N \g__ctex_ltj_default_features_clist
1290 \@onlypreamble \setCJKmainfont
1291 \@onlypreamble \setCJKsansfont
1292 \@onlypreamble \setCJKmonofont
1293 \@onlypreamble \setCJKmathfont
1294 \@onlypreamble \setCJKromanfont
1295 \@onlypreamble \defaultCJKfontfeatures

(End definition for \setCJKmainfont and others.)

1296 \tl_if_exist:NF \CJKfamilydefault
1297 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1298 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1299 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1300 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1301 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
1302 { \ctex_patch_failure:N \rmfamily }
1303 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }
1304 { \ctex_patch_failure:N \sffamily }
1305 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
1306 { \ctex_patch_failure:N \ttfamily }
1307 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
1308 { \cs_set_eq:NN \reset@font \normalfont }
1309 { \ctex_patch_failure:N \normalfont }

```

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

```

1310 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1311 {
1312   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1313   {
1314     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmpa_tl
1315     {
1316       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1317       { \use:n }
1318       {
1319         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmpa_tl
1320         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1321         { \use:n }
1322       }
1323     }
1324     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1325     {
1326       \prop_map_break:n
1327       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1328     }
1329   }
1330 }
1331 \normalfont
1332 \ctex_ltj_update_mathfont:
1333 }
1334 }

```

(End definition for \ctex_ltj_ensure_default_family:.)

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

```

1335 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1336 {
1337   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmpa_tl
1338   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmpa_tl } }
1339   {

```

```

1340         \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1341         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1342     }
1343 }
1344 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1345 {
1346     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1347     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1348     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1349     \cs_if_free:cTF
1350     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1351     {
1352         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1353         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1354     }
1355     {
1356         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1357         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1358     }
1359 }

```

(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

```

1360 \keys_define:nn { ctex_ltj / fontspec }
1361 {
1362     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1363     AlternateFont .value_required: ,
1364     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1365     CharRange .value_required:
1366 }

```

(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。

```

1367 \group_begin:
1368 \char_set_catcode_active:N \ /
1369 \char_set_lccode:nn { \ / } { \ | }
1370 \tex_lowercase:D
1371 {
1372     \group_end:
1373     \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n #1
1374     {
1375         \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1376         {
1377             \tl_set:Nn \l__ctex_ltj_tmp_tl {#1}
1378             \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { // } { || }
1379             \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1380             \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1381             { ! \tl_if_blank_p:n { ##1 } }
1382             \seq_concat:NNN \l__ctex_ltj_alternate_seq
1383             \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1384         }
1385     }
1386 }
1387 \seq_new:N \l__ctex_ltj_tmp_seq
1388 \seq_new:N \l__ctex_ltj_alternate_seq

```

(End definition for \ctex_ltj_set_alternate_seq:n.)

`\ctex_ltj_set_alternate_family:nnF` 如果在字体的选项中设置了 CharRange, 则只设置替代字体。

```

1389 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1390 {
1391   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1392   {
1393     \__ctex_ltj_check_family:n {#1}
1394     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1395     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1396     #3
1397   }
1398   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1399 }

```

(End definition for `\ctex_ltj_set_alternate_family:nnF`.)

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

```

\ctex_ltj_save_alternate_seq:Nnnwn
1400 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1401 {
1402   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1403   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1404 }
1405 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1406 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1407 { m m m +0{ } m u{ \q_stop } }
1408 {
1409   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1410   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1411   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1412   \tl_if_blank:NTF {#5}
1413   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1414   {
1415     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1416     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1417   }
1418   \use:x
1419   {
1420     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1421     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1422     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1423     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1424   }
1425 }
1426 \clist_new:N \l__ctex_ltj_alternate_options_clist

```

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

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

```

1427 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1428 {
1429   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1430   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1431   \ctex_ltj_set_alternate_family:coonn
1432   { \__ctex_ltj_alternate_cs:n {#1} }
1433   { \l__ctex_ltj_char_range_clist }
1434   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1435 }
1436 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1437 {
1438   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1439   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1440   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1441 }
1442 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

(End definition for \ctex_ltj_set_alternate_family:nn.)

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

```
1443 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1444 {
1445   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1446   \cs_gset_protected_nopar:Npx #1
1447     { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1448 }
```

(End definition for \ctex_ltj_save_alternate_family:Nnnn.)

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

```
1449 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1450 {
1451   \group_begin:
1452   \__ctex_ltj_change_encoding:
1453   \cs_set_eq:NN \CJKfamily \use_none:n
1454   \ctex_ltj_swap_cs:NN
1455   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1456   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1457   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1458   \group_end:
1459 }
1460 \tl_new:N \l__ctex_ltj_alternate_family_tl
```

(End definition for \ctex_ltj_set_alternate_family:nnn.)

\ctex_ltj_swap_cs:NN 交换两个控制序列的意义。

```
1461 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1462 {
1463   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1464   \cs_set_eq:NN #1 #2
1465   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1466   \cs_undefine:N \__ctex_ltj_tmp:w
1467 }
```

(End definition for \ctex_ltj_swap_cs:NN.)

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

__ctex_ltj_update_family_uid:N

```
1468 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1469 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1470 {
1471   \int_gincr:N \g__ctex_ltj_family_int
1472   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1473 }
1474 \int_new:N \g__ctex_ltj_family_int
```

(End definition for LTJFONTUID and __ctex_ltj_update_family_uid:N.)

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

```
1475 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1476 {
1477   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1478   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1479     { \l__ctex_ltj_base_family_tl } {#3} {#4}
1480     { \l_fontspec_family_tl } {#3} {#4}
1481 }
```

(End definition for \ctex_ljt_declare_alter_nate_shape:nnnnnn.)

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

```

1482 \cs_new_protected_nopar:Npn \ctex_ljt_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1483 {
1484   \clist_map_inline:Nn #1
1485   {
1486     \prop_get:NnNTF \g__ctex_ljt_char_range_prop { ##1 } \l__ctex_ljt_char_range_tl
1487     {
1488       \ctex_ljt_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1489       \l__ctex_ljt_char_range_tl
1490     }
1491     { \ctex_ljt_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1492   }
1493   \__ctex_ljt_save_alter_nate_shape:cnn
1494   { \__ctex_ljt_alter_nate_cs:n { clear / \l__ctex_ljt_base_CJKfamily_tl } }
1495   { luatexja.jfont.clear_alt_font_latex }
1496   { '\luatexluaescapestring { \CJK@encoding/#2/#3/#4 }' }
1497 }

```

(End definition for \ctex_ljt_set_alter_nate_shape:Nnnnnnn.)

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

```

1498 \NewDocumentCommand \ctex_ljt_set_alter_nate_shape:nnn
1499 { m m > { \SplitArgument { \c_one } { -> } } m }
1500 { \ctex_ljt_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1501 \cs_new_protected_nopar:Npn \ctex_ljt_set_alter_nate_shape:nnnn #1#2#3#4
1502 {
1503   \ctex_ljt_set_alter_nate_shape:n
1504   {
1505     \IfNoValueTF {#4}
1506     { \int_eval:n {#3} , \int_eval:n {#3} , }
1507     {
1508       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1509       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1510     }
1511     '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1512     '\luatexluaescapestring { \CJK@encoding/#1 }'
1513   }
1514 }
1515 \cs_new_protected_nopar:Npn \ctex_ljt_set_alter_nate_shape:n #1
1516 {
1517   \ctex_lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1518   \__ctex_ljt_save_alter_nate_shape:cnn
1519   { \__ctex_ljt_alter_nate_cs:n { reset / \l__ctex_ljt_base_CJKfamily_tl } }
1520   { luatexja.jfont.set_alt_font_latex } {#1}
1521 }

```

(End definition for \ctex_ljt_set_alter_nate_shape:nnn.)

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

```

1522 \cs_new_protected_nopar:Npn \ctex_ljt_set_alter_nate_shape:nnN #1#2#3
1523 {
1524   \tl_map_inline:Nn #3
1525   {
1526     \ctex_ljt_set_alter_nate_shape:n
1527     {
1528       ##1 ,
1529       '\luatexluaescapestring { \CJK@encoding/#2 }' ,

```

```

1530         '\luatexluaescapestring { \CJK@encoding/#1 }'
1531     }
1532 }
1533 }

```

(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。

```

1534 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1535 {
1536     \group_begin:
1537     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1538     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1539     \cs_set_eq:NN \luatexluaescapestring \scan_stop:
1540     \cs_gset_protected_nopar:Npx #1
1541     { \exp_not:o {#1} \exp_not:N \ctex_lua_now_x:n { #2 ( #3 ) } }
1542     \group_end:
1543 }
1544 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nnn { c }

```

(End definition for __ctex_ltj_save_alternate_shape:Nnn.)

clearalternatefont
resetalternatefont

清除和重置操作总是全局的。

```

1545 \keys_define:nn { ctex }
1546 {
1547     clearalternatefont .code:n =
1548     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1549     resetalternatefont .code:n =
1550     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1551     clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1552     resetalternatefont .default:n = \l_ctex_ltj_family_tl
1553 }
1554 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1555 {
1556     \group_begin:
1557     \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1558     {
1559         \cs_if_exist_use:cT { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1560         {
1561             \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1562             {#1} { \l__ctex_ltj_base_family_tl }
1563             \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1564             \selectfont
1565         }
1566     }
1567     { \__ctex_ltj_family_unknown_warning:n {#1} }
1568     \group_end:
1569 }
1570 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1571 {
1572     \group_begin:
1573     \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1574     {
1575         \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1576         \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1577         \selectfont
1578     }
1579     \group_end:
1580 }
1581 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1582 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalternatfont` and `resetalternatfont`. These functions are documented on page 25.)

`declarecharrange` 预先声明字符范围。

```

1583 \keys_define:nn { ctex }
1584 {
1585   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1586   declarecharrange .value_required:
1587 }
1588 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1589 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1590 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1591 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1592 { \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`

```

1593 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1594 {
1595   \tl_clear:N \l__ctex_ltj_char_range_tl
1596   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1597   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1598   \ctex_ltj_def_char_range_key:n {#1}
1599   \tl_clear:N \l__ctex_ltj_char_range_tl
1600 }
1601 \tl_new:N \l__ctex_ltj_char_range_tl
1602 \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。

```

1603 \NewDocumentCommand \ctex_ltj_save_char_range:n
1604 { > { \SplitArgument { \c_one } { -> } } m }
1605 { \ctex_ltj_save_char_range:nn #1 }
1606 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1607 {
1608   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1609   { {
1610     \IfNoValueTF {#2}
1611     { \int_eval:n {#1} , \int_eval:n {#1} }
1612     {
1613       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1614       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1615     }
1616   } }
1617 }

```

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

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

```

1618 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1619 {
1620   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1621   {
1622     \keys_define:nn { ctex_ltj / fontspec }
1623     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1624   }
1625 }

```

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

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

```

1626 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1627 {
1628   \tl_if_blank:nTF {#2}
1629   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1630   {
1631     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1632     {
1633       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1634       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1635     }
1636   }
1637 }

```

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

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

```

1638 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1639 { +o o u { \q_stop } }
1640 {
1641   \exp_args:Nnf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1642   {
1643     \IfNoValueTF {#1} { {#3} }
1644     {
1645       \IfNoValueTF {#2}
1646       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ {#1} ] {#3} } }
1647       { [ {#1} ] { [ #2 ] } }
1648     }
1649   }
1650   \seq_put_right:No \l__ctex_ltj_alterate_seq { \l__ctex_ltj_tmp_tl }
1651 }

```

(End definition for `__ctex_ltj_char_range_parse_feature:w`.)

15.4.3.6 其它设置

`\ctex_ltj_update_xkanjiskip:` `\l__ctex_ltj_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。

```

1652 \cs_new_protected_nopar:Npn \ctex_ltj_update_xkanjiskip:
1653 {
1654   \skip_if_eq:nnT
1655   { \ltjgetparameter { xkanjiskip } } { \l__ctex_ltj_xkanjiskip_skip }
1656   {
1657     \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }
1658     \ltjsetxkanjiskip \l__ctex_ltj_xkanjiskip_skip
1659   }
1660 }

```

注意,此时还没有实际设置字体,所以 `\zw` 是 `0pt`。

```

1661 \skip_new:N \l__ctex_ltj_xkanjiskip_skip
1662 \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }

```

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

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

```

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

1663 \AtBeginDocument
1664 {
1665   \ctex_appto_cmd:NnTF \verbatim@font { \CTEX@verbatim@font@hook }
1666   { } { \ctex_patch_failure:N \verbatim@font }
1667 }
1668 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1669 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

1670 <@@=ctex>

1671 </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。

```

1672 <*pdfTeX|luatex>
1673 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1674 {
1675   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1676   {
1677     \group_begin:
1678     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1679     \tl_gset:Nx \CJKfamilydefault
1680     {
1681       \str_case:onF { \familydefault }
1682       {
1683         { \rmdefault } { \exp_not:N \CJKrmdefault }
1684         { \sfdefault } { \exp_not:N \CJKsfdefault }
1685         { \ttdefault } { \exp_not:N \CJKttdefault }
1686       }
1687       { \CJKfamilydefault }
1688     }
1689     \group_end:
1690   }

```

使用 Lua_{La}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。

```

1691 <*luatex>
1692   \ctex_ltx_ensure_default_family:
1693 </luatex>
1694 }
1695 \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。

```

1696 \tl_new:N \l__ctex_family_default_init_tl
1697 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1698 \tl_set:Nx \l__ctex_family_default_init_tl
1699 {

```

```

1700     \exp_not:N \__ctex_family_default_wrap:n
1701     { \exp_not:o { \CJKfamilydefault } }
1702   }
1703   \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1704   </pdf|luatex>

```

(End definition for \l__ctex_family_default_init_tl.)

15.4.5 操作系统的判断

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

```

1705 <(*luatex)>
1706 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1707 {
1708   \tl_gset:Nx \g__ctex_fontset_tl
1709   {
1710     \ctex_lua_now_x:n
1711     {
1712       if ~ os.name == 'windows' then ~
1713         tex.sprint ( 'windows' )
1714       elseif ~ os.name == 'macosx' then ~
1715         tex.sprint ( 'mac' )
1716       else ~
1717         tex.sprint ( 'fandol' )
1718       end
1719     }
1720   }
1721 }
1722 </luatex>

```

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

```

1723 <(*xetex|pdf|luatex)>
1724 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1725 {
1726   \file_if_exist:nTF { /dev/null }
1727   {
1728     \file_if_exist:nTF { nul: }
1729     {
1730       \file_if_exist:nTF { \c__ctex_engine_file_tl }
1731       { \ctex_if_macosx:TF { mac } { windows } }
1732       { \ctex_if_macosx:TF { mac } { fandol } }
1733     }
1734     { \ctex_if_macosx:TF { mac } { fandol } }
1735   }
1736   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1737 }
1738 \tex_uppercase:D \exp_after:wN
1739 {
1740   \exp_after:wN \tl_const:Nn \exp_after:wN \c__ctex_engine_file_tl
1741   \exp_after:wN { \g_file_current_name_tl }
1742 }

```

(End definition for \ctex_detected_platform:.)

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

```

1743 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1744 {
1745   \file_if_exist:nTF { \c__ctex_macosx_file_tl }
1746   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1747   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }

```

```

1748 }
1749 \tl_const:Nn \c__ctex_macosx_file_tl { /Library/Fonts/Songti.ttc }
1750 </xetex|pdfex>

```

(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。

```

1751 <(*class|ctex)>
1752 \@ifpackageloaded { hyperref }
1753 {
1754   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1755     { \hypersetup {#1} }
1756 }
1757 {
1758   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1759     { \PassOptionsToPackage {#1} { hyperref } }
1760 }
1761 </class|ctex>

```

(End definition for \ctex_hypersetup:n.)

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

```

1762 <(*pdfex)>
1763 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1764 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1765 {
1766   \ctex_hypersetup:n { CJKbookmarks = true }
1767   \ctex_if_pdfmode:TF
1768     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1769     {
1770       \ctex_at_end_package:nn { hyperref }
1771       {
1772         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1773         {
1774           \AtBeginShipoutFirst
1775             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1776         }
1777         { \RequirePackage { xCJK2uni } }
1778       }
1779     }
1780 }
1781 { \ctex_hypersetup:n { pdfencoding = unicode } }
1782 </pdfex>

```

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%

```

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

```

\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...
```

X_YTeX 的维护者 Khaled Hosny 已经注意到了这个问题¹⁷。需要注意的是, `hxeTeX.def` 重载了宏包选项 `unicode`, 目的是不能设置它为 `false`, 但也导致它不会改变 `\HyPsd@pdfencoding`。如果 `hyperref` 先于 `CTEX` 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 X_YTeX 下更保险的 `pdfencoding` 选项来设置。为了与 X_YTeX 下的行为一致 (使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 LuaTeX 下也启用这个选项。

```

1783 <*xetex|luatex>
1784 \ctex_hypersetup:n { pdfencoding = unicode }
1785 </xetex|luatex>

1786 <*pdfTeX|xetex|luatex>

```

15.4.7 CJKfntef、xeCJKfntef 相关设置

CT_EX 宏集对 pdfTeX 与 X_YTeX 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。

```

1787 <*pdfTeX>
1788 \RequirePackage { CJKfntef }
1789 \normalem
1790 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1791 { \tl_clear:c { CJK#1color } }
1792 </pdfTeX>
1793 <*xetex>
1794 \RequirePackage { xeCJKfntef }
1795 \ifpackageafter { xeCJKfntef } { 2014/11/04 }
1796 {
1797
1798 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1799 { \xeCJKsetup { #1 / format = { } } }
1800 }
1801 {
1802 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1803 { \tl_clear:c { CJK#1color } }
1804 }

```

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

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

```

1805 </xetex>
1806 <*luatex>
1807 \msg_new:nnn { ctex } { fntef-not-available }
1808 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1809 \msg_warning:nn { ctex } { fntef-not-available }
1810 </luatex>
1811 \clist_map_inline:nn
1812 { underdot , underline , underdblline , underwave , sout , xout }
1813 <*pdfTeXxetex>
1814 {
1815   \__ctex_clear_fntef_color:n {#1}
1816   \cs_new_protected_nopar:cpx { CTEX#1 }
1817   {
1818     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1819     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1820     \exp_not:c { CJK#1 }
1821   }
1822 }
1823 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1824 {
1825   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1826   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1827   \CJKfilltwosides
1828 }
1829 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1830 </pdfTeXxetex>
1831 <*luatex>
1832 { \cs_new_eq:cN { CTEX#1 } \use:n }
1833 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1834 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1835 </luatex>
1836 <*pdfTeX>
1837 \clist_map_inline:nn
1838 {
1839   underdotbasesep , underdotsep , underlinebasesep ,
1840   underlinesep , underdbllinesep , underdbllinebasesep ,
1841   underwavebasesep , underwavesep , southeight ,
1842   underdotcolor , underwavecolor , underlinecolor ,
1843   underdbllinecolor , soutcolor , xoutcolor
1844 }
1845 {
1846   \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1847   \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1848 }
1849 </pdfTeX>

```

15.4.8 \ccwd 的更新

\ctex_update_ccwd:

```

\ccwd
1850 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1851 <*pdfTeXxetex>
1852 {
1853   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1854   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1855 }
1856 </pdfTeXxetex>
1857 <*luatex>
1858 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1859 </luatex>
1860 \dim_new:N \ccwd

```

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

\ctex_update_ccglue: 更新字间距。

```

1861 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1862   (*pdftex)
1863   {
1864     \cs_set_protected_nopar:Npn \CJKglue
1865     { \skip_horizontal:N \l__ctex_ccglue_skip }
1866   }
1867   (/pdftex)
1868   (*xetex)
1869   { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1870   (/xetex)
1871   (*luatex)
1872   { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1873   (/luatex)
1874   \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
1875 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1876   {
1877     (*pdftex|xetex)
1878     \if_meaning:w \CJKglue \l__ctex_ccglue:
1879     \prg_return_false: \else: \prg_return_true: \fi:
1880     (/pdftex|xetex)
1881     (*luatex)
1882     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1883     { \prg_return_false: } { \prg_return_true: }
1884     (/luatex)
1885   }

```

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

```

1886   (*pdftex|xetex)
1887   \ctex_at_end:n { \cs_new_eq:NN \l__ctex_ccglue: \CJKglue }
1888   (/pdftex|xetex)

```

(End definition for \ctex_if_ccglue_touched:TF.)

\ctex_update_em_unit: 将当前汉字的宽度保存到 \ccwd 中备用。不采用 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_T_EX 和 X_Y_T_EX 下, 直接使用 \f@size\p@ 作为汉字的宽度, 这应该对大多数汉字字体都成立, 但不适用于诸如“方正兰亭黑长”之类的特殊字体。在 X_Y_T_EX 可以用 \fontcharwd 来改进。而在 pdf_T_EX 下, 若使用 zhmetrics 技术, 所有的汉字共享同一个 .tfm, \fontcharwd 也就没有意义。在 Lua_T_EX 下, Lua_T_EX-ja 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 \zw 作为汉字宽度。

```

1889 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1890   (pdftex|xetex) { \dim_set:Nn \ccwd { \f@size \p@ } }
1891   (luatex) { \dim_set:Nn \ccwd { \zw } }

```

(End definition for \ctex_update_em_unit:.)

