CTEX 宏集手册

CTEX.ORG

2026/06/23 v2.6.0*

简介

CTEX 宏集是面向中文排版的通用 LATEX 排版框架，为中文 LATEX 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。

CTEX 宏集支持 LATEX、pdfLATEX、XeLATEX、LuaLATEX、upLATEX 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 以及中文文档类 ctexart、ctexrep、ctexbook 和 ctexbeamer 实现。

目录

第 1 节 介绍 **2**

第 2 节 简明教程 **3**

2.1 CTEX 宏集的组成 3
2.2 CTEX 宏集的安装和更新 3
2.3 使用 CTEX 文档类 4
2.4 使用 ctex 宏包 5

第 3 节 宏包选项与 \ctexset 命令 **5**

第 4 节 编译方式、编码与中文字库 **6**

4.1 编译方式 6
4.2 中文编码 6
4.3 中文字库 6

第 5 节 排版格式设定 **9**

5.1 文档默认字号 9
5.2 章节标题风格 10
5.3 排版方案选项 11

第 6 节 文档汉化 **13**

6.1 日期汉化 13
6.2 文档标题汉化 13
6.3 页面格式设置与汉化 15

第 7 节 章节标题样式设置 **16**

7.1 编号相关 17
7.2 格式相关 19
7.3 间距、缩进相关 23
7.4 目录、附录相关 27
7.5 辅助命令 28
7.6 示例 29

第 8 节 实用命令 **29**

8.1 字号与间距 29
8.2 中文数字转换 31
8.3 杂项 31

第 9 节 LuaLATEX 下的中文支持方式 **32**

9.1 LuaLATEX 下替代字体的设置 . . 32

第 10 节 CTEX 宏集的配置文件 **33**

10.1 修改宏包默认选项 33
10.2 宏包载入后的配置 34
10.3 配置标题中文翻译 34
10.4 自定义字体集 34

第 11 节 第三方宏包兼容 **34**

*ctex-kit rev. 358feb25.

第 12 节　对旧版本的兼容性　35
12.1 CTEX 0.8a 及以前的版本 35
12.2 CTEX 0.9–CTEX 1.0d 35
12.3 CTEX 1.02c 以后的 SVN 开发版 37
12.4 CTEX 2.2 之前的版本 37
12.5 CTEX 2.4.1 和 2.4.2 37
12.6 CTEX 2.5 之前的版本 38
第 13 节　宏集依赖情况与手工安装方法　38
第 14 节　开发人员　39
第 15 节　代码实现　41
版本历史　166
代码索引　169

第 1 节　介绍

历史

CTEX 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 XETEX 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 LATEX 的最新发展，特别是 LATEX3 的逐渐成熟，李清用 LATEX3 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 CTEX 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 TEX 的时候没有考虑到多国文字支持，特别是对多字节的中日韩表意文字的支持。这使得 TEX 以至后来的 LATEX 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 LATEX 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。由于中文和西文书写习惯的差异，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 CTEX 宏集的目的之一就是解决这些 LATEX 文档的汉化难题。

另一方面，随着 TEX 引擎和 LATEX 宏包的不断发展，LATEX 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 CTEX 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。

CTEX 宏集的许多实现细节离不开热心朋友们在 bbs.ctex.org 论坛[4]上的讨论，在此对参与讨论的朋友们表示感谢。

[1] http://code.google.com/p/ctex-kit/，该链接现已失效。
[2] https://github.com/CTeX-org/ctex-kit
[3] 比如：pdfTEX 引擎下的 CJK、zhmCJK 宏包，XETEX 引擎下的 xeCJK 宏包和 LuaTEX 引擎下的 LuaTeX-ja 宏包。
[4] 2018 年，CTEX 论坛因故无限期关闭，此链接现已失效。

关于宏集名字的说明

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

CTEX 宏集是由 CTEX 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 CTEX 套装的 TEX 发行版，与本文档所述的 CTEX 宏集并非是同一事物。