15.4.9 其它

\ctex_add_to_selectfont:n \EverySelectfont 直到文档开始时才有效。为了 \ccwd 和 Lua_T_EX-ja 的字体设置在导言区也可用, 我们还需要在这里手工修改 \selectfont。every_{sel} 宏包会用 \CheckCommand 来检查 \selectfont 是否为标准定义。我们修改了 \selectfont, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。

```

1892 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1893   {

```

```

1894 \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1895 { \exp_not:o { \CTEX@selectfont@hook #1 } }
1896 }
1897 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1898 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1899 \use:n
1900 {
1901   \ExplSyntaxOff
1902   \ctex_preto_cmd:NnTF \selectfont { \CTEX@selectfont@hook }
1903   {
1904     \tl_put_left:Nn \@EverySelectfont@Init
1905     { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1906   }
1907   { \ctex_patch_failure:N \selectfont }
1908 }
1909 \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。

```

1910 (*pdfTeX)
1911 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1912 /pdfTeX)
1913 (*xetex|luatex)
1914 \EverySelectfont { \CTEX@selectfont@hook }
1915 /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。

```

1916 (*luatex)
1917 \ctex_add_to_selectfont:n
1918 {
1919   \ctex_ltj_select_font:
1920   \ctex_ltj_select_alternate_font:
1921   \ctex_ltj_update_xkanjiskip:
1922 }
1923 \tl_set:Nn \CJK@family { song } \selectfont
1924 \tl_clear:N \CJK@family
1925 /luatex)

```

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

```

1926 \keys_define:nn { ctex }
1927 {
1928   (*pdfTeX|xetex)
1929   space .choice: ,
1930   space / true .code:n =
1931   { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
1932   { \xeCJKsetup { CJKspace = true } } ,
1933   space / auto .code:n =
1934   { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
1935   { \xeCJKsetup { CJKspace = false } } ,
1936   space / false .code:n =
1937   { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
1938   { \xeCJKsetup { CJKspace = false } } ,
1939   space .default:n = { true } ,
1940   space .initial:n = { auto }
1941   /pdfTeX|xetex)
1942   (*luatex)
1943   space .code:n =
1944   { \msg_warning:nn { ctex } { invalid-option } }
1945   /luatex)
1946 }

```

(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。

```

1947 \keys_define:nn { ctex }
1948 {
1949   punct .code:n =
1950   {
1951     \tl_set:Nx \l__ctex_punct_tl { #1 }
1952     <pdfTeX> \punctstyle { \l__ctex_punct_tl }
1953     <xetex> \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
1954     <luatex> \ctex_mono_jfm:o { \l__ctex_punct_tl }
1955   } ,
1956   punct .default:n = { quanjiao } ,
1957 }

```

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

```

1958 </pdfTeX|xetex|luatex>
1959 <*class|ctex>

```

15.4.10 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

1960 \pdfTeX_if_engine:TF
1961 {
1962   \tl_set:Nx \l__ctex_encoding_tl { \l__ctex_encoding_tl }
1963   \ctex_file_input:n { ctex-engine-pdfTeX.def }
1964 }
1965 {
1966   \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
1967   \xetex_if_engine:TF
1968   { \ctex_file_input:n { ctex-engine-xetex.def } }
1969   { \ctex_file_input:n { ctex-engine-luatex.def } }
1970 }

```

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。

```

1971 \keys_define:nn { ctex }
1972 {
1973   autoindent .choice: ,
1974   autoindent .default:n = { true } ,
1975   autoindent / true .code:n =
1976   {
1977     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
1978     \ctex_select_size:
1979   } ,
1980   autoindent / false .code:n =
1981   { \tl_clear:N \l__ctex_autoindent_tl } ,
1982   autoindent / unknown .code:n =
1983   {
1984     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
1985     \ctex_select_size:
1986   }
1987 }

```

(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。

```

1988 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
1989 \cs_new_protected_nopar:Npn \ctex_select_size:
1990 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

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

`\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。

```

1991 \cs_new_protected_nopar:Npn \ctex_update_size:
1992 {
1993   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
1994   {
1995     \ctex_update_stretch:
1996     \ctex_update_parindent:
1997   }
1998   { \ctex_update_ziju: }
1999 }
2000 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2001 \tl_new:N \l__ctex_ziju_tl
2002 \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。

```

2003 \ctex_add_to_selectfont:n
2004 { \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`。

```

2005 \keys_define:nn { ctex }
2006 {
2007   linestretch .code:n =
2008   {
2009     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2010     \ctex_select_size:
2011   } ,
2012   linestretch .value_required:
2013 }
2014 \tl_new:N \l__ctex_line_stretch_tl
2015 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

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

`\ctex_update_stretch:` 首先计算一行上汉字的字数, `\CJKglue` 相当于将 `\linewidth` 与汉字总宽度之差均匀地填充到汉字之间。 $\varepsilon\text{-TeX}$ 的除法是四舍五入, 而我们这里应该用截断。由于没有可展性的要求, 直接用原语 `\tex_divide:D` 要比 `\int_div_truncate:nn` 快一些。下面的算法还兼顾到了 `\linewidth` 不为汉字字宽的整数倍的情况。若用户禁用 `linestretch` 并且修改过 `\CJKglue`, 则只更新 `\ccwd`, 否则设置伸展量为 0.08 倍 `\baselineskip`。注意 `everysel` 的钩子位于 `\size@update` 之前, `\baselineskip` 还未更新, 不能直接使用它。

```

2016 \cs_new_protected_nopar:Npn \ctex_update_stretch:
2017 {
2018   \ctex_update_em_unit:
2019   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2020   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2021   {
2022     \ctex_if_ccglue_touched:TF
2023     { \ctex_update_ccwd: }
2024     {
2025       \dim_set:Nn \l__ctex_tmp_dim
2026       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2027       \skip_set:Nn \l__ctex_ccglue_skip
2028       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2029       \ctex_update_ccglue:

```

```

2030     }
2031   }
2032   {
2033     \int_set:Nn \l__ctex_tmp_int
2034     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2035     \tex_divide:D \l__ctex_tmp_int \ccwd
2036     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2037     {
2038       \skip_set:Nn \l__ctex_ccglue_skip
2039       {
2040         \c_zero_dim plus \dim_eval:n
2041         {
2042           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2043           \l__ctex_tmp_int
2044         }
2045       }
2046     }
2047     { \skip_zero:N \l__ctex_ccglue_skip }
2048     \ctex_update_ccglue:
2049   }
2050 }

```

(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。

```

2051 \cs_new_protected_nopar:Npn \ctex_update_parindent:
2052 {
2053   \tl_if_empty:NF \l__ctex_autoindent_tl
2054   {
2055     \dim_compare:nNnF \parindent = \c_zero_dim
2056     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2057   }
2058 }

```

(End definition for \ctex_update_parindent:.)

\ziju 若参数为 0, 则恢复正常间距。

```

2059 \NewDocumentCommand \ziju { m }
2060 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2061 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2062 {
2063   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2064   \ctex_select_size:
2065 }

```

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

\ctex_update_ziju: 更新字距。若字距不大于 -1, 即 \ccwd 为非正值, 则不计算伸缩值。否则, 首先假定汉字的宽度为正常宽度加上字距, 看一行上能正常放下多少个汉字。

```

2066 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2067 {
2068   \ctex_update_em_unit:
2069   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2070   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2071   \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。

```

2072   {
2073     \dim_set:Nn \l__ctex_tmp_dim
2074     { \linewidth - \ccwd + \l__ctex_ziju_dim }
2075     \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }

```

```

2076      \tex_divide:D \l__ctex_tmp_int \ccwd
2077      \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。

```

2078      \dim_compare:nNnF \parindent = \c_zero_dim
2079      {
2080        \int_compare:nNnF \l__ctex_tmp_int < \c_three
2081        { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2082      }
2083      \skip_set:Nn \l__ctex_ccglue_skip
2084      {
2085        \l__ctex_ziju_dim
2086        plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2087        minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2088        { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2089      }
2090    }
2091    { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2092    \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 $\text{T}_{\text{E}}\text{X}$ 是伸展还是收缩, 之前以“当前行是否还放得下一个汉字”为标准加上或减去伸缩值的做法也未必与实际结果一致, 所以只好还是设置为 2\ccwd 。

```

2093      \ctex_update_parindent:
2094    }
2095    \dim_new:N \l__ctex_ziju_dim

```

(End definition for `\ctex_update_ziju:`.)

`\CTEXindent` 过时命令。
`\CTEXnoindent`

```

2096 \NewDocumentCommand \CTEXindent { }
2097 {
2098   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2099   { \parindent is~ set~ to~ 2\ccwd. }
2100   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2101 }
2102 \NewDocumentCommand \CTEXnoindent { }
2103 {
2104   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2105   { \parindent is~ set~ to~ Opt. }
2106   \dim_zero:N \parindent
2107 }

```

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

15.6 中文数字与日期

```

2108 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2109 \RequirePackage { zhnumber }

```

`\chinese`

```

2110 \cs_new_eq:NN \chinese \zhnum
2111 \cs_new_eq:NN \Chinese \chinese
2112 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

`\CTEXnumber`

`\CTEXdigits`

```

2113 \NewDocumentCommand \CTEXnumber { m m }

```

```

2114 { \protected@edef #1 { \zhnumber {#2} } }
2115 \NewDocumentCommand \CTEXdigits { m m }
2116 { \protected@edef #1 { \zhdigits {#2} } }

```

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

today

```

2117 \cs_set_eq:NN \CTEX@todayold \today
2118 \keys_define:nn { ctex }
2119 {
2120   today .choice: ,
2121   today / old .code:n =
2122     { \cs_set_eq:NN \today \CTEX@todayold } ,
2123   today / small .code:n =
2124     {
2125       \cs_set_eq:NN \today \zhtoday
2126       \zhnumsetup { time = Arabic }
2127     } ,
2128   today / big .code:n =
2129     {
2130       \cs_set_eq:NN \today \zhtoday
2131       \zhnumsetup { time = Chinese }
2132     } ,
2133   today / unknown .code:n =
2134     { \msg_error:nxx { ctex } { today-undef } {#1} }
2135 }
2136 \msg_new:nnnn { ctex } { today-undef }
2137 { Today~format~`#1'~is~undefined. }
2138 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

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

15.7 其它中文标题定义

```

2139 \keys_define:nn { ctex }
2140 {
2141   contentsname .tl_set:N = \contentsname ,
2142   listfigurename .tl_set:N = \listfigurename ,
2143   listtablename .tl_set:N = \listtablename ,
2144   figurename .tl_set:N = \figurename ,
2145   tablename .tl_set:N = \tablename ,
2146   abstractname .tl_set:N = \abstractname ,
2147   indexname .tl_set:N = \indexname ,
2148   appendixname .tl_set:N = \appendixname ,
2149   <article> bibname .tl_set:N = \refname
2150   <book|report> bibname .tl_set:N = \bibname
2151 }
2152 <*ctex>
2153 \msg_new:nnn { ctex } { ctexbibname }
2154 {
2155   Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2156   The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2157 }
2158 \tl_if_exist:NTF \bibname
2159 { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2160 {
2161   \tl_if_exist:NTF \refname
2162   { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2163   {
2164     \msg_warning:nn { ctex } { ctexbibname }
2165     \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2166   }
2167 }
2168 </ctex>

```

2169 `</class|ctex>`

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。

2170 `<*class|heading>`

15.8.1 定义标题格式选项

`\c__ctex_headings_seq`

```

2171 \seq_new:N \c__ctex_headings_seq
2172 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2173 {
2174   <article> part , section , subsection , subsubsection ,
2175   <book|report> part , chapter , section , subsection , subsubsection ,
2176   paragraph , subparagraph
2177 }
```

(End definition for \c__ctex_headings_seq.)

`__ctex_initial_heading:n`

```

2178 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2179 {
2180   \tl_new:c { CTEX@pre#1 }
2181   \tl_new:c { CTEX@post#1 }
2182   \tl_const:cx { CTEX@the#1 }
2183   {
2184     \exp_not:c { CTEX@pre#1 }
2185     \exp_not:c { CTEX@the#1 }
2186     \exp_not:c { CTEX@post#1 }
2187   }
2188   \tl_const:cx { CTEX@#1name }
2189   {
2190     \exp_not:c { CTEX@#1@nameformat }
2191     {
2192       \exp_not:c { CTEX@pre#1 }
2193       \exp_not:N \tl_if_empty:NTF \exp_not:c { CTEX@#1@numberformat }
2194       { \exp_not:c { CTEX@the#1 } }
2195       {
2196         \group_begin:
2197         \exp_not:c { CTEX@#1@numberformat }
2198         \exp_not:c { CTEX@the#1 }
2199         \group_end:
2200       }
2201       \exp_not:c { CTEX@post#1 }
2202     }
2203     \exp_not:c { CTEX@#1@aftername }
2204   }
2205 }
```

(End definition for __ctex_initial_heading:n.)

`__ctex_def_heading_keys:n`

```

2206 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2207 {
2208   \tl_put_right:Nx \l__ctex_tmp_tl
2209   {
2210     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2211     #1 / name .code:n =
2212     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2213     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
```

```

2214      #1 / format      .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2215      #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2216      #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2217      #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2218      #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2219      #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2220      #1 / before skip .tl_set:N = \exp_not:c { CTEX@#1@before skip } ,
2221      #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2222      #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2223      #1 / format+ .code:n =
2224      { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2225      #1 / format~+ .code:n =
2226      { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2227      #1 / nameformat+ .code:n =
2228      { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2229      #1 / nameformat~+ .code:n =
2230      { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2231      #1 / numberformat+ .code:n =
2232      { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2233      #1 / numberformat~+ .code:n =
2234      { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2235      #1 / aftername+ .code:n =
2236      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2237      #1 / aftername~+ .code:n =
2238      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2239      #1 / titleformat+ .code:n =
2240      { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2241      #1 / titleformat~+ .code:n =
2242      { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2243      #1 / aftertitle+ .code:n =
2244      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2245      #1 / aftertitle~+ .code:n =
2246      { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2247      #1 / before skip .initial:n = \c_zero_skip ,
2248      #1 / afterskip .initial:n = \c_zero_skip ,
2249      #1 / indent .initial:n = \c_zero_dim ,
2250      #1 / before skip .value_required: ,
2251      #1 / afterskip .value_required: ,
2252      #1 / indent .value_required: ,
2253    }
2254  }

```

(End definition for __ctex_def_heading_keys:n.)

`ctex_assign_heading_name:nn` `name` 的值是一个至多两个元素的逗号分隔列表。由于 \LaTeX 3 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。

```

2255 \NewDocumentCommand \ctex_assign_heading_name:nn
2256 { m > { \SplitArgument { \c_one } { , } } +m }
2257 { \__ctex_assign_heading_name:nnn {#1} #2 }
2258 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2259 {
2260   \tl_set:cn { CTEX@pre#1 } {#2}
2261   \IfNoValueTF {#3}
2262   { \tl_clear:c { CTEX@post#1 } }
2263   { \tl_set:cn { CTEX@post#1 } {#3} }
2264 }

```

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

`part/pagestyle` 只在 `ctexbook` 和 `ctexrep` 下有定义。
`chapter/pagestyle`

```

2265 (*book|report)
2266 \tl_set:Nn \l__ctex_tmp_tl

```

```

2267 {
2268   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2269   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2270 }
2271 </book|report>

```

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

定义标题键值选项。

```

2272 \seq_map_inline:Nn \c__ctex_headings_seq
2273 {
2274   \__ctex_initial_heading:n {#1}
2275   \__ctex_def_heading_keys:n {#1}
2276 }
2277 \use:x { \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } } }
2278 \tl_clear:N \l__ctex_tmp_tl

```

15.8.2 标准标题命令的修改

15.8.2.1 part 的标题

```

2279 <@@=>

```

\part

```

2280 <*article>
2281 \renewcommand\part{%
2282   \if@noskipsec \leavevmode \fi
2283   \par
2284   % \addvspace{4ex}%
2285   \@tempskipa \CTEX@part@beforeskip \relax
2286   \ifdim \@tempskipa <\z@
2287     \@tempskipa -\@tempskipa \@afterindentfalse
2288   \else
2289     \@afterindenttrue
2290   \fi
2291   \addvspace{\@tempskipa}%
2292   \secdef\@part\@spart}
2293 </article>

```

(End definition for \part.)

\@part

```

2294 <*article>
2295 \def\@part[#1]#2{%
2296   \ifnum \c@secnumdepth >\m@ne
2297     \refstepcounter{part}%
2298     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2299     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2300   \else
2301     \addcontentsline{toc}{part}{#1}%
2302   \fi
2303   {\interlinepenalty \@M
2304   % \normalfont \parindent \z@ \raggedright
2305   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2306   \ifnum \c@secnumdepth >\m@ne
2307     % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2308     \CTEX@partname
2309   \fi
2310   % \huge\bfseries #2%
2311   \CTEX@part@titleformat{#2}%
2312   \markboth{}{}%
2313   \CTEX@part@aftertitle}%
2314   \nobreak
2315   % \vskip 3ex

```

```

2316 \vskip \CTEX@part@afterskip
2317 \@afterheading}
2318 </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。

```

2319 <*book|report>
2320 \def\@part[#1]#2{%
2321   \thispagestyle{\CTEX@part@pagestyle}%
2322   \ifnum \c@secnumdepth >-2\relax
2323     \refstepcounter{part}%
2324     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2325     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2326   \else
2327     \addcontentsline{toc}{part}{#1}%
2328   \fi
2329   \markboth{}{}%
2330   {\interlinepenalty \@M
2331   % \normalfont \centering
2332   \normalfont \CTEX@part@format
2333   \ifnum \c@secnumdepth >-2\relax
2334     % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2335     \CTEX@partname
2336   \fi
2337   % \Huge\bfseries #2\par}%
2338   \CTEX@part@titleformat{#2}%
2339   \CTEX@part@aftertitle}%
2340   \@endpart}
2341 </book|report>

```

(End definition for `\@part`.)

`\@spart`

```

2342 <*article>
2343 \def\@spart#1{%
2344   {\interlinepenalty \@M
2345   % \normalfont \parindent \z@ \raggedright
2346   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2347   % \huge \bfseries #1\par}%
2348   \CTEX@part@titleformat{#1}%
2349   \CTEX@part@aftertitle}%
2350   \nobreak
2351   % \vskip 3ex
2352   \vskip \CTEX@part@afterskip
2353   \@afterheading}
2354 </article>
2355 <*book|report>
2356 \def\@spart#1{%
2357   {\interlinepenalty \@M
2358   % \normalfont \centering
2359   \normalfont \CTEX@part@format
2360   % \Huge \bfseries #1\par}%
2361   \CTEX@part@titleformat{#1}%
2362   \CTEX@part@aftertitle}%
2363   \@endpart}
2364 </book|report>

```

(End definition for `\@spart`.)

15.8.2.2 chapter 的标题

```

2365 <*book|report>

```

`\@chapter`

```

2366 \def\@chapter[#1]#2{%
2367   \ifnum \c@secnumdepth >\m@ne
2368   (book)   \if@mainmatter
2369             \refstepcounter{chapter}%
2370             % \typeout{\@chapapp\space\thechapter.}%
2371             \typeout{\CTEXthechapter}%
2372             \addcontentsline{toc}{chapter}
2373             % {\protect\numberline{\thechapter}#1}%
2374             {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2375   (book)   \else
2376   (book)   \addcontentsline{toc}{chapter}{#1}%
2377   (book)   \fi
2378   \else
2379     \addcontentsline{toc}{chapter}{#1}%
2380   \fi
2381   \chaptermark{#1}%
2382   \addtocontents{lof}{\protect\addvspace{10\p@}}%
2383   \addtocontents{lot}{\protect\addvspace{10\p@}}%
2384   \if@twocolumn
2385     \topnewpage[\@makechapterhead{#2}]%
2386   \else
2387     \@makechapterhead{#2}%
2388   \@afterheading
2389   \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2390 \def\@makechapterhead#1{%
2391   \thispagestyle{\CTEX@chapter@pagestyle}%
2392   % \vspace*{50\p@}%
2393   \@tempskipa \CTEX@chapter@beforeskip \relax
2394   \ifdim \@tempskipa <\z@
2395     \@tempskipa -\@tempskipa \@afterindentfalse
2396   \else
2397     \@afterindenttrue
2398   \fi
2399   \vspace*{\@tempskipa}%
2400   % {\normalfont \parindent \z@ \raggedright
2401   {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2402     \ifnum \c@secnumdepth >\m@ne
2403     (book)   \if@mainmatter
2404     %       \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2405     \CTEX@chaptername
2406   (book)   \fi
2407   \fi
2408   \interlinepenalty\@M
2409   % \Huge \bfseries #1\par\nobreak
2410   \CTEX@chapter@titleformat{#1}%
2411   \CTEX@chapter@aftertitle
2412   \nobreak
2413   % \vskip 40\p@
2414   \vskip \CTEX@chapter@afterskip
2415   }}

```

(End definition for \@makechapterhead.)

\@makeschapterhead

```

2416 \def\@makeschapterhead#1{%
2417   \thispagestyle{\CTEX@chapter@pagestyle}%
2418   % \vspace*{50\p@}%
2419   \@tempskipa \CTEX@chapter@beforeskip \relax
2420   \ifdim \@tempskipa <\z@
2421     \@tempskipa -\@tempskipa \@afterindentfalse
2422   \else

```

```

2423     \@afterindenttrue
2424     \fi
2425     \vspace*{\@tempskipa}%
2426 % {\normalfont \parindent \z@ \raggedright
2427 {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2428 \interlinepenalty\@M
2429 % \Huge \bfseries #1\par\nobreak
2430 \CTEX@chapter@titleformat{#1}
2431 \CTEX@chapter@aftertitle
2432 \nobreak
2433 % \vskip 40\p@
2434 \vskip \CTEX@chapter@afterskip
2435 }}

```

(End definition for \@makeschapterhead.)

```
2436 </book|report>
```

15.8.2.3 section 类的标题

\@secntformat

```

2437 \def\@secntformat#1{%
2438   \@ifundefined{CTEX@#1name}%
2439   {\csname the#1\endcsname\quad}%
2440   {\csname CTEX@#1name\endcsname}}

```

(End definition for \@secntformat.)

\@sect

```

2441 \def\@sect#1#2#3#4#5#6[#7]#8{%
2442   \ifnum #2>\c@secnumdepth
2443     \let\@svsec\@empty
2444   \else
2445     \refstepcounter{#1}%
2446     \protected@edef\@svsec{\@secntformat{#1}\relax}%
2447   \fi
2448   \@tempskipa #5\relax
2449   \ifdim \@tempskipa>\z@
2450     \begingroup
2451       #6{%
2452         \@hangfrom{\hskip #3\relax\@svsec}%
2453 %       \interlinepenalty \@M #8\@@par}%
2454       \interlinepenalty \@M
2455       \csname CTEX@#1@titleformat\endcsname{#8}%
2456       \csname CTEX@#1@aftertitle\endcsname}%
2457     \endgroup
2458     \csname #1mark\endcsname{#7}%
2459     \addcontentsline{toc}{#1}{%
2460       \ifnum #2>\c@secnumdepth \else
2461 %       \protect\numberline{\csname the#1\endcsname}%
2462       \protect\numberline{\@ifundefined{CTEXthe#1}%
2463         {\csname the#1\endcsname}%
2464         {\csname CTEXthe#1\endcsname}}}%
2465     \fi
2466     #7}%
2467   \else
2468     \def\@svsechd{%
2469       #6{\hskip #3\relax
2470 %       \@svsec #8}%
2471       \@svsec
2472       \csname CTEX@#1@titleformat\endcsname{#8}%
2473       \csname CTEX@#1@aftertitle\endcsname}%
2474     \csname #1mark\endcsname{#7}%
2475     \addcontentsline{toc}{#1}{%

```

```

2476 \ifnum #2>\c@secnumdepth \else
2477 % \protect\numberline{\csname the#1\endcsname}%
2478 \protect\numberline{\@ifundefined{CTEXthe#1}%
2479 \csname the#1\endcsname}%
2480 \csname CTEXthe#1\endcsname}}%
2481 \fi
2482 #7}}%
2483 \fi
2484 \@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`。

```

2485 \def\@ssect#1#2#3#4#5{%
2486 \@tempskipa #3\relax
2487 \ifdim \@tempskipa>\z@
2488 \begingroup
2489 #4{%
2490 \hangfrom{\hskip #1}%
2491 % \interlinepenalty \@M #5\@par}%
2492 \interlinepenalty \@M
2493 \CTEX@titleformat@n{#5}%
2494 \CTEX@aftertitle}%
2495 \endgroup
2496 \else
2497 % \def\@svsechd{#4{\hskip #1\relax #5}}%
2498 \def\@svsechd{#4{\hskip #1\relax \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2499 \fi
2500 \@xsect{#3}}

```

(End definition for \@ssect.)

2501 `<@@=ctex>`

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

```

2502 \cs_new_protected_nopar:Npn \CTEX@set@titleformat@n #1
2503 {
2504 \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2505 \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2506 }
2507 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2508 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:

```

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

```

2509 \int_zero:N \l__ctex_tmp_int
2510 \clist_map_inline:nn
2511 { section , subsection , subsubsection , paragraph , subparagraph }
2512 {
2513 \int_incr:N \l__ctex_tmp_int
2514 \cs_gset_protected_nopar:cpx {#1}
2515 {
2516 \exp_not:N \@startsection {#1}
2517 { \int_use:N \l__ctex_tmp_int }
2518 { \exp_not:c { CTEX@#1@indent } }
2519 { \exp_not:c { CTEX@#1@beforeskip } }
2520 { \exp_not:c { CTEX@#1@afterskip } }
2521 {
2522 \CTEX@set@titleformat@n {#1}
2523 \exp_not:N \normalfont \exp_not:c { CTEX@#1@format }
2524 }

```

```

2525     }
2526 }

```

15.8.2.4 附录标题

appendix/name
appendix/number

```

2527 \keys_define:nn { ctex }
2528 {
2529     appendix .meta:nn = { ctex / appendix } {#1} ,
2530     appendix / name .code:n =
2531         { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2532     appendix / number .tl_set:N = \CTEX@appendixnumber
2533 }
2534 \tl_new:N \CTEX@preappendix
2535 \tl_new:N \CTEX@postappendix

```

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

\appendix

```

2536 \cs_new_eq:NN \CTEX@save@appendix \appendix
2537 \cs_gset_protected_nopar:Npn \appendix
2538 {
2539     \CTEX@save@appendix
2540     (*article)
2541     \gdef \CTEX@presection { \CTEX@preappendix }
2542     \gdef \CTEX@thesection { \CTEX@appendixnumber }
2543     \gdef \CTEX@postsection { \CTEX@postappendix }
2544     /article)
2545     (*book|report)
2546     \gdef \CTEX@prechapter { \CTEX@preappendix }
2547     \gdef \CTEX@thechapter { \CTEX@appendixnumber }
2548     \gdef \CTEX@postchapter { \CTEX@postappendix }
2549     /book|report)
2550 }

```

(End definition for \appendix.)

15.8.3 目录标签的宽度

\numberline

```

2551 \cs_new_protected:Npn \CTEX@toc@width@n #1
2552 {
2553     \hbox_set:Nn \l__ctex_tmp_box {#1}
2554     \dim_set:Nn \@tempdima
2555     {
2556         \dim_max:nn { \@tempdima }
2557         { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2558     }
2559 }
2560 \group_begin:
2561 \char_set_catcode_other:N \#
2562 \use:n
2563 {
2564     \group_end:
2565     \ExplSyntaxOff
2566     \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2567     { \ctex_patch_failure:N \numberline }
2568     \@ifpackageloaded { tocloft }
2569     { }
2570     {
2571         \ctex_at_end_package:nn { tocloft }
2572     }

```

```

2573         \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2574         { \ctex_patch_failure:N \numberline }
2575     }
2576 }
2577 \ExplSyntaxOn
2578 }

```

(End definition for \numberline.)

15.8.4 页眉信息的修改

\ps@headings

```

2579 (*article)
2580 \ctex_patch_cmd:Nnn \ps@headings { \thesection } { \CTEXthesection }
2581 \if@twoside
2582   \ctex_patch_cmd:Nnn \ps@headings { \thesubsection } { \CTEXthesubsection }
2583 \fi:
2584 (/article)
2585 (*book|report)
2586 \ctex_patch_cmd:Nnn \ps@headings
2587   { \@chapapp\ \thechapter.\~\ } { \CTEXthechapter \quad }
2588 \if@twoside
2589   \ctex_patch_cmd:Nnn \ps@headings { \thesection.\~\ } { \CTEXthesection \quad }
2590 \fi:
2591 (/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。

```

2592 \ctex_at_end_package:nn { fancyhdr }
2593 {
2594   (*article)
2595     \ctex_patch_cmd:Nnn \ps@fancy
2596       { \thesection \hspace{1em} \relax } { \CTEXthesection \quad }
2597     \ctex_patch_cmd:Nnn \ps@fancy
2598       { \thesubsection \hspace{1em} \relax } { \CTEXthesubsection \quad }
2599   (/article)
2600   (*book|report)
2601     \ctex_patch_cmd:Nnn \ps@fancy
2602       { \@chapapp\ \thechapter.\~\ } { \if@mainmatter \CTEXthechapter \quad \fi }
2603     \ctex_patch_cmd:Nnn \ps@fancy
2604       { \@chapapp\ \thechapter.\~\ } { \CTEXthechapter \quad }
2605     \ctex_patch_cmd:Nnn \ps@fancy
2606       { \thesection.\~\ } { \CTEXthesection \quad }
2607   (/book|report)
2608 }

```

(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。

```

2608 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
2609 {
2610   \protected@edef \@currentlabel
2611   {
2612     \cs_if_exist:CTF { CTEX@the#1 }
2613     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
2614     { \exp_not:o { \@currentlabel } }
2615   }
2616 }

```

(End definition for \refstepcounter.)

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

```
2617 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
2618 {
2619   \seq_map_inline:Nn \c__ctex_headings_seq
2620   { \ctex_fix_varioref_label:n { ##1 } }
2621 }
```

(End definition for \ctex_varioref_hook:.)

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

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

它的第二个参数必须放在花括号内,否则将会被作为宏的定界符号。

```
2622 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
2623 {
2624   \tl_if_empty:cT { p@#1 }
2625   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
2626 }
```

(End definition for __ctex_fix_varioref_label:n.)

如果 `varioref` 已经被载入,则使用它来设置。

```
2627 \@ifpackageloaded { varioref }
2628 { \ctex_varioref_hook: }
2629 {
2630   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
2631   \RenewDocumentCommand \refstepcounter { m }
2632   {
2633     \CTEX@save@refstepcounter {#1}
2634     \CTEX@setcurrentlabel@n {#1}
2635   }
2636   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
2637 }
```

15.8.6 载入 `<scheme>` 文件

```
2638 \ctex_scheme_input:o { \l__ctex_scheme_tl }
2639 </class|heading>
```

15.8.7 标题格式的 `scheme` 定义

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

```
2640 <*scheme&(article|book|report)>

2641 \keys_set:nn { ctex / part }
2642 {
2643   aftertitle = \par ,
2644   <*plain>
2645   name       = \partname \space ,
2646   number     = \thepart ,
```

```

2647 </plain>
2648 (*chinese)
2649     number      = \chinese { part } ,
2650 </chinese>
2651 (*article)
2652     nameformat   = \Large \bfseries ,
2653     afterskip    = 3ex ,
2654 (*plain)
2655     format       = \raggedright ,
2656     aftername    = \par \nobreak ,
2657     titleformat  = \huge \bfseries ,
2658     beforekip    = -4ex
2659 </plain>
2660 (*chinese)
2661     format       = \centering ,
2662     aftername    = \quad ,
2663     titleformat  = \Large \bfseries ,
2664     beforekip    = 4ex
2665 </chinese>
2666 </article>
2667 (*book|report)
2668     format       = \centering ,
2669     nameformat   = \huge \bfseries ,
2670     aftername    = \par \vskip 20 \p@ ,
2671     pagestyle    = plain ,
2672 (*plain)
2673     titleformat  = \Huge \bfseries
2674 </plain>
2675 (*chinese)
2676     titleformat  = \huge \bfseries
2677 </chinese>
2678 </book|report>
2679 }

2680 (*book|report)
2681 \keys_set:nn { ctex / chapter }
2682 {
2683     pagestyle    = plain ,
2684     nameformat   = \huge \bfseries ,
2685     aftertitle   = \par ,
2686     afterskip    = 40 \p@ ,
2687 (*plain)
2688     name         = \chaptername \space ,
2689     number       = \thechapter ,
2690     format       = \raggedright ,
2691     aftername    = \par \nobreak \vskip 20 \p@ ,
2692     titleformat  = \Huge \bfseries ,
2693     beforekip    = -50 \p@
2694 </plain>
2695 (*chinese)
2696     number      = \chinese { chapter } ,
2697     format       = \centering ,
2698     aftername    = \quad ,
2699     titleformat  = \huge \bfseries ,
2700     beforekip    = 50 \p@
2701 </chinese>
2702 }
2703 </book|report>

2704 (@@=)

2705 \keys_set:nn { ctex / section }
2706 {
2707     number       = \thesection ,
2708     aftername    = \quad ,
2709     aftertitle   = \@@par ,
2710     afterskip    = 2.3ex \@plus .2ex ,
2711 (*plain)

```

```

2712     format      = \Large \bfseries ,
2713     beforekip    = -3.5ex \@plus -1ex \@minus -.2ex
2714 \</plain>
2715 \<*/chinese>
2716     format      = \Large \bfseries \centering ,
2717     beforekip    = 3.5ex \@plus 1ex \@minus .2ex
2718 \</chinese>
2719 }

2720 \keys_set:nn { ctex / subsection }
2721 {
2722     number      = \thesubsection ,
2723     format      = \large \bfseries ,
2724     aftername    = \quad ,
2725     aftertitle   = \@@par ,
2726     afterskip    = 1.5ex \@plus .2ex ,
2727 \<*/plain>
2728     beforekip    = -3.25ex \@plus -1ex \@minus -.2ex
2729 \</plain>
2730 \<*/chinese>
2731     beforekip    = 3.25ex \@plus 1ex \@minus .2ex
2732 \</chinese>
2733 }

2734 \keys_set:nn { ctex / subsubsection }
2735 {
2736     number      = \thesubsubsection ,
2737     format      = \normalsize \bfseries ,
2738     aftername    = \quad ,
2739     aftertitle   = \@@par ,
2740     afterskip    = 1.5ex \@plus .2ex ,
2741 \<*/plain>
2742     beforekip    = -3.25ex \@plus -1ex \@minus -.2ex
2743 \</plain>
2744 \<*/chinese>
2745     beforekip    = 3.25ex \@plus 1ex \@minus .2ex
2746 \</chinese>
2747 }

2748 \keys_set:nn { ctex / paragraph }
2749 {
2750     number      = \theparagraph ,
2751     format      = \normalsize \bfseries ,
2752     aftername    = \quad
2753 }

2754 \keys_set:nn { ctex / subparagraph }
2755 {
2756     number      = \thesubparagraph ,
2757     format      = \normalsize \bfseries ,
2758     aftername    = \quad
2759 }

    处理 sub3section 与 sub4section 的格式。

2760 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2761 {
2762     \keys_set:nn { ctex / paragraph }
2763     {
2764         aftertitle = \@@par ,
2765 \<*/plain>
2766         beforekip  = -3.25ex \@plus -1ex \@minus -.2ex ,
2767 \</plain>
2768 \<*/chinese>
2769         beforekip  = 3.25ex \@plus 1ex \@minus .2ex ,
2770 \</chinese>
2771         afterskip   = 1ex \@plus .2ex
2772     }

```

```

2773 }
2774 {
2775   \keys_set:nn { ctex / paragraph }
2776   {
2777     beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2778     afterskip  = -1em
2779   }
2780 }
2781 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
2782 {
2783   \keys_set:nn { ctex / subparagraph }
2784   {
2785     aftertitle = \@@par ,
2786     (*plain)
2787     beforeskip = -3.25ex \@plus -1ex \@minus -.2ex ,
2788     (/plain)
2789     (*chinese)
2790     beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2791     (/chinese)
2792     afterskip  = 1ex \@plus .2ex
2793   }
2794 }
2795 {
2796   \keys_set:nn { ctex / subparagraph }
2797   {
2798     beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
2799     afterskip  = -1em
2800   }
2801 }
2802 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2803 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
2804 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
2805 (@@=ctex)

```

处理附录的格式。

```

2806 \keys_set:nn { ctex / appendix }
2807 (*article)
2808 { number = \@Alph \c@section }
2809 (/article)
2810 (*book|report)
2811 {
2812   name = \appendixname \space ,
2813   number = \@Alph \c@chapter
2814 }
2815 (/book|report)
2816 (/scheme&(article|book|report))

```

15.8.8 ctex.sty 的 heading 选项

```

2817 (*ctex|ctexheading)

```

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

```

2818 \clist_map_inline:nn { article , book , report }
2819 {
2820   \ifclassloaded {#1}
2821   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
2822   { }
2823 }

```

(End definition for `\c__ctex_std_class_tl`)

若标准文档类被引入，则载入对应的标题定义文件。否则视 `\chapter` 是否有定义来引入 `book` 或者 `article`。

```

2824 \msg_new:nnn { ctex } { not-standard-class }
2825 {
2826   None~of~the~standard~document~classes~was~loaded.\\
2827   Heading~`#1'~is~selected.\\
2828   ctex~may~not~work~as~expected.
2829 }
2830 <ctex>\bool_if:NTF \l__ctex_heading_bool
2831 <ctexheading>\use:n
2832 {
2833   \tl_if_exist:NTF \c__ctex_std_class_tl
2834   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
2835   {
2836     \cs_if_exist:NTF \chapter
2837     {
2838       \cs_if_exist:NF \if@mainmatter
2839       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
2840       \tl_const:Nn \c__ctex_class_tl { book }
2841     }
2842     { \tl_const:Nn \c__ctex_class_tl { article } }
2843     \msg_warning:nxx { ctex } { not-standard-class } { \c__ctex_class_tl }
2844   }
2845   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
2846 }
2847 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
2848 </ctex><ctexheading>

```

15.8.9 标题配置文件

```

2849 <*UTF8|GBK>
2850 \keys_set:nn { ctex }
2851 {
2852   contentsname = 目录 ,
2853   listfigurename = 插图 ,
2854   listtablename = 表格 ,
2855   figurename = 图 ,
2856   tablename = 表 ,
2857   abstractname = 摘要 ,
2858   indexname = 索引 ,
2859   bibname = 参考文献 ,
2860   appendixname = 附录
2861 }
2862 \keys_if_exist:nnT { ctex / part } { name }
2863 {
2864   \keys_set:nn { ctex / part } { name = { 第 , 部分 } }
2865   \keys_if_exist:nnT { ctex / chapter } { name }
2866   { \keys_set:nn { ctex / chapter } { name = { 第 , 章 } } }
2867 }
2868 </UTF8|GBK>

```

15.9 chinese 方案的其他设置

```

2869 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

2870 <article|book|report>\pagestyle { headings }

```

汉化标题名字。

```

2871 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
2872 { \ctex_file_input:n { ctex-name-gbk.cfg } }
2873 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

日期格式。

```

2874 \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。

```
2875 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
2876 { \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。

```
2877 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
2878 { \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。

```
2879 \ctex_if_autoindent_touched:F
2880 { \keys_set:nn { ctex } { autoindent = true } }
```

若 `ctex` 宏包与标准文档类及其衍生文档类联用, 则将载入 `indentfirst` 宏包, 实现章节标题后首个段落的段首缩进。

```
2881 <*generic>
2882 \tl_if_exist:NT \c__ctex_std_class_tl
2883 { \RequirePackage{ indentfirst } }
2884 </generic>
2885 </scheme&chinese>
```

15.10 中文字号

```
2886 <*class|ctex|ctexsize>
```

`\zihao`

```
2887 \NewDocumentCommand \zihao { m }
2888 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }
```

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

`\ctex_zihao:n`

```
2889 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
2890 {
2891   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
2892   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
2893   { \msg_error:nnn { ctex } { fontsize } {#1} }
2894 }
2895 \msg_new:nnnn { ctex } { fontsize }
2896 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
2897 {
2898   The~old~font~size~is~used~if~you~continue.\\
2899   The~available~font~sizes~are~listed~as~follow.\\
2900   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
2901 }
```

(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 -T_EX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确¹⁸。

```
2902 \prop_new:N \c__ctex_font_size_prop
2903 \seq_new:N \c__ctex_font_size_seq
2904 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
2905 {
2906   \use:x
2907   {
2908     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
```

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

```

2909         {
2910             { \dim_to_decimal:n {#2} }
2911             { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
2912         }
2913     }
2914     \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
2915 }
2916 \clist_map_inline:nn
2917 {
2918     { 8 } { 5 bp } ,
2919     { 7 } { 5.5 bp } ,
2920     { -6 } { 6.5 bp } ,
2921     { 6 } { 7.5 bp } ,
2922     { -5 } { 9 bp } ,
2923     { 5 } { 10.5 bp } ,
2924     { -4 } { 12 bp } ,
2925     { 4 } { 14 bp } ,
2926     { -3 } { 15 bp } ,
2927     { 3 } { 16 bp } ,
2928     { -2 } { 18 bp } ,
2929     { 2 } { 22 bp } ,
2930     { -1 } { 24 bp } ,
2931     { 1 } { 26 bp } ,
2932     { -0 } { 36 bp } ,
2933     { 0 } { 42 bp }
2934 }
2935 { \__ctex_save_font_size:nn #1 }

```

(End definition for \c__ctex_font_size_prop.)

\ctex_declare_math_sizes:nnnn

```

2936 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
2937 {
2938     \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
2939     \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
2940 }

```

(End definition for \ctex_declare_math_sizes:nnnn.)

__ctex_get_font_sizes:Nn

```

2941 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
2942 {
2943     \tl_clear:N #1
2944     \tl_map_inline:nn {#2}
2945     {
2946         \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
2947         { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
2948         { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
2949     }
2950 }

```

(End definition for __ctex_get_font_sizes:Nn.)

```

2951 \clist_map_inline:nn
2952 {
2953     { 8 } { 8 } { 5pt } { 5pt } ,
2954     { 7 } { 7 } { 5pt } { 5pt } ,
2955     { -6 } { -6 } { 5pt } { 5pt } ,
2956     { 6 } { 6 } { 5pt } { 5pt } ,
2957     { -5 } { -5 } { 6pt } { 5pt } ,
2958     { 5 } { 5 } { 7pt } { 5pt } ,
2959     { -4 } { -4 } { 8pt } { 6pt } ,
2960     { 4 } { 4 } { 5 } { 6 } ,
2961     { -3 } { -3 } { -4 } { -5 } ,
2962     { 3 } { 3 } { 4 } { 5 } ,

```

```

2963 { -2 }{ -2 }{ -3 }{ -4 } ,
2964 { 2 }{ 2 }{ 3 }{ 4 } ,
2965 { -1 }{ -1 }{ -2 }{ -3 } ,
2966 { 1 }{ 1 }{ 2 }{ 3 } ,
2967 { -0 }{ -0 }{ -1 }{ -2 } ,
2968 { 0 }{ 0 }{ 1 }{ 2 }
2969 }
2970 { \ctex_declare_math_sizes:nnnn #1 }

```

15.10.2 修改默认字号大小

`\ctex_set_font_size:Nnn`

```

2971 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
2972 {
2973   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
2974   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
2975   { \msg_error:nnn { ctex } { fontsize } {#2} }
2976 }
2977 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
2978 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

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

```

2979 \if_case:w \g__ctex_font_size_flag
2980   \ctex_file_input:n { ctex-c5size.clo }
2981 \or:
2982   \ctex_file_input:n { ctex-cs4size.clo }
2983 \fi:
2984 </class|ctex|ctexsize>
2985 <*c5size>
2986 \ctex_set_font_size:Nnn \normalsize { 5 }
2987 {
2988   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2989   \abovedisplayshortskip \z@ \@plus3\p@
2990   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2991   \belowdisplayskip \abovedisplayskip
2992   \let\@listi\@listI
2993 }
2994 \ctex_set_font_size:Nnn \small { -5 }
2995 {
2996   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
2997   \abovedisplayshortskip \z@ \@plus2\p@
2998   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
2999   \def\@listi{\leftmargin\leftmarginI
3000             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3001             \parsep 2\p@ \@plus\p@ \@minus\p@
3002             \itemsep \parsep}
3003   \belowdisplayskip \abovedisplayskip
3004 }
3005 \ctex_set_font_size:Nnn \footnotesize { 6 }
3006 {
3007   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3008   \abovedisplayshortskip \z@ \@plus\p@
3009   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3010   \def\@listi{\leftmargin\leftmarginI
3011             \topsep 3\p@ \@plus\p@ \@minus\p@
3012             \parsep 2\p@ \@plus\p@ \@minus\p@
3013             \itemsep \parsep}
3014   \belowdisplayskip \abovedisplayskip
3015 }
3016 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3017 \ctex_set_font_size:Nnn \tiny { 7 } { }
3018 \ctex_set_font_size:Nnn \large { -4 } { }
3019 \ctex_set_font_size:Nnn \Large { -3 } { }

```

```

3020 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3021 \ctex_set_font_size:Nnn \huge { 2 } { }
3022 \ctex_set_font_size:Nnn \Huge { 1 } { }
3023 </cs4size>
3024 <*cs4size>
3025 \ctex_set_font_size:Nnn \normalsize { -4 }
3026 {
3027   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3028   \abovedisplayshortskip \z@ \@plus3\p@
3029   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3030   \belowdisplayskip \abovedisplayskip
3031   \let\@listi\@listI
3032 }
3033 \ctex_set_font_size:Nnn \small { 5 }
3034 {
3035   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3036   \abovedisplayshortskip \z@ \@plus3\p@
3037   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3038   \def\@listi{\leftmargin\leftmarginI
3039             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3040             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3041             \itemsep \parsep}
3042   \belowdisplayskip \abovedisplayskip
3043 }
3044 \ctex_set_font_size:Nnn \footnotesize { -5 }
3045 {
3046   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3047   \abovedisplayshortskip \z@ \@plus3\p@
3048   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3049   \def\@listi{\leftmargin\leftmarginI
3050             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3051             \parsep 3\p@ \@plus2\p@ \@minus\p@
3052             \itemsep \parsep}
3053   \belowdisplayskip \abovedisplayskip
3054 }
3055 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3056 \ctex_set_font_size:Nnn \tiny { -6 } { }
3057 \ctex_set_font_size:Nnn \large { -3 } { }
3058 \ctex_set_font_size:Nnn \Large { -2 } { }
3059 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3060 \ctex_set_font_size:Nnn \huge { -1 } { }
3061 \ctex_set_font_size:Nnn \Huge { 1 } { }
3062 </cs4size>
3063 <ctexsize>\normalsize
3064 <*classctex>