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 CTEX 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。

第 2 节　简明教程

2.1　CTEX 宏集的组成

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

表 1　CTEX 宏集的组成

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

2.2　CTEX 宏集的安装和更新

最常见的 TEX 发行版（TEX Live 和 MiKTEX）已收录 CTEX 宏集及其依赖的宏包和宏集。[5]如果本地安装 TEX Live 或 MiKTEX 不是完整版本，则可能需要通过这两个发行版提供的包管理器来安装宏包。

TEX Live 的包管理器是 tlmgr（TEX Live Manager）。用户可以在系统命令行中[6]执行

```
tlmgr gui
```

启动管理器的图形界面（Windows 用户也可以通过开始菜单的 TeX Live 20XX（年份）➔ TeX Live Manager 打开）。连接上远程仓库之后，搜索 `ctex` 即可安装。tlmgr 的图形界面使用 Perl 编写，在实践中发现容易造成系统假死。遇到这种问题的用户，也可以直接在系统命令行执行

[5] zhmCJK 宏包是个例外。当用户显式指定选项 `zhmap = zhmCJK` 时，CTEX 宏集依赖它。由于，它没有被 TEX Live 和 MiKTEX 收录，用户可能需要遵照其说明文档自行安装。

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

```
tlmgr install ctex
```

来安装 CTEX 宏集[7]。

MiKTEX 通常会在缺失宏包时自动完成安装。如需手动安装，可以使用其管理维护工具 MiKTEX Console。用户可以打开管理器，连接上远程仓库之后，在“Package”选项卡中搜索“ctex”并安装即可。也可以使用 mpm（MiKTEX Package Manager），在命令行执行

```
mpm --admin --install=ctex
```

来安装 CTEX 宏集。

若希望了解 CTEX 宏集具体的依赖情况或手工安装宏集的方法，请参阅第 13 节。

当我们将宏集的新版本发布于 CTAN，且为发行版的远程仓库更新后，用户就可以在本地通过包管理器获取新版本。

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

```
tlmgr update --all
```

来完整更新已安装的宏包。

对于 MiKTEX，在 MiKTEX Console 中找到“Updates”选项卡，检查更新后即可选择升级宏包。也可以使用 mpm，在命令行执行

```
mpm --admin --update
```

来进行更新。

2.3　使用 CTEX 文档类

如果用户需要在三个标准文档类或 *beamer* 的基础上添加中文及版式的支持，我们建议用户使用 *CTEX* 宏集提供的四个中文文档类。

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

例 1

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

你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译，也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。对高级用户，我们也推荐使用 upLaTeX 编译。
\end{document}
```

以下是使用 ctexbeamer 文档类编写中文演示文稿的一个示例。

[7] *nix 用户可能需要超级用户权限（sudo）才能正确安装宏集。

[8] 使用 (pdf)LATEX 时也能够使用 GBK 编码，但不推荐。（见 4.2 节）

例 2

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

2.4　使用 ctex 宏包

用户在使用非标准文档类及 *beamer* 时，如果需要添加中文及版式的支持，则可以使用 *ctex* 宏包。

对于建立在 LaTeX 标准文档类之上开发的文档类，在使用 ctex 宏包时加上 `heading` 选项，可以将章节标题设置为中文风格。

例 3

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

第 3 节　宏包选项与 \ctexset 命令

CTEX 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 CTEX 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，CTEX 宏集也提供了一系列选项。用户可以使用这些选项来控制 CTEX 宏集的行为。按形式分类，这些选项有的以传统的方式提供，有的以 ⟨*key*⟩=⟨*value*⟩ 的形式提供。按指定位置分类，这些选项又可以分为以下三类：

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

后续文档将在使用说明中对某些特殊的选项加以说明。

\ctexset　`\ctexset {⟨键值列表⟩}`

是 CTEX 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 ⟨*key*⟩=⟨*value*⟩ 格式的定义。例如设置摘要与参考文献标题名称（6.2 节）就可以使用：

例 4

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

\ctexset 采用 LaTeX3 风格的键值设置，支持不同类型的选项与层次化的选项设置，相关示例见 7 节。

第 4 节　编译方式、编码与中文字库

4.1　编译方式

CTEX 宏集会根据用户使用的编译方式[9]，在底层选择不同的中文支持方式（见表 2）。

表 2　CTEX 宏集的中文支持方式

编译方式	(pdf)LaTeX	XeLaTeX	LuaLaTeX	upLaTeX*
支持宏包	CJK	xeCJK	LuaTeX-ja	原生

* pLaTeX-ng（或称 ApLaTeX）与 upLaTeX 兼容。使用 pLaTeX-ng 编译时，ctex 采用与 upLaTeX 相同的设置。

不同的编译方式和中文支持方式会在一定程度上影响 CTEX 宏集的行为，比如宏包对文档编码、字体选择、空格、标点等的处理。具体细节将在本文档后续内容中进行阐述。

4.2　中文编码

GBK ☆
UTF8 ☆

指明编写文档时使用的编码。CTEX 宏集无法检测文档源文件的实际编码格式，因此需要用户通过选项声明。如果没有显式指定，则默认采用 UTF-8 编码。

使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译时，CTEX 宏集强制使用 UTF-8 编码，此时 GBK 选项无效；使用 (pdf)LaTeX 编译时，CTEX 宏集默认使用 UTF-8 编码，但用户也可以显式声明 GBK 选项，使 CTEX 宏集按 GBK 编码处理文档。

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。

我们建议编写新文档时始终使用 *UTF-8* 编码，而仅把 *GBK* 编码留给历史遗留文档。

4.3　中文字库

以往，为 LaTeX 文档配置中文支持是一件相当繁琐的事情。默认情况下，CTEX 宏集能自动检测用户使用的编译方式（参见 4.1 节）和操作系统[10]，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 CTEX 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 CTEX 仍然提供了一系列选项，供在 CTEX 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。

[9] LaTeX、pdfLaTeX、XeLaTeX、LuaLaTeX 及 upLaTeX。
[10] CTEX 宏集现在能够识别 macOS 及 Windows 系统，并将其他系统统一归为 Linux。

表 3　CTEX 宏集自动配置字体策略

	macOS Old[1]	macOS New[2]	Windows[3]	其他
XeLaTeX	xeCJK 华文字库	xeCJK 华文字库 + 苹方	xeCJK 中易字库 + 微软雅黑	xeCJK Fandol 字库[4]
LuaLaTeX[5]	LuaTeX-ja 华文字库	LuaTeX-ja 华文字库 + 苹方	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja Fandol 字库
pdfLaTeX	不可用	不可用	CJK + zhmetrics 中易字库 + 微软雅黑[6]	不可用
LaTeX + DVIPDFMx	不可用	CJK + zhmetrics 华文字库 + 苹方	CJK + zhmetrics 中易字库 + 微软雅黑[6]	CJK + zhmetrics Fandol 字库
upLaTeX + DVIPDFMx	不可用	zhmetrics-uptex 华文字库 + 苹方	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex Fandol 字库

1 Yosemite (10.10) 及以前的 macOS 系统。
2 El Capitan (10.11) 及以后的 macOS 系统。
3 仅支持 Windows Vista 及以后的 Windows 操作系统。
4 由马起园、苏杰、黄晨成等人开发的开源中文字体, 参见:https://www.ctan.org/pkg/fandol。
5 LuaLaTeX 编译时使用 LuaTeX-ja 宏包。对此, 第 9 节有特别说明。
6 微软雅黑字体并不总是有效, 这和选项 zhmap 的取值有关。

zhmap ☆ `zhmap = ⟨true|false|zhmCJK⟩`

指定字体映射机制。本选项只在使用 (pdf)LaTeX 编译时有意义。

`true` 这是该选项的默认值。[11]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)[12]。
`zhmCJK` 载入 zhmCJK 宏包[13], 由 zhmCJK 宏包提供从 CJK 字库到 `.ttf` 的映射。

fontset `fontset = ⟨adobe|fandol|founder|hanyi|mac|macnew|macold|ubuntu|windows|none|...⟩`

指定 CTEX 宏集加载的字库。

如果没有指定 `fontset` 的值, CTEX 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)。

CTEX 预定义了以下七种中文字库。

`adobe` 使用 Adobe 公司的四款中文字体, 不支持 *pdfLaTeX*。
`fandol` 使用 Fandol 中文字体, 不支持 *pdfLaTeX*。
`founder` 使用方正公司的中文字体。
`hanyi` 使用汉仪字库的中文字体。
`mac` 使用 macOS 系统下的字体, 不支持 *pdfLaTeX*, 根据版本分为 `macnew` 和 `macold` 两种。
`macnew` 使用 El Capitan 或之后的多字重华文字体和苹方字体。从 macOS 15 (Sequoia) 开始, 部分字体(苹方、楷体、仿宋、隶书、圆体)转为 *downloadable*, CTEX 会自动检测可用性: 可用时使用, 不可用时静默跳过或回退。若需要完整的字体集, 请在系统字体册中手动下载相关字体。

注意: 上述自动检测仅对 `macnew` 字体集中的默认字体生效。如果需要在 LuaLaTeX 下通过 `\setCJKmainfont` 等命令自定义使用 macOS 的 *downloadable* 字体, 需要配置 `OSFONTDIR` 环境

11 对于以键值对形式提供的选项, 文档以**粗体**来表示其默认值。下同, 不再额外解释。
12 使用 (pdf)LaTeX 编译时, 如果需要使用自定义的字体映射文件(比如需要使用 LaTeX + Dvips 编译), 或者希望使用 Type1 字库, 请禁用本选项。为此, 你可能需要安装 CJK 字体。参考 zhmetrics 宏包提供的脚本 CTeXFonts.lua。
13 zhmCJK 宏包基于 zhmetrics 和 CJK 宏包, 提供与 xeCJK 宏包类似的用户接口。

变量, 将字体目录加入搜索路径。例如, 在 shell 配置文件中添加:

例 5