```

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`。

```

3065 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3066 {
3067   \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。

```

3068 \group_begin: \footnotesize \exp_args:NNNo \group_end:
3069 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3070 }

```

(End definition for `\footnotesep`.)

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

```
3071 \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。

```
3072 \NewDocumentCommand \CTeX { }
3073 { \ifincsname CTeX \else: C \TeX \fi: }
3074 \ctex_at_end_package:n { hyperref }
3075 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

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

`captiondelimenter` 过时选项。

```
3076 \keys_define:nn { ctex }
3077 {
3078   captiondelimenter .code:n =
3079   {
3080     \msg_warning:nnn { ctex } { deprecated-option }
3081     { You~can~load~the~package~`caption'~to~get~its~functionality. }
3082   }
3083 }
```

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

```
3084 </class>ctex>
```

15.12.1 列表环境的缩进

`\verse` 只在使用文档类的时候修改诗歌和引用环境的缩进。
`\quotation`

```
3085 <*<scheme!<generic!<chinese>
3086 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3087 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3088 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3089 </scheme!<generic!<chinese>
```

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

```
3090 <*<class>ctex>
```

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,我们不打算处理这种情况。

```
3091 \group_begin:
```

```

3092 \char_set_catcode_other:N \#
3093 \use:n
3094 {
3095   \group_end:
3096   \ExplSyntaxOff
3097   \ctex_at_end_package:nn { breqn }
3098   {
3099     \ctex_patch_cmd:NnnTF \end { \csname end#1 \endcsname }
3100     {
3101       \csuse { @end@#1@hook }
3102       \csname end#1 \endcsname
3103     } { }
3104     {
3105       \ctex_preto_cmd:NnTF \end { \csuse { @end@#1@hook } }
3106       { } { \ctex_patch_failure:N \end }
3107     }
3108   }
3109   \ExplSyntaxOn
3110 }

```

(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。

```

3111 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3112 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3113 \msg_new:nnn { ctex } { fontset-unavailable }
3114 { 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。

```

3115 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3116 {
3117   \tl_if_empty:NTF \g__ctex_fontset_tl
3118   {
3119     \ctex_detected_platform:
3120     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3121   }
3122   {
3123     \file_if_exist:NTF { ctex-fontset- \g__ctex_fontset_tl .def }
3124     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3125     {
3126       \use:x
3127       {
3128         \ctex_detected_platform:
3129         \msg_error:nxxx { ctex } { fontset-not-found }
3130         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3131       }
3132       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3133     }
3134   }
3135 }
3136 \@onlypreamble \ctex_load_fontset:
3137 \msg_new:nnnn { ctex } { fontset-not-found }
3138 {
3139   CTEX~fontset~`#1'~could~not~be~found.\\
3140   Fontset~`#2'~will~be~used~instead.
3141 }
3142 { 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。

```

3143 \keys_define:nn { ctex }
3144 {
3145   fontset .code:n =
3146   {
3147     \ctex_if_preamble:TF
3148     {
3149       \str_if_eq_x:nnTF {#1} { none }
3150       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3151       {
3152         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3153         {
3154           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3155           \ctex_load_fontset:
3156         }
3157         {
3158           \msg_error:nxxx { ctex } { fontset-loaded }
3159           { \g__ctex_fontset_tl } {#1}
3160         }
3161       }
3162     }
3163     { \msg_error:nn { ctex } { fontset-only-preamble } }
3164   }
3165 }
3166 \msg_new:nnnn { ctex } { fontset-loaded }
3167 {
3168   CTeX~fontset~`#1'~has~been~loaded.
3169   \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
3170 }
3171 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3172 \msg_new:nnn { ctex } { fontset-only-preamble }
3173 {
3174   The~`fontset'~option~can~be~used~only~in~preamble.
3175 }

```

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

载入中文字库。

```

3176 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3177 { \ctex_load_fontset: }

```

15.14 宏包配置文件

15.14.1 ctex.cfg

```

3178 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3179 </class>ctex>
3180 <*config>
3181 %%
3182 </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。

```

3183 <*ctexopts>
3184 %%
3185 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3186 </ctexopts>

```

15.15 字体定义文件

15.15.1 传统定义方式

```

3187 (*c19|c70)
3188 %%
3189 %% Chinese characters
3190 %%
3191 (c19)%% character set: GBK (extension of GB 2312)
3192 (c70)%% character set: Unicode
3193 %% font encoding: Unicode
3194 %%
3195 /c19|c70)

3196 (rm&c19)\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3197 (rm&c70)\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3198 (sf&c19)\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3199 (sf&c70)\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3200 (tt&c19)\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3201 (tt&c70)\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

3202 (*rm)
3203 (*c19)
3204 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3205 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3206 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3207 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3208 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3209 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3210 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3211 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3212 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3213 /c19)
3214 (*c70)
3215 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3216 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3217 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3218 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3219 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3220 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3221 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3222 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3223 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3224 /c70)
3225 /rm)

3226 (*sf)
3227 (*c19)
3228 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3229 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3230 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3231 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3232 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3233 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3234 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3235 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3236 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3237 /c19)
3238 (*c70)
3239 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
3240 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
3241 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
3242 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
3243 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3244 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3245 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
3246 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
3247 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
3248 /c70)

```

```

3249 </sf>
3250 <*tt>
3251 <*c19>
3252 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbksf}{\CJKnormal}
3253 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbksf}{\CJKbold}
3254 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbksf}{\CJKbold}
3255 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbksfsl}{\CJKnormal}
3256 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbksfsl}{\CJKbold}
3257 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbksfsl}{\CJKbold}
3258 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbksf}{\CJKnormal}
3259 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbksf}{\CJKbold}
3260 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbksf}{\CJKbold}
3261 </c19>
3262 <*c70>
3263 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifsf}{\CJKnormal}
3264 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifsf}{\CJKbold}
3265 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifsf}{\CJKbold}
3266 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifsssl}{\CJKnormal}
3267 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifsssl}{\CJKbold}
3268 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifsssl}{\CJKbold}
3269 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifsf}{\CJKnormal}
3270 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifsf}{\CJKbold}
3271 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifsf}{\CJKbold}
3272 </c70>
3273 </tt>
3274 <*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。

```

3275 <*windows>
3276 \file_if_exist:nTF { C:/boot.ini }
3277 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3278 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3279 </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。

```

3280 <*windowsnew|windowsold>
3281 \pdfTeX_if_engine:TF
3282 {
3283   \ctex_zhmap_case:nnn
3284   {
3285     \ctex_punct_set:n { windows }
3286     \setCJKmainfont
3287     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3288 <*windowsold>
3289     \setCJKsansfont { simhei.ttf }
3290     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3291 </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。

```

3292 <*windowsnew>
3293   \file_if_exist:nTF { C:/Windows/Fonts/msyh.ttc }
3294   {
3295     \setCJKsansfont [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3296     \setCJKfamilyfont { zhyahei }
3297     [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3298   }
3299   {
3300     \setCJKsansfont [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3301     \setCJKfamilyfont { zhyahei }

```

```

3302         [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3303     }
3304     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3305     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3306 </windowsnew>
3307     \setCJKmonofont { simfang.ttf }
3308     \setCJKfamilyfont { zhkai } { simkai.ttf }
3309     \setCJKfamilyfont { zhfs } { simfang.ttf }
3310     \setCJKfamilyfont { zhsong } { simsun.ttc }
3311     \setCJKfamilyfont { zhhei } { simhei.ttf }
3312     \setCJKfamilyfont { zhli } { simli.ttf }
3313     \setCJKfamilyfont { zhyou } { simyou.ttf }
3314     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3315     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3316     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3317     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3318 }
3319 {
3320     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3321     \ctex_punct_set:n { windows }
3322     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3323     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3324     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3325 }
3326 {
3327     \tl_set:Nn \CJKrmdefault { rm }
3328     \tl_set:Nn \CJKsfdefault { sf }
3329     \tl_set:Nn \CJKttdefault { tt }
3330 }
3331 }
3332 {
3333 <*windowsold>
3334     \setCJKmainfont
3335     [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3336     \setCJKsansfont { SimHei }
3337     \setCJKmonofont { FangSong_GB2312 }
3338     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3339     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3340 </windowsold>
3341 <*windowsnew>
3342     \setCJKmainfont
3343     [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3344     \setCJKsansfont
3345     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3346     \setCJKmonofont { FangSong }
3347     \setCJKfamilyfont { zhkai } { KaiTi }
3348     \setCJKfamilyfont { zhfs } { FangSong }
3349 </windowsnew>
3350     \setCJKfamilyfont { zhsong } { SimSun }
3351     \setCJKfamilyfont { zhhei } { SimHei }
3352     \setCJKfamilyfont { zhli } { LiSu }
3353     \setCJKfamilyfont { zhyou } { YouYuan }
3354 <*windowsnew>
3355     \setCJKfamilyfont { zhyahei }
3356     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3357 </windowsnew>
3358 }
3359 </windowsnew|windowsold>

```

15.15.3 ctex-fontset-adobe.def

```

3360 <*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。

```

3361 \pdfTEX_if_engine:TF
3362 {
3363   \ctex_if_pdfmode:TF
3364   { \ctex_fontset_error:n { adobe } }
3365   {
3366     \ctex_zhmap_case:nnn
3367     {
3368       \setCJKmainfont
3369       [
3370         cmap = UniGB-UTF16-H ,
3371         BoldFont = AdobeHeitiStd-Regular.otf ,
3372         ItalicFont = AdobeKaitiStd-Regular.otf
3373       ] { AdobeSongStd-Light.otf }
3374       \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3375       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3376       \setCJKfamilyfont { zhsong }
3377       [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3378       \setCJKfamilyfont { zhhei }
3379       [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3380       \setCJKfamilyfont { zhkai }
3381       [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3382       \setCJKfamilyfont { zhfs }
3383       [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3384       \ctex_punct_set:n { adobe }
3385       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3386       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3387       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3388       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3389       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3390     }
3391     {
3392       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3393       \ctex_punct_set:n { adobe }
3394       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3395       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3396       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3397     }
3398     { \ctex_fontset_error:n { adobe } }
3399   }
3400 }
3401 {
3402   \setCJKmainfont
3403   [
3404     BoldFont = AdobeHeitiStd-Regular ,
3405     ItalicFont = AdobeKaitiStd-Regular
3406   ] { AdobeSongStd-Light }
3407   \setCJKsansfont { AdobeHeitiStd-Regular }
3408   \setCJKmonofont { AdobeFangsongStd-Regular }
3409   \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3410   \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3411   \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3412   \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3413 }
3414 </adobe>
3415 <fandol>

```

15.15.4 ctex-fontset-fandol.def

```

3416 \pdfTEX_if_engine:TF
3417 {
3418   \ctex_if_pdfmode:TF
3419   { \ctex_fontset_error:n { fandol } }
3420   {
3421     \ctex_zhmap_case:nnn
3422     {
3423       \setCJKmainfont
3424       [

```

```

3425         cmap = UniGB-UTF16-H ,
3426         BoldFont = FandolSong-Bold.otf ,
3427         ItalicFont = FandolKai-Regular.otf
3428     ] { FandolSong-Regular.otf }
3429 \setCJKsansfont
3430 [
3431     cmap = UniGB-UTF16-H ,
3432     BoldFont = FandolHei-Bold.otf
3433 ] { FandolHei-Regular.otf }
3434 \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3435 \setCJKfamilyfont { zhsong }
3436 [
3437     cmap = UniGB-UTF16-H ,
3438     BoldFont = FandolSong-Bold.otf
3439 ] { FandolSong-Regular.otf }
3440 \setCJKfamilyfont { zhhei }
3441 [
3442     cmap = UniGB-UTF16-H ,
3443     BoldFont = FandolHei-Bold.otf
3444 ] { FandolHei-Regular.otf }
3445 \setCJKfamilyfont { zhfs }
3446 [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3447 \setCJKfamilyfont { zhkai }
3448 [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
3449 \ctex_punct_set:n { fandol }
3450 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3451 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3452 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3453 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3454 \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3455 \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
3456 }
3457 {
3458     \ctex_load_zhmap:n { rm } { zhhei } { zhfs } { zhfandolfonts }
3459     \ctex_punct_set:n { fandol }
3460     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3461     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3462     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3463 }
3464 { \ctex_fontset_error:n { fandol } }
3465 }
3466 }
3467 {
3468     \setCJKmainfont
3469     [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
3470     { FandolSong-Regular }
3471 \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3472 \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
3473 \setCJKfamilyfont { zhsong }
3474     [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
3475 \setCJKfamilyfont { zhhei }
3476     [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3477 \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
3478 \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
3479 }
3480 </fandol>
3481 <*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

```

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

```

/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...

```

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。

```

3482 \pdfTeX_if_engine:TF
3483 { \ctex_fontset_error:n { mac } }
3484 {
3485   \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
3486   \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
3487   \setCJKmonofont { STFangsong }
3488   \setCJKfamilyfont { zhsong } { STSong }
3489   \setCJKfamilyfont { zhhei } { STHeiti }
3490   \setCJKfamilyfont { zhfs } { STFangsong }
3491   \setCJKfamilyfont { zhkai } { STKaiti }
3492 }

```

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

```

3493 </mac>
3494 <*founder>

```

15.15.6 ctex-fontset-founder.def

```

3495 \pdfTeX_if_engine:TF
3496 {
3497   \ctex_zhmap_case:nnn
3498   {
3499     \setCJKmainfont
3500       [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
3501     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
3502     \setCJKmonofont { FZFSK.TTF }
3503     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
3504     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
3505     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
3506     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
3507     \setCJKfamilyfont { zhli } { FZLSK.TTF }
3508     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
3509     \ctex_punct_set:n { founder }
3510     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3511     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
3512     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3513     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3514     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3515     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
3516     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
3517   }
3518   {
3519     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
3520     \ctex_punct_set:n { founder }
3521     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3522     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3523     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3524   }
3525   { \ctex_fontset_error:n { founder } }
3526 }
3527 {
3528   \setCJKmainfont
3529     [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }

```

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

```

3530   \setCJKsansfont [ BoldFont = FZHei-B01 ]
3531   { \xetex_if_engine:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
3532   \setCJKmonofont { FZFangSong-Z02 }
3533   \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
3534   \setCJKfamilyfont { zhhei } { FZHei-B01 }
3535   \setCJKfamilyfont { zhkai } { FZKai-Z03 }
3536   \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
3537   \setCJKfamilyfont { zhli } { FZLiShu-S01 }
3538   \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
3539 }
3540 </founder>
3541 <*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。

```
3542 \pdfTeX_if_engine:TF
3543 {
3544   \ctex_zhmap_case:nnn
3545   {
3546     \setCJKmainfont
3547       [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
3548     \setCJKsansfont { wqy-zenhei.ttc }
3549     \setCJKmonofont { uming.ttc }
3550     \setCJKfamilyfont { zhsong } { uming.ttc }
3551     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
3552     \setCJKfamilyfont { zhkai } { ukai.ttc }
3553     \ctex_punct_set:n { ubuntu }
3554     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3555     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3556     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
3557     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3558     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3559   }
3560   {
3561     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
3562     \ctex_punct_set:n { ubuntu }
3563     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3564     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3565     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3566   }
3567   { \ctex_fontset_error:n { ubuntu } }
3568 }
3569 {
3570   \setCJKmainfont
3571     [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
3572   \setCJKsansfont { WenQuanYi~Zen~Hei }
3573   \setCJKmonofont { AR~PL~UMing~CN }
3574   \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
3575   \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
3576   \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
3577 }
3578 </ubuntu>
```

15.15.8 中文字体命令

```
3579 <!*windows>
3580 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
3581 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
3582 <ubuntu> \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
3583 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
3584 <*windowsnew|windowsold|founder>
3585 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
3586 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
3587 </windowsnew|windowsold|founder>
3588 <windowsnew> \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
3589 </!windows>
3590 </fontset>
3591 <*zhmap>
```

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

15.15.9 zhmetrics 的字体映射

确认 `\catcode`, 没有重复载入检查。

```

3592 \begingroup\catcode61\catcode48\catcode32=10\relax%
3593 \catcode 35=6 % #
3594 \catcode123=1 % {
3595 \catcode125=2 % }
3596 \toks0{\endlinechar=\the\endlinechar\relax}%
3597 \toks2{\endlinechar=13 }%
3598 \def\x#1 #2 {%
3599   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
3600   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
3601 \x 13 5 % carriage return
3602 \x 32 10 % space
3603 \x 35 6 % #
3604 \x 40 12 % (
3605 \x 41 12 % )
3606 \x 45 12 % -
3607 \x 46 12 % .
3608 \x 47 12 % /
3609 \x 58 12 % :
3610 \x 60 12 % <
3611 \x 61 12 % =
3612 \x 64 11 % @
3613 \x 91 12 % [
3614 \x 93 12 % ]
3615 \x 123 1 % {
3616 \x 125 2 % }
3617 \edef\x#1{\endgroup%
3618   \edef\noexpand#1{%
3619     \the\toks0 %
3620     \let\noexpand\noexpand\noexpand#1%
3621     \noexpand\noexpand\noexpand\undefined%
3622     \noexpand\noexpand\noexpand\endinput}%
3623     \the\toks2}%
3624 \expandafter\x\csname ctex@zhmap@endinput\endcsname

3625 \begingroup\expandafter\endgroup
3626 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
3627 \expandafter\ifx\csname ifctexpdf\endcsname\relax
3628   \expandafter\ifx\csname pdfoutput\endcsname\relax
3629     iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
3630   \else ifctexpdf\fi
3631 \endcsname

```

提供非 \LaTeX 格式下的 `\ProvidesFile`。

```

3632 \begingroup
3633 \expandafter\ifx\csname ProvidesFile\endcsname\relax
3634   \long\def\x#1\ProvidesFile#2[#3]{%
3635     #1%
3636     \immediate\write-1{File: #2 #3}%
3637     \expandafter\xdef\csname ver@#2\endcsname{#3}}
3638   \expandafter\x%
3639 \fi
3640 \endgroup

```

15.15.9.1 zhwindowsfonts.tex

```

3641 (*windows)
3642 \ProvidesFile{zhwindowsfonts.tex}%
3643 [2015/05/25 v2.1 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
3644
3645 \ifzhmappdf
3646   \pdfmapline{=gbk@UGBK@ <simsun.ttc}
3647   \pdfmapline{=gbksong@UGBK@ <simsun.ttc}

```

```

3648 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
3649 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
3650 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
3651 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
3652 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
3653
3654 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
3655 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
3656 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
3657 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
3658 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
3659 \pdfmapline{=unili@Unicode@ <simli.ttf}
3660 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
3661
3662 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
3663 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
3664 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
3665 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
3666 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
3667 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
3668
3669 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
3670 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
3671 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
3672 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
3673 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
3674 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
3675
3676 \else
3677 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
3678 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
3679 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
3680 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
3681 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
3682 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
3683 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
3684
3685 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
3686 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
3687 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
3688 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
3689 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
3690 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
3691 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
3692
3693 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
3694 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
3695 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
3696 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
3697 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
3698 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
3699
3700 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
3701 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
3702 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
3703 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
3704 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
3705 \special{pdf:mapline uniyouisl@Unicode@ unicode simyou.ttf -s .167 -v 60}
3706
3707 </windows>

```

15.15.9.2 zhadobefonts.tex

```

3708 <*adobe>
3709 \ProvidesFile{zhadobefonts.tex}%
3710 [2015/05/25 v2.1 Adobe font map loader for DVIPDFMx (CTEX)]
3711
3712 \ifzhmappdf

```

```

3713 %% pdfTeX does not support OTF fonts
3714
3715 \else
3716 \special{pdf:mapline gbk@UGBK@      UniGB-UTF16-H AdobeSongStd-Light.otf}
3717 \special{pdf:mapline gbksong@UGBK@  UniGB-UTF16-H AdobeSongStd-Light.otf}
3718 \special{pdf:mapline gbkkai@UGBK@   UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3719 \special{pdf:mapline gbkhei@UGBK@   UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3720 \special{pdf:mapline gbkfs@UGBK@    UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3721
3722 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3723 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3724 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3725 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3726 \special{pdf:mapline unifs@Unicode@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3727
3728 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3729 \special{pdf:mapline gbkkaisl@UGBK@  UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3730 \special{pdf:mapline gbkheisl@UGBK@   UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3731 \special{pdf:mapline gbkfssl@UGBK@    UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3732
3733 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3734 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3735 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3736 \special{pdf:mapline unifssl@Unicode@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3737
3738 </adobe>

```

15.15.9.3 zhfangdolfonts.tex

```

3739 <*fandol>
3740 \ProvidesFile{zhfangdolfonts.tex}%
3741 [2015/05/25 v2.1 Fandol font map loader for DVIPDFMx (CTEX)]
3742
3743 \ifzhmappdf
3744 %% pdfTeX does not support OTF fonts
3745
3746 \else
3747 \special{pdf:mapline gbk@UGBK@      UniGB-UTF16-H FandolSong-Regular.otf}
3748 \special{pdf:mapline gbksong@UGBK@  UniGB-UTF16-H FandolSong-Regular.otf}
3749 \special{pdf:mapline gbkkai@UGBK@   UniGB-UTF16-H FandolKai-Regular.otf}
3750 \special{pdf:mapline gbkhei@UGBK@   UniGB-UTF16-H FandolHei-Regular.otf}
3751 \special{pdf:mapline gbkfs@UGBK@    UniGB-UTF16-H FandolFang-Regular.otf}
3752
3753 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3754 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3755 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
3756 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
3757 \special{pdf:mapline unifs@Unicode@  UniGB-UTF16-H FandolFang-Regular.otf}
3758
3759 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3760 \special{pdf:mapline gbkkaisl@UGBK@  UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3761 \special{pdf:mapline gbkheisl@UGBK@   UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3762 \special{pdf:mapline gbkfssl@UGBK@    UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3763
3764 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3765 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3766 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3767 \special{pdf:mapline unifssl@Unicode@  UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3768
3769 </fandol>

```

15.15.9.4 zhfounderfonts.tex

```

3770 <*founder>
3771 \ProvidesFile{zhfounderfonts.tex}%
3772 [2015/05/25 v2.1 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
3773
3774 \ifzhmappdf

```

```

3775 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
3776 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
3777 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
3778 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
3779 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
3780 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
3781 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
3782
3783 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
3784 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
3785 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
3786 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
3787 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
3788 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
3789 \pdfmapline{=uniyoud@Unicode@ <FZY1K.TTF}
3790
3791 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
3792 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
3793 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
3794 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
3795 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
3796 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
3797
3798 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
3799 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
3800 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
3801 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
3802 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
3803 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
3804
3805 \else
3806 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
3807 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
3808 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
3809 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
3810 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
3811 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
3812 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
3813
3814 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
3815 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
3816 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
3817 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
3818 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
3819 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
3820 \special{pdf:mapline uniyoud@Unicode@ unicode FZY1K.TTF}
3821
3822 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
3823 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
3824 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
3825 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
3826 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
3827 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
3828
3829 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
3830 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
3831 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
3832 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
3833 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
3834 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
3835
3836 </founder>

```

15.15.9.5 zhubuntufonts.tex

```

3837 <*ubuntu>
3838 \ProvidesFile{zhubuntufonts.tex}%
3839 [2015/05/25 v2.1 Ubuntu font map loader for pdfTeX and DVIPDFx (CTEX)]

```

```

3840
3841 \ifzhmappdf
3842 \pdfmapline{=gbk@UGBK@ <uming.ttc}
3843 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
3844 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
3845 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
3846 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
3847 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
3848
3849 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
3850 \pdfmapline{=unisong@Unicode@ <uming.ttc}
3851 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
3852 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
3853 \pdfmapline{=unifs@Unicode@ <uming.ttc}
3854 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
3855
3856 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
3857 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
3858 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
3859 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
3860 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
3861
3862 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
3863 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
3864 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
3865 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
3866 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
3867
3868 \else
3869 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
3870 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
3871 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
3872 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
3873 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
3874
3875 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
3876 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
3877 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
3878 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
3879 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
3880
3881 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
3882 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
3883 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
3884 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
3885
3886 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
3887 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
3888 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
3889 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
3890
3891 </ubuntu>
3892 \fi
3893
3894 \ctex@zhmap@endinput
3895 </zhmap>

```

15.15.10 制作 spa 文件

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

```

3896 <*spa>
3897 <*macro>
3898 \input expl3-generic %
3899 \ExplSyntaxOn

```

```

3900 \xetex_if_engine:F
3901 {
3902   \msg_new:nn { ctex } { xetex }
3903   { XeTeX~is~required~to~compile~this~document! }
3904   \msg_fatal:nn { ctex } { xetex }
3905 }

```

CJKpunct 定义的标点符号是：

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

注意顺序不能改变。

```

3906 \seq_new:N \c__ctex_punct_seq
3907 \seq_gset_from_clist:Nn \c__ctex_punct_seq
3908 {
3909   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
3910   "3008 , "300A , "3016 , "3010 ,
3911   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
3912   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
3913   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
3914 }

```

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

```

3915 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
3916 {
3917   \iow_open:Nn \g__ctex_spa_iow {#1}
3918   \clist_map_inline:nn {#2}
3919   { \__ctex_write_family:nn ##1 }
3920   \iow_close:N \g__ctex_spa_iow
3921 }
3922 \iow_new:N \g__ctex_spa_iow
3923 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

```

(End definition for \ctex_make_spa:nn.)