```
export OSFONTDIR="/System/Library/Fonts;/Library/Fonts;~/Library/Fonts"
```

配置后需运行 `luaotfload-tool --update` 重建字体数据库。XeLaTeX 不受此限制, 因为 XeTeX 自身可通过系统接口发现字体。

`macold` 使用 Yosemite 或之前的华文字体。

`ubuntu` 使用 Ubuntu 系统下的思源宋体、思源黑体和 TeX 发行版自带的文鼎楷体, 不支持 *pdfLaTeX*。

`windows` 使用 Windows 系统下的中易字体和微软雅黑字体。当使用 (pdf)LaTeX 编译时, 微软雅黑仅在以下两种情形有效: 安装有 zhmCJK 宏包且选项 `zhmap=zhmCJK` 时, 或者安装有微软雅黑的 Type1 字体且选项 `zhmap=false` 时。

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

`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。CTeX 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 CTeX 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*(这会禁用 CTeX 宏集的操作系统检测功能和自动设定字库功能)。例如:

例 6

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

CTeX 宏集预定义的中文字库还定义了一些字体命令。除了在 `ubuntu` 字库中没有 `\fangsong` 的定义外, 所有字库都有以下四个字体命令:

`\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}`。

在 `windows`、`founder` 和 `macnew` 字库中, 额外定义了 `\lishu` 和 `\youyuan`:

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

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

在 `windows` 字库中还定义了 `\yahei`。出于兼容性的考虑, `\yahei` 命令在 `macnew` 字库中也有定义, 但实际调用苹方黑体:

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

在 `macnew` 字库中, 还定义了 `\pingfang`:

`\pingfang` 苹方黑体, CJK 等价命令 `\CJKfamily{zhpf}`。在 macOS 15 及之后, 苹方字体可能尚未下载, 此时该命令将回退为 `\heiti` 的别名。

第 5 节　排版格式设定

5.1　文档默认字号

experiment/font-size-system ☆ experiment/font-size-system = ⟨**word**|*letterpress*|⟨自定义名称⟩⟩

New: 2026-05-04

选择字号系统。本选项位于 experiment 命名空间下，表明其实验性质。本选项只能作为文档类或宏包选项使用，不能在导言区通过 \ctexset 设置。例如：