```

3924 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
3925 {
3926   \group_begin:
3927   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
3928   \l__ctex_punct_font
3929   \clist_clear:N \l__ctex_punct_bounds_clist
3930   \seq_map_inline:Nn \c__ctex_punct_seq
3931   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \XeTeXcharglyph ##1 } }
3932   \iow_now:Nx \g__ctex_spa_iow
3933   {
3934     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

3935       { \l__ctex_punct_bounds_clist , , , }
3936     }
3937   \group_end:
3938 }
3939 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
3940 {
3941   \clist_put_right:Nx \l__ctex_punct_bounds_clist
3942   {
3943     \__ctex_calc_bounds:nn { \c_one } {#1} ,
3944     \__ctex_calc_bounds:nn { \c_three } {#1}
3945   }
3946 }
3947 \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。

```

3948 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
3949 {
3950   \fp_eval:n
3951   {
3952     round
3953     (
3954       \dim_to_decimal_in_unit:nn
3955       { 100 \XeTeXglyphbounds #1 ~ #2 }
3956       { 1 em }
3957     )
3958   }
3959 }
3960 \ExplSyntaxOff
3961 </macro>

```

下面是 C_TE_X 定义的一些字体。

```

3962 <*make>
3963 \input ctexspamacro %
3964
3965 \MAKESPA {ctexpunct.spa}
3966 {
3967   {adobezhsong}      {AdobeSongStd-Light} ,
3968   {adobezhhei}       {AdobeHeitiStd-Regular} ,
3969   {adobezhkai}       {AdobeKaitiStd-Regular} ,
3970   {adobezhfs}        {AdobeFangsongStd-Regular} ,
3971   {fandolzhsong}     {FandolSong} ,
3972   {fandolzhsongb}    {FandolSong-Bold} ,
3973   {fandolzhhei}      {FandolHei} ,
3974   {fandolzhheib}     {FandolHei-Bold} ,
3975   {fandolzhkai}      {FandolKai} ,
3976   {fandolzhfs}       {FandolFang} ,
3977   {founderzhsong}    {FZShuSong-Z01} ,
3978   {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
3979   {founderzhhei}     {FZHei-B01} ,
3980   {founderzhheil}    {FZXiHeiI-Z08} ,
3981   {founderzhkai}     {FZKai-Z03} ,
3982   {founderzhfs}      {FZFangSong-Z02} ,
3983   {founderzhli}      {FZLiShu-S01} ,
3984   {founderzhyou}     {FZXiYuan-M01} ,
3985   {founderzhyoub}    {FZZhunYuan-M02} ,
3986   {ubuntuzhsong}     {AR PL UMinG CN} ,
3987   {ubuntuzhhei}      {WenQuanYi Zen Hei} ,
3988   {ubuntuzhkai}      {AR PL UKai CN} ,
3989   {windowszhsong}    {SimSun} ,
3990   {windowszhhei}     {SimHei} ,
3991   {windowszhkai}     {KaiTi} ,
3992   {windowszhfs}      {FangSong} ,
3993   {windowszhli}      {LiSu} ,
3994   {windowszhyou}     {YouYuan} ,
3995   {windowszhyawei}    {Microsoft YaHei} ,
3996   {windowszhyaweib}  {Microsoft YaHei Bold}
3997 }
3998
3999 \primitive\end
4000 </make>
4001 </spa>

```

第 16 节 版本历史

v2.0

General: c5size, cs4size 是过时选项。	36
captiondelimenter 是过时选项。	97
fancyhdr 成为过时选项, 原选项功能总是打开。	40
fntef 成为过时选项, 原选项功能总是打开。	40
hyperref 成为过时选项, 原选项功能总是打开。	40
indent, noindent 是过时选项。	37
nofonts, adobefonts, winfonts 是过时选项。	37
nopunct 是过时选项。	38
nospace 是过时选项。	39
nozhmap 是过时选项。	38
punct 选项可以设置标点格式。	38
ctex 宏包新增 heading 选项。	39
\CTEXindent, \CTEXnoindent 是过时命令。	77
\CTEXsetup, \CTEXoptions 是过时命令。	14, 42
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; CTEXfilltwosides 是过时环境。	70
标题设置新增 pagestyle 选项。	80
调整 \footnotesep 的大小, 以适合行距的变化。	96
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。	40
将标题汉化功能加入 ctex.sty。	78
将中文字号功能提取到可以独立使用的 ctexsize。	93
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	70
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jaf 宏包支持 Lua \TeX 。	47
新增 autoindent 选项。	37
新增 fontset 选项。	37
新增 linespread 选项。	37
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。	93
自动检测操作系统, 载入对应的字体配置。	68
\CJK@surr: 解决与 \nouppercase 的冲突。	44
\end: 解决 etoolbox 与 breqn 关于 \end 的冲突。	97
v2.0.1	
General: 修复 10pt, 11pt 等选项无效的问题。	40
v2.0.2	
General: 修复加载 ctex 宏包后章节标题后第一段无段首缩进的问题。	93
v2.1	
General: format+, nameformat+ 等带加号的选项, 加号与前面的文字之间可以有可选的空格。	79
nameformat 可以接受章节名字为参数。	79
不依赖 ifpdf 宏包。	34
不再设置 hyperref 宏包的 colorlinks 选项。	69
给 pdf \LaTeX 下的非 UTF8 编码 CJK 字体族加上 CMap。	42
将章节标题设置功能提取到可以独立使用的宏包 ctex-heading 中。	1
新的标题格式选项 aftertitle。	79
修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。	88
\ctex_if_macosx:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	68
hyperref: 补充定义 \hypersetup。	40

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	
_	2587, 2589, 2602, 2603, 2605
\#	168, 584, 2561, 3092
.../aftername+	17
.../aftertitle+	18
.../format+	16
.../nameformat+	16
.../numberformat+	16
.../titleformat+	17
\/	112, 1005, 1368, 1369
\<	111
\ \	19, 37, 38, 112, 189, 198, 254, 256, 258, 765, 1163, 1245, 2155, 2826, 2827, 2898, 2899, 3139, 3169
\{	111, 1174, 1175
\}	1174, 1175
10pt	9, 40
11pt	9, 40
12pt	9, 40
A	
\A	113, 114
\abovedisplayshortskip	2989, 2997, 3008, 3028, 3036, 3047
\abovedisplayskip	2988, 2991, 2996, 3003, 3007, 3014, 3027, 3030, 3035, 3042, 3046, 3053
\abstractname	2146
abstractname	12
\addCJKfontfeature	1245, 1268
\addCJKfontfeatures	58, 1263, 1268
\addcontentsline	2298, 2299, 2301, 2324, 2325, 2327, 2372, 2376, 2379, 2459, 2475
\addtocontents	2382, 2383
\addvspace	2284, 2291, 2382, 2383

adobefonts	27	chapter/afterskip	19
\AfterEndPreamble	216	chapter/aftertitle	18
AlternateFont	24, 60	chapter/beforeskip	18
\appendix	86, 2536, 2537	chapter/format	16
appendix/name	20, 86	chapter/indent	19
appendix/number	20, 86	chapter/name	14
\appendixname	2148, 2812	chapter/nameformat	16
appendixname	12	chapter/number	15
\apptocmd	164	chapter/numberformat	16
\AtBeginDocument	668, 1663	chapter/pagestyle	20, 80
\AtBeginDvi	611	chapter/titleformat	17
\AtBeginShipoutFirst	613, 1774	\chaptermark	2381
\AtBeginUTFCommand	809	\chaptername	2688
\AtEndOfClass	244	char commands:	
\AtEndOfPackage	245	\char_set_catcode:nn	87, 641, 642
\AtEndPreamble	468, 679, 1695	\char_set_catcode_active:N	1368
\AtEndUTFCommand	815	\char_set_catcode_letter:n	81
autoindent	11, 37, 74	\char_set_catcode_math_superscript:n	636
		\char_set_catcode_other:N	
			114, 168, 230, 231, 584, 2561, 3092
		\char_set_catcode_other:n	635, 818
		\char_set_lccode:nn	111, 112, 113, 822, 1369
		\char_value_catcode:n	87, 641, 642
		CharRange	24, 60
		\Chinese	28, 2111
		\chinese	23, 77, 2110, 2111, 2649, 2696
		\CJK	1819
		\CJKbold	
			3211, 3212, 3222, 3223, 3229, 3230, 3232, 3233, 3235,
			3236, 3240, 3241, 3243, 3244, 3246, 3247, 3253, 3254,
			3256, 3257, 3259, 3260, 3264, 3265, 3267, 3268, 3270, 3271
		\CJKfamily	56, 621, 623, 625, 627, 1100, 1122, 1301, 1303,
			1305, 1307, 1453, 3580, 3581, 3582, 3583, 3585, 3586, 3588
		\CJKfamilydefault	616, 617, 627, 682, 1296, 1297, 1307,
			1314, 1316, 1320, 1327, 1340, 1675, 1679, 1687, 1701, 1703
		\CJKfilltwosides	1827
		\CJKfontspec	58, 1258
		\CJKglue	1853, 1864, 1878, 1887
		\CJKhook	46, 697
		\CJKnormal	3204, 3205, 3206, 3207, 3208, 3209, 3210,
			3215, 3216, 3217, 3218, 3219, 3220, 3221, 3228, 3231,
			3234, 3239, 3242, 3245, 3252, 3255, 3258, 3263, 3266, 3269
		\CJKpunctmapfamily	
			730, 731, 732, 733, 734, 735, 739, 740, 741, 742, 743, 744
		\CJKrmdefault	608, 617, 618, 621, 1170, 1181, 1271, 1297,
			1298, 1301, 1316, 1319, 1320, 1683, 3314, 3316, 3317,
			3322, 3323, 3324, 3327, 3385, 3388, 3389, 3394, 3395,
			3396, 3450, 3453, 3454, 3460, 3461, 3462, 3510, 3513,
			3514, 3521, 3522, 3523, 3554, 3557, 3558, 3563, 3564, 3565
		\CJKsfdefault	
			609, 619, 623, 1171, 1182, 1277, 1299, 1303, 1684,
			3290, 3304, 3305, 3328, 3386, 3451, 3455, 3511, 3515, 3555
		\CJKtilde	683
		\CJKttdefault	610, 620, 625, 1172, 1183,
			1282, 1300, 1305, 1685, 3315, 3329, 3387, 3452, 3512, 3556
		clearalternatefont	25, 64
B			
\baselinestretch	2026		
\begingroup	2450, 2488, 3592, 3625, 3632		
\belowdisplayshortskip	2990, 2998, 3009, 3029, 3037, 3048		
\belowdisplayskip	2991, 3003, 3014, 3030, 3042, 3053		
\bfdefault	1026, 1032, 1350, 1357		
\bfseries	2307, 2310, 2334, 2337, 2347,		
	2360, 2404, 2409, 2429, 2652, 2657, 2663, 2669, 2673,		
	2676, 2684, 2692, 2699, 2712, 2716, 2723, 2737, 2751, 2757		
\bibname	2150, 2155, 2158, 2159		
bibname	12		
bool commands:			
\bool_gset_false:N	374, 379		
\bool_gset_true:N	369		
\bool_if:NT	1203		
\bool_if:NTF	84, 2830		
\bool_if:nTF	120		
\bool_new:N	240, 1240		
\bool_set_false:N	1215		
\bool_set_true:N	1233		
box commands:			
\box_ht:N	3069		
\box_new:N	52		
\box_wd:N	1854, 2557		
C			
c5size	27		
cap	27		
captiondelimiter	28, 97		
\catcode	977, 3592, 3593, 3594, 3595, 3599, 3600		
CCT	27		
CCTfont	27		
\ccwd	21, 71, 222, 297, 1854, 1858, 1860,		
	1890, 1891, 1977, 2015, 2034, 2035, 2042, 2069, 2070,		
	2071, 2074, 2076, 2077, 2088, 2099, 2100, 3086, 3087, 3088		
\centering	2331, 2358, 2661, 2668, 2697, 2716		
\chapter	2836		
chapter/aftername	17		

clist commands:

`\clist_clear:N` 1047, 3929
`\clist_concat:NNN` 1069
`\clist_const:Nn` 710
`\clist_gput_right:Nn` 489, 511, 513
`\clist_gput_right:No` 501
`\clist_gset:Nn` 1288
`\clist_if_empty:NT` 1375, 1631
`\clist_if_empty:NTF` 1391
`\clist_map_break:n` 2821
`\clist_map_function:nN` 1582, 1596
`\clist_map_function:xN` 1548, 1550
`\clist_map_inline:Nn` 700, 1484
`\clist_map_inline:nn`
 . 474, 722, 1589, 1811, 1837, 2510, 2818, 2916, 2951, 3918
`\clist_new:N` 247, 1066, 1289, 1426, 3947
`\clist_put_left:Nx` 1070
`\clist_put_right:Nn` 1232
`\clist_put_right:Nx` 1472, 3941
`\clist_set:Nn` 1409, 1410
`\clist_set:No` 1051
`\contentsname` 2141
`contentsname` 12
cs commands:
`\cs:w` 2625
`\cs_end:` 2625
`\cs_generate_variant:Nn`
 64, 70, 107, 161, 560, 572, 1015,
 1134, 1148, 1221, 1241, 1242, 1405, 1442, 1544, 1582, 1590
`\cs_gset_eq:cc` 1208, 1211
`\cs_gset_eq:NN` 178, 270, 271,
 272, 370, 375, 380, 417, 419, 484, 488, 592, 878, 882, 2876
`\cs_gset_nopar:Npx` 879
`\cs_gset_protected_nopar:cpx` 1096, 2514
`\cs_gset_protected_nopar:Npn` 826, 2537
`\cs_gset_protected_nopar:Npx` 917, 1446, 1540
`\cs_if_eq:NNT` 73
`\cs_if_exist:cF` 873, 915, 953
`\cs_if_exist:cTF` 2612
`\cs_if_exist:NF` ... 465, 569, 885, 1445, 1537, 1911, 2838
`\cs_if_exist:NTF` 202, 2836
`\cs_if_exist_p:c` 120
`\cs_if_exist_use:c` 1118, 1119
`\cs_if_exist_use:cF` 8, 856
`\cs_if_exist_use:cT` 67, 1559
`\cs_if_exist_use:cTF` 1140
`\cs_if_free:cF` 554, 702, 717, 1205
`\cs_if_free:cTF` 1349
`\cs_if_free:NF` 564, 840, 2004
`\cs_if_free:NTF` 1990
`\cs_new:Npn` 159, 217, 224, 235, 1043, 1121
`\cs_new_eq:cc` 1846
`\cs_new_eq:cN` 10, 11, 12, 13, 24, 27, 1832
`\cs_new_eq:Nc` 1898
`\cs_new_eq:NN` 192, 201,
 206, 207, 212, 213, 215, 242, 243, 261, 466, 561, 687,

 748, 894, 919, 1268, 1274, 1697, 1833, 1834, 1887, 1897,
 2110, 2111, 2112, 2507, 2508, 2536, 2630, 2834, 2839, 3923
`\cs_new_nopar:Npn` 934, 1113, 1166, 1177, 3948
`\cs_new_protected:Npn` 53,
 59, 76, 108, 118, 132, 165, 171, 185, 819, 1044, 1063,
 1373, 1461, 1475, 1626, 1892, 2258, 2551, 2971, 2977, 3924
`\cs_new_protected_nopar:cpx` 1816
`\cs_new_protected_nopar:Npn`
 65, 91, 93, 162, 163, 164, 244, 245,
 552, 562, 567, 573, 606, 631, 648, 689, 695, 698, 715,
 720, 728, 737, 746, 854, 863, 876, 881, 883, 895, 909,
 936, 951, 964, 1008, 1021, 1067, 1075, 1094, 1114, 1124,
 1149, 1187, 1199, 1201, 1223, 1225, 1310, 1335, 1344,
 1389, 1400, 1427, 1436, 1443, 1449, 1469, 1482, 1501,
 1515, 1522, 1534, 1554, 1570, 1588, 1591, 1593, 1606,
 1618, 1652, 1668, 1673, 1706, 1724, 1743, 1754, 1758,
 1790, 1798, 1802, 1823, 1829, 1850, 1861, 1889, 1989,
 1991, 2016, 2051, 2061, 2066, 2178, 2206, 2502, 2608,
 2617, 2622, 2889, 2904, 2936, 2941, 3111, 3115, 3915, 3939
`\cs_new_protected_nopar:Npx` 577, 1254
`\cs_set_eq:cc` 704, 718
`\cs_set_eq:cN` 1905
`\cs_set_eq:Nc` 2504, 2505
`\cs_set_eq:NN` 180, 216, 298, 303, 308, 628, 646, 666, 692,
 696, 841, 888, 889, 1041, 1100, 1308, 1445, 1453, 1463,
 1464, 1465, 1537, 1538, 1539, 1678, 2117, 2122, 2125, 2130
`\cs_set_nopar:cpx` 1847
`\cs_set_protected_nopar:Npn` 691, 1864, 2978
`\cs_set_protected_nopar:Npx` 1894
`\cs_to_str:N` 109, 1253
`\cs_undefine:c` 68, 1079, 1080, 1084, 1085
`\cs_undefine:N` 181, 1466
cs4size 27
`\csname` 981, 983, 985, 1001, 2439, 2440, 2455,
 2456, 2458, 2461, 2463, 2464, 2472, 2473, 2474, 2477,
 2479, 2480, 3099, 3102, 3624, 3626, 3627, 3628, 3633, 3637
`\csuse` 3101, 3105
`\CTeX` 23, 97, 3072, 3075
ctex commands:
`\ctex_add_cmap:cn` 565
`\ctex_add_cmap:n` 42, 557, 562
`\ctex_add_cmap:Nn` 567, 572
`\ctex_add_to_selectfont:n` 72, 1892, 1917, 2003
`\ctex_appto_cmd:NnTF` 33, 164, 589, 1665
`\ctex_assign_heading_name:nn` ... 80, 2212, 2255, 2531
`__ctex_assign_heading_name:nnn` ... 80, 2257, 2258
`\ctex_at_end:n` 35, 244, 245, 402, 463, 1887, 3178
`\ctex_at_end_package:nn` 31, 31, 53,
 591, 612, 838, 843, 1768, 1770, 2571, 2592, 2636, 3074, 3097
`\ctex_auto_ignorespaces:` 45, 687, 696, 1934
`\l__ctex_autoindent_tl`
 35, 241, 297, 302, 307, 1977, 1981, 1984, 2053, 2056
`__ctex_calc_bounds:nn` 3943, 3944, 3948
`__ctex_ccglue:` 1878, 1887
`\l__ctex_ccglue_skip`
 1865, 1869, 1872, 1874, 1882, 2027, 2038, 2047, 2083, 2091