```
\documentclass[experiment/font-size-system=letterpress]{ctexart}
```

CTEX 提供两种预设字号系统：

word 默认值。源自 Microsoft Word / Adobe 等桌面排版软件的字号定义，亦与 GB/T 40070—2021 国家标准一致。这是 CTEX 历史以来使用的字号体系。

letterpress 金属活字排印字号体系的一种，基于严格的倍数关系。初号/二号/五号/七号形成 ×2 等比序列（42 : 21 : 10.5 : 5.25），一号/四号形成 ×2 序列（28 : 14），三号/六号/八号形成 ×2 序列（16 : 8 : 4）。历史上金属活字字号曾存在多种命名与点数约定，本预设仅为其中一种，并非唯一定义。

两种字号系统仅在以下六个字号上存在差异：

	word (bp)	letterpress (bp)	倍数关系
一号 (1)	26	28	四号 ×2
二号 (2)	22	21	五号 ×2
六号 (6)	7.5	8	三号 ÷2
小六 (-6)	6.5	7.5	小三 ÷2
七号 (7)	5.5	5.25	二号 ÷4
八号 (8)	5	4	六号 ÷2

两种字号系统的其余十个字号完全相同，且行距公式（字号的 1.2 倍）不变。

用户也可以定义自己的字号系统。将字号数据保存在 ctex-fontsize-⟨*name*⟩.def 文件中，放到工作目录或 TEX 目录树（TDS）中，然后指定 experiment/font-size-system=⟨*name*⟩ 即可。自定义文件应使用 \ctex_save_font_size:nn 为每个字号注册数据。例如：

```
% 文件：ctex-fontsize-custom.def
\clist_map_inline:nn
  {
    {  8 } { 5    bp } ,
    {  7 } { 5.5  bp } ,
    { -6 } { 6.5  bp } ,
    {  6 } { 7.5  bp } ,
    { -5 } { 9    bp } ,
    {  5 } { 10.5 bp } ,
    { -4 } { 12   bp } ,
    {  4 } { 14   bp } ,
    { -3 } { 15   bp } ,
    {  3 } { 16   bp } ,
    { -2 } { 18   bp } ,
    {  2 } { 22   bp } ,
    { -1 } { 24   bp } ,
    {  1 } { 26   bp } ,
    { -0 } { 36   bp } ,
    {  0 } { 42   bp }
  }
  { \ctex_save_font_size:nn #1 }
```

zihao ☆ `zihao = ⟨-4|5|false⟩`

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

该选项的默认值与 `scheme` 的取值有关。当 `scheme = chinese` 时，对标准文档类默认值为 5，即设置 `\normalsize` 为五号字；对 beamer 则为 `false`，使用文档类原有的设置。当 `scheme = plain` 时，该选项不设默认值，沿用标准文档类或 beamer 的原有设置。

表 4　标准字体命令与字号的对应

	zihao = 5		zihao = -4		10pt	11pt	12pt
字体命令	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

表 4 中字体命令对应的字号大小（bp）基于默认的 `word` 字号系统。当启用 `experiment/font-size-s = letterpress` 时，一号、二号、六号、小六、七号、八号这六个字号的 bp 数值会发生变化，其余字号保持不变。具体差异见 5.1 节。

10pt ☆
11pt ☆
12pt ☆

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

标准文档类的其他选项在 CTEX 文档类中依旧有效。例如，设置纸张大小和方向的 `a4paper` 和 `landscape`，设置单双面的 `oneside` 和 `twoside` 等。CTEX 会将这些选项传给标准文档类[14]。

5.2　章节标题风格

heading ☆ `heading = ⟨true|false⟩`

本选项只能在调用 ctex.sty 时作为宏包选项使用。

CTEX 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第 6.3 节和第 7 节。

CTEX 宏集提供的四个文档类总是启用该功能。如果在 ctex.sty 下启用该选项，将会检查当前是否使用 LATEX 标准文档类。若然，则该选项将会使得 ctex.sty 宏包的行为和 CTEX 宏集提供的四个中文文档类完全一致；若不然，则会根据 `\chapter` 是否有定义来使用 ctexbook 或者 ctexart 的标题设置。

[14] 事实上，LATEX 在文档类中的选项是全局设定的，除了对使用的文档类有影响外，也可能会影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中，那么该选项将会被自动激活。

sub3section ☆
sub4section ☆

修改 \paragraph 和 \subparagraph 的格式。

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

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

具体格式可参考 7.3 小节中的 runin 和 afterskip 选项。

注意，上述两个选项只有在非 beamer 文档类下 heading 选项启用的时候才有意义。亦即，只有在使用除了 ctexbeamer 的三个 CTEX 文档类或启用了 heading 的 ctex.sty 的时候才有意义。

5.3　排版方案选项

scheme ☆　scheme = ⟨**chinese**|*plain*⟩

选择文章的排版方案，预设有 chinese 和 plain 两种方案。

chinese　对 beamer 以外的文档类，调整默认字号为五号字，并调整行距为 1.3；汉化文档中的标题名字（如“图”、“表”、“目录”和“参考文献”等，见 6.2 节）；在 heading = true 的情况下[15]（5.2 节），还会将章节标题的风格修改为中文样式（见 7 节）。

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

plain　不调整默认字号和行距，不会汉化文档中的标题名字，也不会将章节标题风格修改为中文样式，同时不会调整 \pagestyle，并禁用 autoindent 选项。事实上，此时的 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*⟩

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[16]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**⟩

是否在生成的 PDF 中保留汉字后面的空格。该选项仅在使用 XeLaTeX/(pdf)LaTeX 编译时有效。

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[17]。
false　使用 (pdf)LaTeX 编译时：总是忽略掉汉字后面的空格，不论汉字后是什么；使用 XeLaTeX 编译时，等同于 auto 的效果。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

[15] 使用 CTEX 文档类，或者使用 ctex 宏包且开启该选项时。
[16] 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。
[17] LaTeX 将单个换行视作一个空格。

例 7

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

将得到“汉字 分词 技术 English”；使用

例 8

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

则会得到“汉字分词技术 English”。

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

linespread ☆ `linespread = ⟨数值⟩`

接受一个浮点数值，设置行距倍数。本选项的初始值与 scheme 有关。

scheme = chinese　对标准文档类初始值为 1.3，即 1.3 倍行距。此时，相邻两行的基线（\baselineskip）距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。对 beamer 不改变行距，即使用默认的单倍行距。

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

autoindent　`autoindent = ⟨true|false|数值|带单位的数值⟩`

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。若要显式使用 \ccwd 为单位，则必须在导言区进行设置。

true　等价于设置 autoindent = 2 或在导言区设置 autoindent = 2\ccwd。

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 9

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

linestretch ⋆ `linestretch = ⟨数值或长度⟩`

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

experiment/CJKecglue　experiment/CJKecglue = {⟨弹性长度⟩}

New: 2026-05-03

设置 CJK 文字与西文之间的间距（汉英文间胶）。这是一个实验性选项，目前仅在 XeLaTeX、LuaLaTeX 和 upLaTeX 下有效，使用 pdfLaTeX 编译时会产生警告。

在 XeLaTeX 下，此选项等价于设置 `\xeCJKsetup{CJKecglue={\hskip ...}}`；在 LuaLaTeX 和 upLaTeX 下，此选项设置 `xkanjiskip`。例如：

例 10