`\ctex_CJK_input:n` 44, 631, 646
`\c__ctex_class_tl`
 98, 100, 516, 521, 526, 2834, 2840, 2842, 2843, 2845
`__ctex_clear_fntef_color:n` ... 1790, 1798, 1802, 1815
`\c__ctex_cmap_encoding_seq` 42, 549, 550, 556
`\ctex_declare_math_sizes:nnnn` 94, 2936, 2970
`__ctex_def_heading_keys:n` 79, 2206, 2275
`\ctex_default_pt:n` 35, 220, 224
`__ctex_default_pt:w` 226, 235
`\ctex_detected_platform:` .. 68, 1706, 1724, 3119, 3128
`\l__ctex_encoding_tl` 35, 237, 238,
 330, 331, 599, 602, 674, 682, 1764, 1962, 1966, 2108, 2871
`\c__ctex_engine_file_tl` 1730, 1740
`\ctex_family_cmap:nn` 42, 552, 560
`\ctex_family_cmap:xn` 561
`\l__ctex_family_default_init_tl`
 67, 1675, 1696, 1698, 1703
`__ctex_family_default_wrap:n` 1678, 1697, 1700
`\ctex_file_input:n`
 32, 91, 504, 749, 1963, 1968, 1969, 2845,
 2872, 2873, 2980, 2982, 3120, 3124, 3132, 3178, 3277, 3278
`\ctex_file_wrapper:nnn` 31, 76, 92, 95, 633
`__ctex_fix_varioref_label:n` 88
`\ctex_fix_varioref_label:n` 2620, 2622
`\g__ctex_font_size_flag`
 .. 36, 261, 270, 271, 272, 484, 488, 510, 2875, 2876, 2979
`\c__ctex_font_size_prop` 93, 2891, 2902, 2908, 2946, 2973
`\c__ctex_font_size_seq` 2900, 2903, 2914
`\l__ctex_font_size_tl` 2891, 2892, 2938, 2939, 2973, 2974
`\ctex_fontset_error:n`
 98, 3111, 3364, 3398, 3419, 3464, 3483, 3525, 3567
`\g__ctex_fontset_tl` ... 334, 1708, 1736, 1746, 1747,
 3117, 3120, 3123, 3124, 3130, 3132, 3152, 3154, 3159, 3176
`\ctex_gadd_hook:cn` 31, 57
`\ctex_gadd_hook:Nn` 31, 59, 64, 697
`__ctex_get_font_sizes:Nn` 94, 2938, 2941
`\l__ctex_heading_bool` 413, 2830
`\c__ctex_headings_seq` 79, 2171, 2172, 2272, 2619
`\ctex_hypersetup:n`
 69, 69, 466, 1754, 1758, 1763, 1766, 1781, 1784
`\ctex_if_autoindent_touched:F`
 35, 242, 298, 303, 308, 2879
`\ctex_if_ccglue_touched:` 1875
`\ctex_if_ccglue_touched:TF` 72, 2022
`\ctex_if_ccglue_touched_p:` 72
`\ctex_if_macosx:TF` 68, 1731, 1732, 1734, 1743
`\ctex_if_pdfmode:TF` . 34, 207, 213, 585, 1767, 3363, 3418
`\ctex_if_preamble:TF` 34, 215, 216, 3147
`\ctex_ignorespaces_case:N` .. 46, 689, 1931, 1934, 1937
`__ctex_initial_heading:n` 79, 2178, 2274
`\l__ctex_line_spread_fp` 96, 290, 2877, 3065, 3067
`\l__ctex_line_stretch_tl` 2009, 2014, 2015, 2019
`\ctex_load_fontset:` 98, 3115, 3136, 3155, 3177
`\ctex_load_zhmap:n` 615
`\ctex_load_zhmap:nnnn`
 43, 606, 3320, 3392, 3458, 3519, 3561
`\l__ctex_ljtj_add_alternate_bool` 1203, 1215, 1233, 1240
`\ctex_ljtj_add_font_features:n` 57, 1223, 1241
`\ctex_ljtj_add_font_features:nn` 57, 1225, 1242
`\ctex_ljtj_add_font_features:x` 1265
`\ctex_ljtj_add_font_features:xn` 1224
`__ctex_ljtj_alternate_cs:n`
 1080, 1083, 1084, 1085, 1118, 1119, 1121, 1206,
 1209, 1210, 1212, 1213, 1395, 1432, 1494, 1519, 1559, 1576
`\l__ctex_ljtj_alternate_family_tl` 1457, 1460
`\l__ctex_ljtj_alternate_options_clist`
 1410, 1411, 1422, 1426
`\l__ctex_ljtj_alternate_seq`
 1048, 1382, 1383, 1388, 1394, 1402, 1650
`\l__ctex_ljtj_base_CJKfamily_tl`
 1049, 1065, 1116, 1494, 1519
`\l__ctex_ljtj_base_family_tl` 64,
 1081, 1111, 1117, 1438, 1479, 1538, 1557, 1562, 1563, 1575
`__ctex_ljtj_change_encoding:` ... 53, 1021, 1099, 1452
`\l__ctex_ljtj_char_range_clist` 1047,
 1364, 1375, 1391, 1409, 1421, 1433, 1456, 1478, 1629, 1631
`\ctex_ljtj_char_range_key:nn` 66, 1623, 1626
`__ctex_ljtj_char_range_parse_feature:w`
 66, 1634, 1638
`\g__ctex_ljtj_char_range_prop` ... 65, 1486, 1597, 1602
`\l__ctex_ljtj_char_range_tl`
 1486, 1489, 1595, 1597, 1599, 1601, 1608
`__ctex_ljtj_check_family:n` 55, 1075, 1393
`\ctex_ljtj_clear_alternate_font:n` 1548, 1554
`\l__ctex_ljtj_current_font_tl`
 50, 856, 860, 861, 873, 915, 920
`\l__ctex_ljtj_current_shape_tl` 938, 943, 948, 949
`\ctex_ljtj_declare_alternate_shape:nnnnnn`
 62, 1455, 1475, 1477
`\ctex_ljtj_declare_char_range:n` 1588, 1590
`__ctex_ljtj_declare_char_range:nn` 1589, 1591
`\ctex_ljtj_declare_char_range:nn` 65, 1592, 1593
`\ctex_ljtj_declare_char_range:x` 1585
`\ctex_ljtj_def_char_range_key:n` 65, 1598, 1618
`\g__ctex_ljtj_default_features_clist` 1069, 1288, 1289
`\ctex_ljtj_ensure_default_family:` ... 59, 1310, 1692
`\ctex_ljtj_extract_font:` 50, 888, 895
`__ctex_ljtj_family_csname:n` ... 1079, 1096, 1113, 1140
`\g__ctex_ljtj_family_font_name_prop`
 55, 1054, 1073, 1077, 1151, 1227, 1312, 1324
`\g__ctex_ljtj_family_font_options_prop`
 55, 1055, 1074, 1230
`\ctex_ljtj_family_if_exist:nTF` 56
`\ctex_ljtj_family_if_exist:xN` 1135
`\ctex_ljtj_family_if_exist:xNF` 1314
`\ctex_ljtj_family_if_exist:xNT` 1340
`\ctex_ljtj_family_if_exist:xNTF` 1126, 1319, 1337, 1557
`\g__ctex_ljtj_family_int` . 1193, 1195, 1471, 1472, 1474
`\g__ctex_ljtj_family_name_prop`
 55, 1072, 1081, 1103, 1137, 1438
`\l__ctex_ljtj_family_tl`
 1128, 1133, 1190, 1191, 1206, 1210, 1213, 1224, 1551, 1552

`__ctex_ljt_family_unknown_warning:n` 56, 1131, 1149, 1567
`\l__ctex_ljt_font_options_clist` 1051, 1056,
1057, 1058, 1066, 1102, 1231, 1232, 1235, 1429, 1430, 1434
`\ctex_ljt_fontspec:nn` 57, 1187, 1200
`__ctex_ljt_fontspec:nnn` 1201, 1221
`__ctex_ljt_fontspec:xnn` 1194
`\ctex_ljt_fontspec:xx` 1199, 1234, 1260
`\g__ctex_ljt_fontspec_family_tl`
..... 1101, 1104, 1105, 1106, 1112, 1117, 1142
`\g__ctex_ljt_fontspec_prop` 1189, 1217, 1222
`__ctex_ljt_gset_family_cs:nn` 55, 1059, 1094
`\ctex_ljt_if_alternate_shape_exist:n` 929
`\ctex_ljt_if_alternate_shape_exist:nF` 911
`\ctex_ljt_if_alternate_shape_exist:nT` 898, 938
`\ctex_ljt_if_alternate_shape_exist:nTF` 51
`\l__ctex_ljt_jfm_tl` 53, 1011, 1012, 1014, 1025, 1027, 1070
`\c__ctex_ljt_math_fam_int` 1033, 1034
`\c__ctex_ljt_math_family_tl`
..... 1346, 1348, 1350, 1353, 1357
`\c__ctex_ljt_math_tl`
... 1028, 1029, 1031, 1033, 1286, 1337, 1347, 1352, 1356
`__ctex_ljt_msg_def_family_map:n` 1164, 1166
`__ctex_ljt_msg_family_map:n` 1093, 1163, 1177
`__ctex_ljt_noembed_wrap:n` 1041, 1043
`__ctex_ljt_patch_external_font:w` 52, 901, 934
`\ctex_ljt_pickup_font:` 50, 869, 883, 894, 958
`__ctex_ljt_pop_fontname:` 872, 881, 960
`__ctex_ljt_push_fontname:n` 868, 876, 957
`\ctex_ljt_reset_alternate_font:n` 1550, 1570
`\g__ctex_ljt_reset_alternate_prop`
..... 1086, 1561, 1573, 1581
`\ctex_ljt_save_alternate_family:Nnnn`
..... 62, 1420, 1440, 1443
`\ctex_ljt_save_alternate_seq:cn` 1395
`\ctex_ljt_save_alternate_seq:Nn` 61, 1400, 1405
`\ctex_ljt_save_alternate_seq:Nnnwn` 61
`\ctex_ljt_save_alternate_seq:Nnnwnw` 1403, 1406
`__ctex_ljt_save_alternate_shape:cnn` ... 1493, 1518
`__ctex_ljt_save_alternate_shape:Nnn` 64, 1534, 1544
`\ctex_ljt_save_char_range:n` 65, 1596, 1603
`\ctex_ljt_save_char_range:nn` 1605, 1606
`__ctex_ljt_save_fontname:w` 878, 882
`\ctex_ljt_select_alternate_font:` 52, 936, 1920
`\ctex_ljt_select_font:` 49, 854, 1919
`__ctex_ljt_select_font_aux:` 49, 857, 863, 874
`\ctex_ljt_set_alternate_family:coonn` 1431
`__ctex_ljt_set_alternate_family:n` 1107, 1114
`\ctex_ljt_set_alternate_family:nn` .. 61, 1398, 1427
`\ctex_ljt_set_alternate_family:nnF` .. 61, 1052, 1389
`\ctex_ljt_set_alternate_family:nnn`
..... 62, 1439, 1447, 1449
`\ctex_ljt_set_alternate_family:Nnnnn` ... 1436, 1442
`\ctex_ljt_set_alternate_seq:n` 60, 1362, 1373
`\ctex_ljt_set_alternate_shape:n` ... 1503, 1515, 1526
`\ctex_ljt_set_alternate_shape:nnN` .. 63, 1488, 1522
`\ctex_ljt_set_alternate_shape:nnn` .. 63, 1491, 1498
`\ctex_ljt_set_alternate_shape:nnnn` 1500, 1501
`\ctex_ljt_set_alternate_shape:Nnnnnnn` 63, 1478, 1482
`\ctex_ljt_set_family:nnn` 54, 1044, 1064, 1218
`\ctex_ljt_set_family:xxx`
..... 1063, 1249, 1256, 1271, 1277, 1282, 1286
`\ctex_ljt_subst_font:` 51, 889, 909
`\ctex_ljt_swap_cs:NN` 62, 1454, 1461
`\ctex_ljt_switch_family:n` 1124, 1134, 1219, 1255
`\ctex_ljt_switch_family:x` 1123, 1191
`__ctex_ljt_tmp:w` 1463, 1465, 1466
`\l__ctex_ljt_tmp_seq` 1379, 1380, 1383, 1387
`\l__ctex_ljt_tmp_tl` .. 1050, 1051, 1077, 1088, 1091,
1228, 1236, 1252, 1255, 1256, 1337, 1338, 1340, 1341,
1377, 1378, 1379, 1413, 1415, 1416, 1423, 1633, 1641, 1650
`\l__ctex_ljt_tmpa_tl` 1314, 1319
`__ctex_ljt_um_char:Nw` 819, 841
`\g__ctex_ljt_unknown_family_seq` ... 1153, 1155, 1160
`__ctex_ljt_update_family_uid:N` 62, 1057, 1429, 1469
`\ctex_ljt_update_mathfont:` 59, 1332, 1335
`\ctex_ljt_update_mathfont:n` 1338, 1341, 1344
`\ctex_ljt_update_xkanjiskip:` 66, 1652, 1921
`__ctex_ljt_use_global_options:N`
..... 1058, 1067, 1411, 1430
`\l__ctex_ljt_xkanjiskip_skip`
..... 66, 1655, 1657, 1658, 1661, 1662
`\ctex_lua_now_x:n` 34, 201, 812,
830, 905, 931, 935, 940, 945, 967, 1020, 1517, 1541, 1710
`\c__ctex_macosx_file_tl` 1745, 1749
`\ctex_make_spa:nn` 113, 3915, 3923
`\ctex_mono_jfm:n` 53, 1008, 1015
`\ctex_mono_jfm:o` 1016, 1954
`\ctex_package_end_hook:n` 31, 65, 70
`\ctex_package_end_hook:o` 31, 74
`\ctex_parse_name:NN` 32, 108, 162, 163, 164
`\ctex_parse_name:NNn` 118, 161
`__ctex_parse_name:nNNnnN` 126, 132
`\ctex_parse_name:NNx` 109
`__ctex_parse_name:w` 144, 150, 159
`__ctex_patch_cmd:Nnn` 169, 171
`\ctex_patch_cmd:Nnn` 33, 165, 2580,
2582, 2586, 2589, 2595, 2597, 2601, 2604, 3086, 3087, 3088
`\ctex_patch_cmd:NnnTF` 33, 162, 176, 3099
`\ctex_patch_failure:N` . 183, 185, 594, 622, 624, 626,
629, 1302, 1304, 1306, 1309, 1666, 1907, 2567, 2574, 3106
`\ctex_plane_to_utfxvibe:Nn` 44, 648, 666
`\ctex_preto_cmd:NnTF` 33, 163, 621, 623,
625, 627, 1301, 1303, 1305, 1307, 1902, 2566, 2573, 3105
`\l__ctex_punct_bounds_clist` ... 3929, 3935, 3941, 3947
`\c__ctex_punct_family_clist` 700, 710
`\l__ctex_punct_font` 3927, 3928
`\ctex_punct_map_bfseries:nn`
..... 46, 720, 3305, 3317, 3323, 3389,
3395, 3454, 3455, 3461, 3514, 3515, 3516, 3522, 3558, 3564

<code>\ctex_punct_map_family:nn</code> . . .	46, 715, 3290, 3304, 3314, 3315, 3322, 3385, 3386, 3387, 3394, 3450, 3451, 3452, 3460, 3510, 3511, 3512, 3521, 3554, 3555, 3556, 3563
<code>\ctex_punct_map_itshape:nn</code>	47, 737, 3316, 3324, 3388, 3396, 3453, 3462, 3513, 3523, 3557, 3565
<code>\ctex_punct_map_series:nnn</code>	724, 725, 728
<code>\c_ctex_punct_seq</code>	3906, 3907, 3930
<code>\ctex_punct_set:n</code>	46, 698, 3285, 3321, 3384, 3393, 3449, 3459, 3509, 3520, 3553, 3562
<code>\ctex_punct_space:nn</code>	47, 746, 748
<code>\l_ctex_punct_tl</code>	389, 670, 671, 757, 1016, 1951, 1952, 1953, 1954
<code>_ctex_save_bounds:n</code>	3931, 3939
<code>_ctex_save_cmap:Nn</code>	569, 573
<code>_ctex_save_font_size:nn</code>	93, 2904, 2935
<code>_ctex_save_selectfont:</code>	1898, 1905
<code>\ctex_scheme_input:n</code>	32, 93, 107
<code>\ctex_scheme_input:o</code>	2638, 2847
<code>\l_ctex_scheme_tl</code>	422, 2638, 2847
<code>\g_ctex_section_depth_flag</code>	34, 192, 417, 419, 2760, 2781, 2802
<code>\ctex_select_size:</code>	1978, 1985, 1988, 1989, 2010, 2064
<code>\ctex_set_default_ccwd:Nn</code>	35, 217, 307, 1984, 2009
<code>\ctex_set_font_size:Nnn</code>	95, 2971, 2986, 2994, 3005, 3016, 3017, 3018, 3019, 3020, 3021, 3022, 3025, 3033, 3044, 3055, 3056, 3057, 3058, 3059, 3060, 3061
<code>_ctex_set_font_size:nnNn</code>	2974, 2977
<code>\ctex_set_ignorespaces:</code>	46, 691, 693, 695, 697
<code>\g_ctex_spa_iow</code>	3917, 3920, 3922, 3932
<code>\c_ctex_std_class_tl</code>	91, 2821, 2833, 2834, 2882
<code>\g_ctex_std_options_clist</code>	36, 247, 489, 501, 511, 513, 517, 522, 527
<code>_ctex_tmp:w</code>	178, 180, 181
<code>\l_ctex_tmp_box</code>	30, 52, 1853, 1854, 2553, 2557
<code>\l_ctex_tmp_dim</code>	30, 51, 2019, 2020, 2025, 2028, 2034, 2073, 2075, 2077, 2086, 2088
<code>\l_ctex_tmp_int</code>	30, 50, 651, 653, 654, 656, 660, 662, 2033, 2035, 2036, 2042, 2043, 2075, 2076, 2077, 2080, 2081, 2086, 2088, 2509, 2513, 2517
<code>\l_ctex_tmp_tl</code>	30, 49, 473, 480, 495, 496, 575, 576, 650, 652, 2208, 2266, 2277, 2278, 2946, 2947
<code>\ctex_update_ccglue:</code>	71, 1861, 2029, 2048, 2092
<code>\ctex_update_ccwd:</code>	71, 1850, 2023, 2100
<code>\ctex_update_default_family:</code>	67, 1673, 1695
<code>\ctex_update_em_unit:</code>	72, 1889, 2018, 2068
<code>\ctex_update_parindent:</code>	76, 1996, 2051, 2093
<code>\ctex_update_size:</code>	75, 1990, 1991, 2004
<code>\ctex_update_stretch:</code>	75, 1995, 2016
<code>\ctex_update_ziju:</code>	76, 1998, 2066
<code>\ctex_varioref_hook:</code>	88, 2617, 2628, 2636
<code>\c_ctex_version_tl</code>	7, 10, 11, 12, 13, 24, 27
<code>_ctex_write_family:nn</code>	3919, 3924
<code>\c_ctex_zero_tl</code>	1993, 2000, 2002
<code>\ctex_zhmap_case:nnn</code>	35, 243, 370, 375, 380, 3283, 3366, 3421, 3497, 3544
<code>\g_ctex_zhmCJK_bool</code>	35, 240, 369, 374, 379, 598, 673
<code>\ctex_zihao:n</code>	93, 2888, 2889
<code>\ctex_ziju:n</code>	2060, 2061
<code>\l_ctex_ziju_dim</code>	2069, 2070, 2074, 2085, 2087, 2091, 2095
<code>\l_ctex_ziju_tl</code>	1993, 2001, 2002, 2063, 2069
<code>\ctexbibname</code>	2156, 2165
<code>\CTEXcounter</code>	2112
<code>\CTEXdigits</code>	23, 77, 2115
<code>\CTEXfilltwosides</code>	28, 1823, 1833
<code>\CTEXindent</code>	28, 77, 2096, 2098
<code>\CTEXnoindent</code>	28, 77, 2102, 2104
<code>\CTEXnumber</code>	23, 77, 2113
<code>\CTEXoptions</code>	28, 42, 541, 543
<code>\ctexset</code>	5, 41, 532, 538, 544
<code>\CTEXsetfont</code>	21, 74, 1988
<code>\CTEXsetup</code>	28, 42, 535, 537
<code>\CTEXsout</code>	28
<code>\ctexspadef</code>	47, 748, 3934
<code>\CTEXthechapter</code>	15, 2371, 2374, 2587, 2602, 2603
<code>\CTEXtheparagraph</code>	15
<code>\CTEXthepart</code>	15, 2299, 2325
<code>\CTEXthesection</code>	15, 2580, 2589, 2596, 2605
<code>\CTEXthesubparagraph</code>	15
<code>\CTEXthesubsection</code>	15, 2582, 2598
<code>\CTEXthesubsubsection</code>	15
<code>\CTEXunderdblline</code>	28
<code>\CTEXunderdot</code>	28
<code>\CTEXunderline</code>	28
<code>\CTEXunderwave</code>	28
<code>\CTEXxout</code>	28
<code>\CurrentOption</code>	501

D

<code>declarecharrange</code>	24, 65
<code>\DeclareFontEncoding</code>	1018
<code>\DeclareFontFamily</code>	43, 589, 594, 1023, 3196, 3197, 3198, 3199, 3200, 3201
<code>\DeclareFontShape</code>	1024, 1026, 3204, 3205, 3206, 3207, 3208, 3209, 3210, 3211, 3212, 3215, 3216, 3217, 3218, 3219, 3220, 3221, 3222, 3223, 3228, 3229, 3230, 3231, 3232, 3233, 3234, 3235, 3236, 3239, 3240, 3241, 3242, 3243, 3244, 3245, 3246, 3247, 3252, 3253, 3254, 3255, 3256, 3257, 3258, 3259, 3260, 3263, 3264, 3265, 3266, 3267, 3268, 3269, 3270, 3271
<code>\DeclareFontSubstitution</code>	1019
<code>\DeclareMathSizes</code>	2939
<code>\DeclareSymbolFont</code>	1029, 1347
<code>\def</code>	974, 980, 993, 999, 2295, 2320, 2343, 2356, 2366, 2390, 2416, 2437, 2441, 2468, 2485, 2497, 2498, 2999, 3010, 3038, 3049, 3598, 3634
<code>\defaultCJKfontfeatures</code>	58, 1287, 1295
<code>dim commands:</code>	
<code>\dim_abs:n</code>	2087
<code>\dim_add:Nn</code>	2070
<code>\dim_compare:nNnF</code>	2055, 2078
<code>\dim_compare:nNnTF</code>	219, 2020, 2071
<code>\dim_eval:n</code>	2040, 2086

\dim_max:nn	2556	\expandafter	1001, 1002, 1004, 3599, 3600, 3624, 3625, 3626, 3627, 3628, 3633, 3637, 3638
\dim_min:nn	2087	\ExplSyntaxOff 80, 86, 175, 588, 777, 973, 1901, 2565, 3096, 3960	
\dim_new:N	51, 1860, 2095	\ExplSyntaxOn	85, 595, 806, 1007, 1909, 2577, 3109, 3899
\dim_set:Nn	1854, 1890, 1891, 2019, 2025, 2056, 2069, 2073, 2100, 2554, 3069	F	
\dim_sub:Nn	2077		
\dim_to_decimal:n	35, 2910, 2911, 2948	\familydefault	1681
\dim_to_decimal_in_unit:nn	3954	fancyhdr	28, <u>40</u>
\dim_use:N	227, 3069	\fangsong	3582
\dim_zero:N	2106	\fi	996, 1002, 1003, 1004, 2282, 2290, 2302, 2309, 2328, 2336, 2377, 2380, 2389, 2398, 2406, 2407, 2424, 2447, 2465, 2481, 2483, 2499, 2602, 3629, 3630, 3639, 3892
\directlua	795, 796, 975	fi commands:	
E		\fi: ..	514, 630, 667, 685, 932, 1879, 2583, 2590, 2983, 3073
		\figurename	2144
\E	114	figurename	12
\edef	987, 3617, 3618	file commands:	
\else	996, 1001, 1003, 2288, 2300, 2326, 2375, 2378, 2386, 2396, 2422, 2444, 2460, 2467, 2476, 2496, 3629, 3630, 3676, 3715, 3746, 3805, 3868	\g_file_current_name_tl	769, 1741
else commands:		\file_if_exist:nTF	1726, 1728, 1730, 1745, 3123, 3276, 3293
\else:	601, 932, 1879, 3073	\file_if_exist_input:nF	100
\end	97, 3099, 3105, 3106, 3999	\file_input:n	92, 101, 103, 611, 613, 639
\endCJKfilltwosides	1829	five commands:	
\endcsname ..	981, 983, 985, 1001, 2439, 2440, 2455, 2456, 2458, 2461, 2463, 2464, 2472, 2473, 2474, 2477, 2479, 2480, 3099, 3102, 3624, 3626, 3627, 3628, 3631, 3633, 3637	\c_five	2911
\endCTEXfilltwosides	1829, 1834	fntef	28, <u>40</u>
\endgroup	2457, 2495, 3617, 3625, 3640	\font	579, 3196, 3197, 3198, 3199, 3200, 3201
\endinput	3622	font@name commands:	
\endlinechar	3596, 3597	\font@name:	990
etex commands:		fontset	7, <u>38</u> , <u>99</u>
\etex_dimexpr:D	227, 2034	\fontsize	2892
\etex_glueexpr:D	2026	fontspec commands:	
\EverySelectfont	1911, 1914	\g_fontspec_encoding_tl	1022
exp commands:		\l_fontspec_family_tl	1480
\exp_after:wN	226, 885, 901, 903, 1738, 1740, 1741, 2892, 2939, 2974	_fontspec_namewrap:n	54, 1041
\exp_args:cc	2613	\fontspec_set_family:Nnn	1101, 1457
\exp_args:Nc	134	\footnotesep	96, 3069
\exp_args:Nnc	88	\footnotesize	3005, 3044, 3068
\exp_args:NNf	1641	four commands:	
\exp_args:NNNo	3068	\c_four	419, 660, 662
\exp_args:Nno	2625	fp commands:	
\exp_args:No	652, 3931	\fp_compare:nNnF	3065
\exp_args:Nx	402, 679, 753, 2060, 2888	\fp_compare:nNnT	2877
\exp_last_unbraced:Nf	144	\fp_eval:n	2063, 3950
\exp_not:c	127, 920, 1818, 1820, 1847, 2184, 2185, 2186, 2190, 2192, 2193, 2194, 2197, 2198, 2201, 2203, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2224, 2226, 2228, 2230, 2232, 2234, 2236, 2238, 2240, 2242, 2244, 2246, 2518, 2519, 2520, 2523	\fp_use:N	2000, 3067
\exp_not:N	127, 579, 681, 682, 683, 847, 848, 850, 956, 1103, 1104, 1105, 1106, 1420, 1541, 1683, 1684, 1685, 1700, 1819, 2193, 2516, 2523, 2908, 3130	G	
\exp_not:n	83, 142, 1100, 1101, 1678		
\exp_not:o	495, 849, 851, 1102, 1235, 1236, 1421, 1422, 1423, 1447, 1541, 1701, 1895, 2277, 2614	GBK	6, <u>37</u>
		\gdef	978, 2541, 2542, 2543, 2546, 2547, 2548
		\globaljfont	903
		group commands:	
		\group_begin:	110, 122, 167, 174, 229, 583, 811, 817, 821, 865, 887, 913, 955, 1046, 1098, 1367, 1451, 1536, 1556, 1572, 1677, 2196, 2560, 3068, 3091, 3926
		\group_end:	117, 125, 173, 179, 183, 234, 587, 597, 815, 825, 837, 870, 891, 926, 959, 1061, 1108, 1372, 1458, 1542, 1568, 1579, 1689, 2199, 2564, 3068, 3095, 3937
		\group_insert_after:N	923, 924

H

hbox commands:

\hbox_set:Nn	1853, 2553
heading	9, 39
\heiti	3581
\hskip	2452, 2469, 2490, 2497, 2498, 2596, 2598
\hspace	2298, 2299, 2324, 2325, 2374
\Huge	2337, 2360, 2409, 2429, 2673, 2692, 3022, 3061
\huge	2310,
	2334, 2347, 2404, 2657, 2669, 2676, 2684, 2699, 3021, 3060
hyperref	28, 40
\hypersetup	465, 466, 1755
\hyphenchar	3196, 3197, 3198, 3199, 3200, 3201

I

if commands:

\if_bool:N	598, 673
\if_case:w	510, 2979
\if_cs_exist:N	647
\if_meaning:w	1878
\ifctexpdf	206, 212
\ifdim	2286, 2394, 2420, 2449, 2487
\ifincsname	3073
\ifmmode	999
\IfNoValueF	539, 545
\IfNoValueTF	1253, 1505, 1610, 1643, 1645, 2261
\ifnum	2296, 2306, 2322, 2333, 2367, 2402, 2442, 2460, 2476, 3629
\ifx	1001, 3627, 3628, 3633
\ifzhmappdf	3645, 3712, 3743, 3774, 3841
\immediate	3636
indent	27, 37
\indexname	2147
indexname	12
\input	3898, 3963

int commands:

\int_compare:nNnF	2080, 2875
\int_compare:nNnTF	204, 653, 2036, 2760, 2781, 2802
\int_const:Nn	1033
\int_div_truncate:nn	75, 660
\int_eval:n	1506, 1508, 1509, 1611, 1613, 1614
\int_from_hex:n	652
\int_gincr:N	1193, 1471
\int_incr:N	2513
\int_mod:nn	662
\int_new:N	50, 1474
\int_set:Nn	643, 651, 2033, 2075
\int_set_eq:NN	637
\int_sub:Nn	656, 2081
\int_to_Hex:n	654, 659, 661
\int_use:N	580, 643, 1195, 1472, 2517, 3931
\int_zero:N	2509
\interlinepenalty	2303, 2330, 2344, 2357, 2408, 2428, 2453, 2454, 2491, 2492

iow commands:

\iow_close:N	3920
\iow_new:N	3922

\iow_now:Nx	3932
\iow_open:Nn	3917
\itemsep	3002, 3013, 3041, 3052

J

\jfam	1003, 1034
-------------	------------

K

\kaishu	3583
kernel commands:	

\l__kernel_expl_bool	84
----------------------------	----

keys commands:

\l_keys_choice_tl	403
\keys_define:nn	264, 495, 498,
	1038, 1360, 1468, 1545, 1583, 1622, 1926, 1947, 1971,
	2005, 2118, 2139, 2159, 2162, 2165, 2277, 2527, 3076, 3143
\keys_if_exist:nnF	1620
\keys_if_exist:nnT	2862, 2865
\l_keys_key_tl	250, 252, 254
\keys_set:nn	26, 280, 286, 342, 351, 360,
	386, 396, 403, 410, 438, 444, 532, 539, 545, 2641, 2681,
	2705, 2720, 2734, 2748, 2754, 2762, 2775, 2783, 2796,
	2803, 2804, 2806, 2850, 2864, 2866, 2874, 2878, 2880, 3185
\keys_set_known:nnN	1050

L

\labelformat	2625
\LARGE	3020, 3059
\Large	2307, 2652, 2663, 2712, 2716, 3019, 3058
\large	2723, 3018, 3057
\leavevmode	2282
\leftmargin	2999, 3010, 3038, 3049
\leftmargini	2999, 3010, 3038, 3049
\let . 979, 982, 984, 995, 996, 1005, 2443, 2992, 3031, 3620, 3626	
\linespread	3067
linespread	10, 37
linestretch	11, 75
\linewidth	2034, 2042, 2074
\lishu	3585
\listfigurename	2142
listfigurename	12
\listtablename	2143
listtablename	12
\LoadClass	518, 523, 528
\long	3634
\lstlistingname	848, 849
\lstlistlistingname	850, 851
\ltjdefcharrange	778, 779, 780, 782, 786, 787, 790, 793
LTJFONTUID	62
\ltjgetparameter	1655, 1858, 1882
\ltjsetkanjiskip	1872
\ltjsetmathletter	835
\ltjsetparameter	794, 800, 1669
\ltjsetxkanjiskip	1658

lua commands:

\lua_now_x:n	34
--------------------	----

luatex commands:

\luatex_directlua:D 201
 \luatexluaescapestring
 .. 931, 943, 970, 976, 1496, 1511, 1512, 1529, 1530, 1539

M

\MAKESPA 3923, 3965
 \markboth 2312, 2329
 \mathgroup 1003

max commands:

\c_max_dim 2020
 \mddefault 1019, 1024, 1030, 1348, 1353

minus commands:

\c_minus_one 261, 637, 2875

mode commands:

\mode_if_math:TF 828

msg commands:

\msg_critical:nnn 3112
 \msg_critical:nnx 769
 \msg_error:nn 3163
 \msg_error:nnn .. 23, 26, 42, 44, 760, 773, 776, 2893, 2975
 \msg_error:nnx 2134
 \msg_error:nnxx 3129, 3158
 \msg_fatal:nn 3904
 \msg_new:nn 3902
 \msg_new:nnn 187, 249, 251, 253, 255,
 257, 763, 1092, 1161, 1243, 1807, 2153, 2824, 3113, 3172
 \msg_new:nnnn 16, 34, 195, 2136, 2895, 3137, 3166
 \msg_warning:nn 1238, 1809, 1944, 2164
 \msg_warning:nnn 278, 284, 312, 322, 337, 346,
 355, 384, 394, 408, 436, 442, 450, 455, 461, 1156, 3080, 3150
 \msg_warning:nnnn 537, 543, 1818, 1825, 2098, 2104
 \msg_warning:nnx 186, 2843
 \msg_warning:nnxx 1088

N

nan commands:

\c_nan_fp 291
 \newCJKfontfamily 58, 1250
 \NewDocumentCommand 532, 535, 541, 1122, 1248,
 1250, 1258, 1263, 1269, 1275, 1280, 1285, 1287, 1406,
 1498, 1603, 1638, 1988, 2059, 2096, 2102, 2113, 2115,
 2255, 2887, 3072, 3580, 3581, 3582, 3583, 3585, 3586, 3588
 \newfontfeature 1035, 1036, 1037

nil commands:

\q_nil 819
 \nobreak
 2307, 2314, 2350, 2404, 2409, 2412, 2429, 2432, 2656, 2691
 \nobreakspace 2307, 2334
 nocap 27
 \noexpand 3618, 3620, 3621, 3622
 nofonts 27
 noindent 27
 nopunct 27
 \normalem 1789

\normalfont 627, 628, 629, 1272,
 1278, 1283, 1307, 1308, 1309, 1331, 2304, 2305, 2331,
 2332, 2345, 2346, 2358, 2359, 2400, 2401, 2426, 2427, 2523
 \normalsize 2737, 2751, 2757, 2986, 3025, 3063, 3071
 nospace 27
 nozhmap 27
 \numberline 86,
 2373, 2374, 2461, 2462, 2477, 2478, 2566, 2567, 2573, 2574

O

\O 114
 one commands:
 \c_one 271, 1499, 1604, 2088, 2256, 3943
 or commands:
 \or: 512, 2981

P

\P 114, 230
 \pagestyle 2870
 \par 2283, 2307, 2334, 2337,
 2347, 2360, 2404, 2409, 2429, 2643, 2656, 2670, 2685, 2691
 paragraph/aftername 17
 paragraph/afterskip 19
 paragraph/aftertitle 18
 paragraph/beforeskip 18
 paragraph/format 16
 paragraph/indent 19
 paragraph/name 14
 paragraph/nameformat 16
 paragraph/number 15
 paragraph/numberformat 16
 paragraph/titleformat 17
 \parindent 2055, 2056, 2078, 2099, 2100, 2105,
 2106, 2304, 2305, 2345, 2346, 2400, 2401, 2426, 2427, 2804
 \parsep 3001, 3002, 3012, 3013, 3040, 3041, 3051, 3052
 \part 81, 2281
 part/aftername 17
 part/afterskip 19
 part/aftertitle 18
 part/beforeskip 18
 part/format 16
 part/indent 19
 part/name 14
 part/nameformat 16
 part/number 15
 part/numberformat 16
 part/pagestyle 20, 80
 part/titleformat 17
 \partname 2307, 2334, 2645
 \PassOptionsToClass 517, 522, 527
 \PassOptionsToPackage 3, 599, 1759, 2108
 \patchcmd 162
 \pdffontattr 579
 \pdflastobj 580
 \pdfmapline 3646,
 3647, 3648, 3649, 3650, 3651, 3652, 3654, 3655, 3656,
 3657, 3658, 3659, 3660, 3662, 3663, 3664, 3665, 3666,

- 3667, 3669, 3670, 3671, 3672, 3673, 3674, 3775, 3776,
3777, 3778, 3779, 3780, 3781, 3783, 3784, 3785, 3786,
3787, 3788, 3789, 3791, 3792, 3793, 3794, 3795, 3796,
3798, 3799, 3800, 3801, 3802, 3803, 3842, 3843, 3844,
3845, 3846, 3847, 3849, 3850, 3851, 3852, 3853, 3854,
3856, 3857, 3858, 3859, 3860, 3862, 3863, 3864, 3865, 3866
- \pdfobj 576
- \pdfoutput 3629
- \pdfstringdefDisableCommands 3075
- pdftex commands:
- \pdftex_if_engine:TF
..... 239, 1960, 3281, 3361, 3416, 3482, 3495, 3542
- \pdftex_pdfoutput:D 202, 204
- \pretocmd 163
- prg commands:
- \prg_do_nothing: ... 1445, 1537, 1834, 1897, 1931, 2508
- \prg_new_conditional:Npnn 929, 1875
- \prg_new_protected_conditional:Npnn 1135
- \prg_return_false: 932, 1145, 1879, 1883
- \prg_return_true: 932, 1138, 1143, 1879, 1883
- \primitive 3999
- \ProcessKeysOptions 507
- prop commands:
- \prop_get:NnN 1230
- \prop_get:NnNT 1438
- \prop_get:NnNTF 1148, 1189, 1227, 1486, 2891, 2946, 2973
- \prop_get:NxNTF 1137
- \prop_gpop:NnNT 1077, 1081, 1573
- \prop_gput:Nnn 1054, 1217, 2908
- \prop_gput:Nno 1055, 1103, 1561, 1597
- \prop_gremove:Nn 1086
- \prop_if_empty:Nf 1151, 1312
- \prop_map_break:n 1326
- \prop_map_inline:Nn 1324
- \prop_new:N ... 1072, 1073, 1074, 1222, 1581, 1602, 2902
- \protect 138, 139, 2373, 2374, 2382, 2383, 2461, 2462, 2477, 2478
- \ProvidesFile 3634, 3642, 3709, 3740, 3771, 3838
- punct 10, 27, 39, 74
- \punctstyle 671, 1952
- Q**
- \quad 2439, 2587, 2589, 2596, 2598,
2602, 2603, 2605, 2662, 2698, 2708, 2724, 2738, 2752, 2758
- \quotation 97, 3088
- R**
- \raggedright 2304, 2345, 2400, 2426, 2655, 2690
- \refname 2149, 2155, 2161, 2162
- \refstepcounter 87, 2297, 2323, 2369, 2445, 2630, 2631
- \relax 978, 999, 1002, 1003, 2285,
2322, 2333, 2393, 2419, 2446, 2448, 2452, 2469, 2486,
2497, 2498, 2596, 2598, 3592, 3596, 3599, 3627, 3628, 3633
- \renewcommand 2281
- \RenewDocumentCommand 2631
- \RequirePackage 31, 47, 48, 457, 468, 600, 603, 604, 605, 752,
771, 774, 808, 816, 847, 1768, 1777, 1788, 1794, 2109, 2883
- \RequirePackageWithOptions 4
- resetalternatefont 25, 64
- reverse commands:
- \reverse_if:N 673
- \rmdefault 1683
- \rmfamily 621, 622, 1301, 1302
- S**
- \S 114
- scan commands:
- \scan_stop: ... 64, 227, 903, 1538, 1539, 2026, 2034, 3927
- scheme 9, 39
- scriptfont@name commands:
- \scriptfont@name: 989
- \scriptsize 3016, 3055
- \secdef 2292
- section/aftername 17
- section/afterskip 19
- section/aftertitle 18
- section/beforeskip 18
- section/format 16
- section/indent 19
- section/name 14
- section/nameformat 16
- section/number 15
- section/numberformat 16
- section/titleformat 17
- \selectfont .. 1129, 1564, 1577, 1902, 1907, 1923, 1990, 2892
- seq commands:
- \seq_clear:N 1048
- \seq_concat:NNN 1382
- \seq_gput_right:Nn 1155, 2914
- \seq_gset_from_clist:Nn 2172, 3907
- \seq_if_empty:Nf 1394
- \seq_if_in:NnF 1153
- \seq_if_in:NnT 556
- \seq_map_inline:Nn 1402, 2272, 2619, 3930
- \seq_new:N 549, 1160, 1387, 1388, 2171, 2903, 3906
- \seq_put_right:No 1650
- \seq_set_filter:NNn 1380
- \seq_set_from_clist:Nn 550
- \seq_set_split:NnV 1379
- \seq_use:Nnnn 2900
- \setCJKfamilyfont 58, 1174, 1248,
3296, 3301, 3308, 3309, 3310, 3311, 3312, 3313, 3338,
3339, 3347, 3348, 3350, 3351, 3352, 3353, 3355, 3376,
3378, 3380, 3382, 3409, 3410, 3411, 3412, 3435, 3440,
3445, 3447, 3473, 3475, 3477, 3478, 3488, 3489, 3490,
3491, 3503, 3504, 3505, 3506, 3507, 3508, 3533, 3534,
3535, 3536, 3537, 3538, 3550, 3551, 3552, 3574, 3575, 3576
- \setCJKmainfont .. 58, 1170, 1269, 1274, 1290, 3286, 3334,
3342, 3368, 3402, 3423, 3468, 3485, 3499, 3528, 3546, 3570
- \setCJKmathfont 58, 1285, 1293
- \setCJKmonofont 58, 1172, 1280, 1292, 3307, 3337,
3346, 3375, 3408, 3434, 3472, 3487, 3502, 3532, 3549, 3573
- \setCJKromanfont 1274, 1294

\setCJKsansfont	19
..... 58, 1171, 1275, 1291, 3289, 3295, 3300, 3336,	14
3344, 3374, 3407, 3429, 3471, 3486, 3501, 3530, 3548, 3572	16
\SetSymbolFont	15
..... 1031, 1352, 1356	16
\sfdefault	17
..... 1684	17
\sffamily	19
..... 623, 624, 1303, 1304	18
six commands:	18
\c_six	19
2911	18
skip commands:	18
\skip_horizontal:N	16
..... 1865, 1869	16
\skip_if_eq:nnT	17
..... 1654	17
\skip_if_eq:nnTF	19
..... 1882	14
\skip_new:N	16
..... 1661, 1874	16
\skip_set:Nn ...	16
..... 1657, 1662, 1858, 2027, 2038, 2083, 2091	15
\skip_zero:N	16
..... 2047	16
\small	17
..... 2994, 3033	17
\songti	17
..... 3580	17
\space	17
..... 2370, 2404, 2645, 2688, 2812	17
space	17
..... 10, 27, 39, 73	18
space commands:	18
\c_space_tl	18
..... 580	16
\special	16
..... 1775, 3677, 3678, 3679, 3680, 3681, 3682,	19
3683, 3685, 3686, 3687, 3688, 3689, 3690, 3691, 3693,	16
3694, 3695, 3696, 3697, 3698, 3700, 3701, 3702, 3703,	19
3704, 3705, 3716, 3717, 3718, 3719, 3720, 3722, 3723,	14
3724, 3725, 3726, 3728, 3729, 3730, 3731, 3733, 3734,	16
3735, 3736, 3747, 3748, 3749, 3750, 3751, 3753, 3754,	15
3755, 3756, 3757, 3759, 3760, 3761, 3762, 3764, 3765,	16
3766, 3767, 3806, 3807, 3808, 3809, 3810, 3811, 3812,	16
3814, 3815, 3816, 3817, 3818, 3819, 3820, 3822, 3823,	16
3824, 3825, 3826, 3827, 3829, 3830, 3831, 3832, 3833,	16
3834, 3869, 3870, 3871, 3872, 3873, 3875, 3876, 3877,	16
3878, 3879, 3881, 3882, 3883, 3884, 3886, 3887, 3888, 3889	17
\SplitArgument	12
..... 1499, 1604, 2256	12
stop commands:	12
\q_stop ...	12
..... 145, 150, 159, 227, 235, 1403, 1407, 1634, 1639	12
str commands:	12
\str_case:nnTF	12
..... 136	12
\str_case:onF	12
..... 1681	12
\str_case:onTF	12
..... 150	12
\str_case_x:nnF	12
..... 1168, 1179	12
\str_if_eq:nnF	12
..... 3169	12
\str_if_eq:nnTF	12
..... 1010	12
\str_if_eq:onF	12
..... 674, 3176	12
\str_if_eq:onTF	12
..... 602, 1764, 1772, 2871, 3152	12
\str_if_eq_x:nnF	12
..... 670	12
\str_if_eq_x:nnTF	12
..... 142, 1316, 3149	12
\string	12
..... 981, 983, 985, 994, 1002	12
\strutbox	12
..... 3069	12
sub3section	12
..... 9, 39	12
sub4section	12
..... 9, 39	12
subparagraph/aftername	12
..... 17	12
subparagraph/afterskip	12
..... 19	12
subparagraph/aftertitle	12
..... 18	12
subparagraph/beforeskip	12
..... 18	12
subparagraph/format	12
..... 16	12
subparagraph/indent	12
..... 19	12
subparagraph/name	12
..... 14	12
subparagraph/nameformat	12
..... 16	12
subparagraph/number	12
..... 15	12
subparagraph/numberformat	12
..... 16	12
subparagraph/titleformat	12
..... 17	12
subsection/aftername	12
..... 17	12
subsection/afterskip	12
..... 19	12
subsection/aftertitle	12
..... 18	12
subsection/beforeskip	12
..... 18	12
subsection/format	12
..... 16	12
subsection/indent	12
..... 19	12
subsection/name	12
..... 14	12
subsection/nameformat	12
..... 16	12
subsection/number	12
..... 15	12
subsection/numberformat	12
..... 16	12
subsection/titleformat	12
..... 17	12
subsubsection/aftername	12
..... 17	12
subsubsection/afterskip	12
..... 19	12
subsubsection/aftertitle	12
..... 18	12
subsubsection/beforeskip	12
..... 18	12
subsubsection/format	12
..... 16	12
subsubsection/indent	12
..... 19	12
subsubsection/name	12
..... 14	12
subsubsection/nameformat	12
..... 16	12
subsubsection/number	12
..... 15	12
subsubsection/numberformat	12
..... 16	12
subsubsection/titleformat	12
..... 17	12
T	
\T	231
\t	113
\tablename	2145
tablename	12
\TeX	3073
TeX and L ^A T _E X ₂ _ε commands:	
\@italiccorr	1005
\@par	2453, 2491, 2709, 2725, 2739, 2764, 2785
\@afterheading	2317, 2353, 2388
\@afterindentfalse	2287, 2395, 2421
\@afterindenttrue	2289, 2397, 2423
\@Alph	2808, 2813
\@chapapp	2370, 2404, 2587, 2602, 2603
\@chapter	82, 2366
\@currentlabel	2610, 2614
\@currentx	8, 73
\@currname	8, 74
\@defaultunits	35
\@empty	88, 2443
\@endpart	2340, 2363
\@EverySelectfont@Init	1904
\@hangfrom	2452, 2490
\@ifclassloaded	2820
\@ifpackagelater	41, 43, 759, 772, 775, 1795
\@ifpackageloaded	22, 25, 55, 768, 1752, 2568, 2627
\@ifundefined	2438, 2462, 2478