```
\ctexset{experiment/CJKecglue = 0.25em plus 0.15em minus 0.05em}
```

请注意，由于 pdfLaTeX 下的 CJK 宏包没有独立的汉英文间胶概念，此选项无法在所有引擎下提供一致的行为。因此，除非确定文档仅会在 *XeLaTeX* 或 *LuaLaTeX* 下编译，否则不建议使用此选项。

第 6 节　文档汉化

6.1　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。如本文档编译时的日期是“2026 年 6 月 23 日”。

today ⋆　today = ⟨**small**|*big*|*old*⟩

该选项用来控制 `\today` 命令的输出格式：

small　效果为“2026 年 6 月 23 日”。使用阿拉伯数字和汉字的日期格式。
big　效果为“二〇二六年六月二十三日”。使用全汉字的日期格式。
old　效果为“June 23, 2026”。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 11

```
\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。

6.2　文档标题汉化

这里主要介绍由 scheme 选项（5.3 节）控制的文档标题汉化功能。

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

contentsname ⋆　contentsname = ⟨名字⟩

设置目录标题名 `\contentsname`。中文默认为“目录”。

listfigurename ⋆　listfigurename = ⟨名字⟩

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

listtablename ⋆ listtablename = ⟨名字⟩

设置表格目录标题名 \listtablename。中文默认为“表格”。

figurename ⋆ figurename = ⟨名字⟩

设置图片环境标题名 \figurename。中文默认为“图”。

tablename ⋆ tablename = ⟨名字⟩

设置表格环境标题名 \tablename。中文默认为“表”。

abstractname ⋆ abstractname = ⟨名字⟩

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

indexname ⋆ indexname = ⟨名字⟩

设置索引标题名 \indexname。中文默认为“索引”。

appendixname ⋆ appendixname = ⟨名字⟩

设置附录标题名 \appendixname。中文默认为“附录”。

bibname ⋆ bibname = ⟨名字⟩

设置参考文献标题名。中文默认为“参考文献”。

在标准文档类中 article 的参考文献名使用宏 \refname，而 book 和 report 使用宏 \bibname。本选项会根据标准文档类的不同，自动设定 \refname 或是 \bibname。因此，对于标准文档类及对应的 CTEX 文档类可以统一地使用 bibname 选项来控制参考文献标题名。

对于 beamer 及对应的 ctexbeamer 来说，它们同时具有宏 \bibname 和宏 \refname。本选项仅控制其中的 \bibname；\refname 则交由 refname 选项控制。

proofname ⋆ proofname = ⟨名字⟩

设置证明环境的名称 \proofname。中文默认为“证明”。

如果使用 ctexbeamer 文档类或者在 beamer 文档类下使用 ctex 包，还会汉化常用定理类环境的诸如“定义”、“定理”和“引理”等名称。此时，还有下列三个选项。

refname ⋆ refname = ⟨名字⟩

设置参考文献标题名 \refname。中文默认为“参考文献”。

注意，三个标准文档类（及相应的 CTEX 文档类）的参考文献标题名由 bibname 选项统一设置，本选项仅适用于 beamer 及其对应的 ctexbeamer。在三个标准文档类（及相应的 CTEX 文档类）中使用 refname 选项会报错。

algorithmname ⋆ algorithmname = ⟨名字⟩

设置 beamer 中的算法环境标题名 \algorithmname。中文默认为“算法”。

continuation ⋆ continuation = ⟨名字⟩

设置 beamer 可断页的帧在续页标题中的延续标识 \insertcontinuationtext。中文默认为“(续)”。

6.3　页面格式设置与汉化

页面格式设置与汉化的功能（及章节标题样式设置功能，见第 7 节）由 ctexheading 宏包完成。加载该宏包时，或者使用 CTEX 文档类时，或者是使用 ctex 宏包并设定选项 heading = true 时，相关功能被激活。此时，整个文档的页面格式（page style）被设定为 headings，即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