\@listI	2992, 3031	\char	48
\@listi	2992, 2999, 3010, 3031, 3038, 3049	\CheckCommand	72
\@M 2303, 2330, 2344, 2357, 2408, 2428, 2453, 2454, 2491, 2492		\Chinese	28, 28
\@makechapterhead	83, 2385, 2387, 2390	\chinese	23, 23, 28, 28
\@makeschapterhead	83, 2416	\CJK@ignorespaces	45, 687
\@minus	2713, 2717, 2728, 2731, 2742, 2745, 2766, 2769, 2777, 2787, 2790, 2798, 2988, 2990, 2996, 2998, 3000, 3001, 3007, 3009, 3011, 3012, 3027, 3029, 3035, 3037, 3039, 3040, 3046, 3048, 3050, 3051	\CJK@encoding	53, 53, 55, 63, 862, 866, 950, 1017, 1018, 1020, 1022, 1023, 1024, 1026, 1030, 1032, 1347, 1350, 1352, 1356, 1496, 1511, 1512, 1529, 1530
\@nil	956	\CJK@envStart	45, 45, 45, 681
\@onlypreamble		\CJK@family	49, 49, 857, 859, 862, 867, 914, 950, 1126, 1563, 1573, 1575, 1923, 1924
..... 34, 615, 1290, 1291, 1292, 1293, 1294, 1295, 3136		\CJK@ignorespaces	46, 46, 692, 696
\@part	81, 2292, 2295, 2320	\CJK@input	44, 45, 646
\@pkgextension	10, 11, 12, 13, 24, 27, 73, 770	\CJK@loadBinding	44, 45, 677
\@plus	2710, 2713, 2717, 2726, 2728, 2731, 2740, 2742, 2745, 2766, 2769, 2771, 2777, 2787, 2790, 2792, 2798, 2988, 2989, 2990, 2996, 2997, 2998, 3000, 3001, 3007, 3008, 3009, 3011, 3012, 3027, 3028, 3029, 3035, 3036, 3037, 3039, 3040, 3046, 3047, 3048, 3050, 3051	\CJK@loadEncoding	44
\@popfilename	31, 71	\CJK@makeActive	45, 676
\@secntformat	84, 2437, 2446	\CJK@plane	44, 73, 564, 565, 575, 1911
\@sect	84, 2441	\CJK@surr	44, 44, 44, 647, 666
\@setfontsize	2978	\CJK@upperReset	45
\@spart	82, 2292, 2343, 2356	\CJKfamily	55
\@ssect	85, 85, 2485	\CJKfamilydefault	59, 67, 67, 67, 67, 67
\@startsection	85, 85, 88, 2516	\CJKglue	75, 75, 97
\@svsec	2443, 2446, 2452, 2470, 2471	\CJKhook	46, 46
\@svsechd	2468, 2497, 2498	\CJKrmdefault	43
\@tempdima	2554, 2556	\CJKsymbol	73
\@tempskipa	2285, 2286, 2287, 2291, 2393, 2394, 2395, 2399, 2419, 2420, 2421, 2425, 2448, 2449, 2486, 2487	\CJKunderdotbasesep	28
\@topnewpage	2385	\CJKunderline	28
\@xsect	2484, 2500	\clearpage	45
\abstractname	12	\contentsname	12, 12
\addCJKfontfeature	64	\CS	50, 50
\AfterEndEnvironment	97	\csname	97, 97
\AfterEndPreamble	34, 34	\CTEX@aftertitle	2494, 2498, 2505, 2508
\appendixname	12, 20, 20, 20	\CTEX@appendixnumber	2532, 2542, 2547
\AtBeginDocument	34, 44	\CTEX@chapter@afterskip	2414, 2434
\AtEndEnvironment	97	\CTEX@chapter@aftertitle	2411, 2431
\AtEndOfClass	35	\CTEX@chapter@beforeskip	2393, 2419
\AtEndOfPackage	35	\CTEX@chapter@format	2401, 2427
\AtEndOfPackageFile*	31	\CTEX@chapter@indent	2401, 2427
\baselineskip	10, 11, 66, 75, 75	\CTEX@chapter@pagestyle	2269, 2391, 2417
\bfseries	46	\CTEX@chapter@titleformat	2410, 2430
\bibname	12	\CTEX@chaptername	2405
\c@chapter	2813	\CTEX@Family@CMap	561, 589, 592
\c@secnumdepth		\CTEX@part@afterskip	2316, 2352
... 2296, 2306, 2322, 2333, 2367, 2402, 2442, 2460, 2476		\CTEX@part@aftertitle	2313, 2339, 2349, 2362
\c@section	2808	\CTEX@part@beforeskip	2285
\caption	28	\CTEX@part@format	2305, 2332, 2346, 2359
\catcode	31, 31, 44, 44, 108	\CTEX@part@indent	2305, 2346
\ccwd . 11, 11, 11, 21, 21, 21, 28, 35, 71, 72, 72, 75, 75, 75, 76		\CTEX@part@pagestyle	2268, 2321
\chapter	9, 14, 20, 20, 21, 82, 91	\CTEX@part@titleformat	2311, 2338, 2348, 2361
\chaptermark	13	\CTEX@partname	2308, 2335
\chaptername	15	\CTEX@postappendix	2535, 2543, 2548
		\CTEX@postchapter	2548
		\CTEX@postsection	2543
		\CTEX@preappendix	2534, 2541, 2546
		\CTEX@prechapter	2546
		\CTEX@presection	2541

\CTEX@save@appendix	2536, 2539	\footnotesize	96
\CTEX@save@refstepcounter	2630, 2633	\get@external@font	897
\CTEX@selectfont@hook		\getanddefine@fonts	979, 993
..... 72, 1894, 1895, 1897, 1902, 1911, 1914		\globaljfont	50
\CTEX@set@titleformat@n	85, 2502, 2522	\heiti	8
\CTEX@setcurrentlabel@n	2608, 2634	\Hy@driver	1772
\CTEX@thechapter	2547	\Hy@unicodetrue	69
\CTEX@thesection	2542	\hypersetup	115
\CTEX@titleformat@n .. 85, 85, 85, 2493, 2498, 2504, 2507		\HyPsd@ConvertToUnicode	69
\CTEX@toc@width@n	2551, 2566, 2573	\HyPsd@LoadUnicode	70
\CTEX@todayold	2117, 2122	\HyPsd@pdfencoding	70, 70
\CTEX@verbatim@font@hook	1665, 1668	\HyPsd@ToBigChars	70
\ctex@zhmap@endinput	3894	\if@mainmatter	2368, 2403, 2602, 2838, 2839
\CTEXdigits	23, 23	\if@noskipsec	2282
\CTEXindent	115	\if@twocolumn	2384
\CTEXnoindent	115	\if@twoside	2581, 2588
\CTEXnumber	23, 23, 23, 23, 23	\ifin@	995, 1003
\CTEXoptions	28, 28, 115	\indexname	12
\ctexset 5, 5, 5, 5, 5, 5, 7, 7, 10, 10, 11, 11, 11, 12,		\itdefault	51
12, 12, 12, 12, 12, 12, 12, 14, 14, 15, 15, 16, 16, 16,		\itshape	47
17, 17, 18, 18, 19, 19, 20, 20, 20, 24, 25, 27, 28, 28, 99, 115		\kaishu	8
\CTEXsetup	28, 115	\labelformat	88, 88
\CTEXsout	115	\leftmark	13
\CTEXthechapter	13, 13, 13, 15	\linewidth	75, 75
\CTEXthesection	13, 13, 15, 87	\lishu	8, 8
\CTEXunderdblline	115	\listfigurename	12
\CTEXunderdot	115	\listtablename	12
\CTEXunderdotbasesep	28	\ltj@@al@getanddefine@fonts	979, 996
\CTEXunderline	28, 115	\ltj@@does@alt@set	51
\CTEXunderwave	115	\ltj@@getjfontnumber	52, 966
\CTEXxout	115	\ltj@@IsFontJapanese	53, 974, 978, 994
\curr@fontshape	868, 898, 907, 911, 957	\ltj@@ja@getanddefine@fonts	980, 995
\DeclareAlternateKanjiFont	63	\ltj@@mathJapaneseFonts	978, 1002
\DeclareFontFamily	42, 51	\ltj@@set@stackfont	988, 989, 990
\DeclareFontShape	51	\ltj@allalchar	813, 831
\DeclareFontShape@	62, 1455	\ltj@curjfont	50, 73
\DeclareRobustCommand	32	\ltj@pickup@altfont@auxy	52, 951
\define@newfont	50, 890	\ltj@pickup@altfont@copy	52, 964
\do@subst@correction	50, 51, 889	\ltj@setpar@global	987
\end	97, 97, 115	\ltj@temp	995, 996, 997
\EverySelectfont	72	\ltj@tempcnta	994, 1002
\external@font	900, 901, 903	\ltj@tempcntc	52, 970
\extract@font	50, 888	\ltjalchar	48
\f@baselineskip	2026	\ltjsetparameter	66
\f@encoding	866, 906, 1001	\m@ne	
\f@family	49, 867, 906, 914	2296, 2306, 2367, 2402, 3196, 3197, 3198, 3199, 3200, 3201	
\f@series	862, 950	\markboth	13
\f@shape	862, 950	\markright	13
\f@size 862, 868, 945, 953, 956, 957, 1854, 1890, 2557		\math@bgroup	1000
\fangsong	8, 8	\math@egroup	1004
\figurename	12	\math@fonts	987
\font	51, 51	\mathbb	97
\font@name	49, 51, 871,	\maxdimen	11, 75
878, 879, 882, 885, 903, 904, 921, 981, 982, 983, 984, 985		\meaning	32
\fontcharwd	72, 72	\newcommand	32
\footnotesep	96, 96, 96, 115	\newrobustcmd	32

<code>\nfss@catcodes</code>	53	<code>\thesecion</code>	13, 15, 15, 20, 87
<code>\normalsize</code>	8, 8, 93	<code>\thispagestyle</code>	82
<code>\nouppercase</code>	44, 115	<code>\today</code>	11, 11
<code>\p@</code>	1854, 1890, 2334, 2382, 2383, 2392, 2404, 2413, 2418, 2433, 2557, 2670, 2686, 2691, 2693, 2700, 2988, 2989, 2990, 2996, 2997, 2998, 3000, 3001, 3007, 3008, 3009, 3011, 3012, 3027, 3028, 3029, 3035, 3036, 3037, 3039, 3040, 3046, 3047, 3048, 3050, 3051	<code>\ttfamily</code>	47
<code>\pagestyle</code>	10	<code>\uppercase</code>	44, 44
<code>\paragraph</code>	9, 9, 9, 9, 9, 18, 18, 19, 34, 34	<code>\use@mathgroup</code>	999
<code>\parindent</code>	11, 21, 75, 77, 97	<code>\usepackage</code>	28, 28
<code>\part</code>	14, 20, 82	<code>\verbatim@font</code>	1665, 1666
<code>\partname</code>	15	<code>\wrong@fontshape</code>	50
<code>\PassOptionsToClass</code>	41	<code>\x@protect</code>	139
<code>\PassOptionsToPackage</code>	69	<code>\XeTeXglyphbounds</code>	112
<code>\patchcmd</code>	32	<code>\yahei</code>	8, 8
<code>\pdfmapline</code>	102, 105	<code>\youyuan</code>	8, 8
<code>\pdfstringdef</code>	69	<code>\z@</code>	800, 803, 2286, 2304, 2345, 2394, 2400, 2420, 2426, 2449, 2487, 2989, 2997, 3008, 3028, 3036, 3047
<code>\pickup@font</code>	49	<code>\zhdigits</code>	23
<code>\pickup@jfont</code>	894, 982, 984, 986	<code>\zhnumber</code>	23, 23, 23, 23
<code>\ProcessKeysOptions</code>	26	<code>\zihao</code>	21, 21
<code>\protected</code>	49	<code>\ziju</code>	21
<code>\protected@edef</code>	2114, 2116, 2446, 2610	<code>\zw</code>	66, 66, 72
<code>\ProvidesFile</code>	108	tex commands:	
<code>\ps@fancy</code>	87, 2595, 2597, 2601, 2604	<code>\tex_divide:D</code>	75, 2035, 2076
<code>\ps@headings</code>	87, 2580, 2582, 2586, 2589	<code>\tex_endlinechar:D</code>	637, 643
<code>\refname</code>	12	<code>\tex_font:D</code>	3927
<code>\refstepcounter</code>	88	<code>\tex_iffalse:D</code>	212
<code>\relax</code>	44, 75	<code>\tex_iftrue:D</code>	206, 2839
<code>\reset@font</code>	628, 1308	<code>\tex_ignorespaces:D</code>	1123, 1261, 1266, 1937, 2060, 2888
<code>\rightmark</code>	13	<code>\tex_immediate:D</code>	576
<code>\rmfamily</code>	47	<code>\tex_lowercase:D</code>	115, 232, 575, 823, 1370
<code>\roman</code>	23	<code>\tex_uppercase:D</code>	1738
<code>\scantokens</code>	32	textfont@name commands:	
<code>\scriptfont@name</code>	984	<code>\textfont@name:</code>	988
<code>\section</code>	9, 9, 14, 14, 20, 21	<code>\the</code>	970, 3596, 3599, 3600, 3619, 3623
<code>\sectionmark</code>	13, 13	<code>\thechapter</code>	2370, 2373, 2404, 2587, 2602, 2603, 2689
<code>\selectfont</code>	52, 66, 72, 72, 72, 75	<code>\theparagraph</code>	2750
<code>\setCJKfamilyfont</code>	24, 24, 24	<code>\thepart</code>	2298, 2307, 2324, 2334, 2646
<code>\setCJKmainfont</code>	24	<code>\thesecion</code>	2580, 2589, 2596, 2605, 2707
<code>\sf@size</code>	983	<code>\thesubparagraph</code>	2756
<code>\sffamily</code>	47	<code>\thesubsection</code>	2582, 2598, 2722
<code>\size@update</code>	75, 75, 1990, 2004	<code>\thesubsubsection</code>	2736
<code>\sldefault</code>	51	<code>\thispagestyle</code>	2321, 2391, 2417
<code>\songti</code>	8	three commands:	
<code>\special</code>	6, 69	<code>\c_three</code>	417, 2080, 2781, 3944
<code>\split@name</code>	956	<code>\tiny</code>	3017, 3056
<code>\SplitArgument</code>	80	tl commands:	
<code>\ssf@size</code>	985	<code>\tl_clear:c</code>	1791, 1803, 2262
<code>\strutbox</code>	96	<code>\tl_clear:N</code>	302, 473, 496, 1595, 1599, 1924, 1981, 2278, 2943
<code>\subparagraph</code>	9, 9, 9, 9, 18, 18, 19, 34	<code>\tl_const:cn</code>	747, 770
<code>\subst@correction</code>	917, 924	<code>\tl_const:cx</code>	2182, 2188
<code>\tablename</code>	12	<code>\tl_const:Nn</code>	516, 521, 526, 617, 1017, 1028, 1297, 1298, 1299, 1300, 1740, 1749, 2821, 2840, 2842
<code>\textfont@name</code>	982	<code>\tl_const:Nx</code>	7, 1346, 2000
<code>\tf@size</code>	981	<code>\tl_gput_right:cn</code>	557
<code>\thechapter</code>	13, 15, 20	<code>\tl_gput_right:Nn</code>	62
		<code>\tl_gset:Nn</code>	1320, 1736, 1746, 1747
		<code>\tl_gset:Nx</code>	654, 657, 1679, 1708, 3154

```

\tl_gset_eq:NN ..... 1105, 1703
\tl_gset_rescan:Nnn ..... 1327
\tl_head:N ..... 2947
\tl_if_blank:nTF 1412, 1508, 1509, 1613, 1614, 1628, 1646
\tl_if_blank_p:n ..... 1381
\tl_if_empty:cT ..... 2624
\tl_if_empty:NF ..... 857, 2053
\tl_if_empty:NTF ..... 2193, 3117
\tl_if_eq:NNT ..... 1675
\tl_if_eq:NNTF ..... 1993
\tl_if_exist:NF .....
..... 61, 616, 618, 619, 620, 1296, 1298, 1299, 1300
\tl_if_exist:NT ..... 2882
\tl_if_exist:NTF ..... 98, 2158, 2161, 2833
\tl_map_function:nN ..... 114
\tl_map_inline:Nn ..... 1524
\tl_map_inline:nn ..... 2944
\tl_new:c ..... 2180, 2181
\tl_new:N ..... 49, 61,
237, 241, 618, 619, 620, 859, 860, 948, 1014, 1065, 1091,
1111, 1112, 1133, 1460, 1601, 1696, 2001, 2014, 2534, 2535
\tl_put_left:Nn ..... 71, 1904
\tl_put_right:Nn ..... 480, 1641, 2224, 2226,
2228, 2230, 2232, 2234, 2236, 2238, 2240, 2242, 2244, 2246
\tl_put_right:Nx ..... 1608, 2208, 2947, 2948
\tl_replace_all:Nnn ..... 1378, 1416
\tl_set:cn ..... 2260, 2263
\tl_set:Nn ..... 221, 222,
297, 330, 331, 608, 609, 610, 848, 850, 861, 949, 1011,
1012, 1049, 1116, 1128, 1377, 1413, 1415, 1456, 1629,
1633, 1923, 1966, 1977, 2015, 2266, 3075, 3327, 3328, 3329
\tl_set:Nx 238, 575, 650, 900, 1252, 1698, 1951, 1962, 2063
\tl_set_eq:NN .....
..... 866, 867, 914, 1022, 1117, 1142, 1563, 1575, 2002
\tl_trim_spaces:n ..... 1592
\today ..... 2117, 2122, 2125, 2130
today ..... 11, 78
token commands:
\token_get_replacement_spec:N ..... 126, 145
\token_to_str:N ..... 186, 1170, 1171,
1172, 1174, 1181, 1182, 1183, 1245, 2155, 2156, 2896, 3934
\toks ..... 3596, 3597, 3599, 3600, 3619, 3623
\topsep ..... 3000, 3011, 3039, 3050
\TrimSpaces ..... 535
\ttdefault ..... 1685
\ttfamily ..... 625, 626, 1305, 1306
two commands:
\c_two ..... 192, 272, 484, 488, 2081, 2557, 2760, 2802
\c_two_hundred_fifty_six ..... 653, 656
\typeout ..... 2370, 2371

```

U

```
um commands:
    \um_cs_set_eq_active_char:Nw ..... 840, 841
\undefined ..... 3621
\updefault 1019, 1024, 1026, 1030, 1032, 1348, 1350, 1353, 1357
```

```

use commands:
\use:c ..... 868, 906, 907, 957, 1033, 1083, 1576
\use:n ..... 210,
242, 1317, 1321, 1697, 1832, 1899, 2507, 2562, 2831, 3093
\use:x ..... 78,
123, 495, 845, 956, 1064, 1200, 1418, 1592, 2277, 2906, 3126
\use_i:nn ..... 207, 215
\use_i:nnn ..... 370
\use_ii:nn ..... 213, 216
\use_ii:nnn ..... 243, 375
\use_iii:nnn ..... 380
\use_none:n .. 298, 303, 308, 1100, 1320, 1453, 1833, 2112
\use_none:nn ..... 592
UTF8 ..... 6, 37

```

V

<code>\verse</code>	97, 3086, 3087
<code>\vskip</code>	2315, 2316, 2334, 2351, 2352, 2404, 2413, 2414, 2433, 2434, 2670, 2691
<code>\vspace</code>	2392, 2399, 2418, 2425

W

```
winfonts ..... 27
\write ..... 3636
```

x

<code>\x</code>	3598, 3601, 3602, 3603, 3604, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613, 3614, 3615, 3616, 3617, 3624, 3634, 3638
<code>\xdef</code>	981, 983, 985, 3637
<code>\xeCJKsetup</code>	753, 1799, 1869, 1932, 1935, 1938, 1953
xetex commands:	

```

xetex commands:
  \xetex_if_engine:F ..... 3900
  \xetex_if_engine:TF ..... 1967, 3531
\XeTeXcharglyph ..... 3931
\XeTeXglyphbounds ..... 3955

```

Y

\yahei	3588
\youyuan	3586

$$\mathbf{Z}$$

```

zero commands:
  \c_zero ..... 204, 270, 818, 822, 2036, 2876
  \c_zero_dim .... 2028, 2040, 2055, 2071, 2078, 2249, 2803
  \c_zero_fp ..... 2000, 2877, 3065
  \c_zero_skip ..... 2247, 2248
\zhdigits ..... 23, 2116
zhmap ..... 6, 27, 38
\zhnum ..... 2110
\zhnumber ..... 23, 2114
\zhnumsetup ..... 2126, 2131
\zhtoday ..... 2125, 2130
\zihao ..... 21, 93, 2887, 2896
zihao ..... 8, 36
\ziju ..... 21, 76, 2059
\zw ..... 801, 1657, 1662, 1858, 1891

```