同时，CTEX 宏包也会对默认的 headings 页面格式进行修改，使之调用 \CTEXthechapter、\CTEXthesection 等宏来正确显示中文的章节编号。

CTEX 宏包的默认页面格式设置是经过汉化的 headings，其基本效果如本文档所示，只在页眉一侧显示章节编号和标题，另一侧显示页码。

更复杂的页面格式可以通过调用 fancyhdr、titleps 等宏包来设置。CTEX 宏包同时也为这些自定义页面格式的包提供了以下宏供使用：

- \CTEXthechapter、\CTEXthesection 等章节编号（见 7.5 小节）。它们用来代替英文文档类中的 \thechapter、\thesection 等宏。
- \leftmark、\rightmark，它们是在使用章节标题命令后，自动设置的宏。它们实际是在与章节标题命令对应的标记命令 \chaptermark、\sectionmark 中调用 \markright 或 \markboth 生成的。

有关 LATEX 页面标记的含义与使用细节，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 12

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

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

CTEX 宏包已经对 fancyhdr 宏包进行了补丁，载入 fancyhdr 后，其 fancy 页面格式将使用 \CTEXthechapter 等宏显示中文章节编号。

关于 fancyhdr 的具体用法可以参见其宏包手册。通常也只要像在标准的英文文档类中使用 fancyhdr 一样定义页眉页脚格式即可，并不需要额外的定义。

下面我则给出一个与前例类似而稍复杂的例子，展示如何在文档中设置页眉内容与页眉的格式。

例 13

```
\documentclass{ctexart}
\ctexset{section={
    name={第,节},
    number=\arabic{section},
  }
}
\usepackage{fancyhdr}
\fancyhf{}
\lhead{\textnormal{\kaishu\rightmark}}
\rhead{--\ \thepage\ --}
\pagestyle{fancy}
% \sectionmark 的重定义需要在 \pagestyle 之后生效
\renewcommand\sectionmark[1]{%
  \markright{\CTEXifname{\CTEXthesection——}{}#1}}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

本例的页眉效果大致如下(有页眉线)：

第 1 节——天地玄黄 – 1 –

第 7 节　章节标题样式设置

CTEX 宏集对 LATEX 的标准文档类（article、report、book）和 beamer 进行了章节标题样式设置功能的扩充。章节标题样式设置功能（及页面格式设置与汉化功能，见第 6.3 节）由 ctexheading 宏包完成。加载该宏包时，或者使用 CTEX 文档类时，或者是使用 ctex 宏包并设定选项 `heading = true` 时，相关功能被激活。其中，独立使用 ctexheading 宏包时，本节介绍各选项的默认值与指定 `scheme = plain` 时相同。

本节涉及的所有选项均需使用 `\ctexset` 命令设置。

章节标题的样式选项是分层设置的。顶层的选项是章节标题名称（例如 `section`），次一层的选项是章节标题的样式(例如 `nameformat`)。章节标题名称包括 `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`。可用的样式选项包括：

- 编号相关(7.1 小节)：`numbering`, `name`, `number`
- 格式相关 (7.2 小节)：`format`, `nameformat`, `numberformat`, `titleformat`, `aftername`, `aftertitle`, `pagestyle`
- 间距、缩进相关 (7.3 小节)：`runin`, `hang`, `indent`, `beforeskip`, `afterskip`, `fixskip`, `break`, `afterindent`
- 目录、 附录相关 (7.4 小节)： `tocline`, `lofskip`, `lotskip`, `appendix/numbering`, `appendix/name`, `appendix/number`

注意，对 article 及其衍生的 ctexart 等文档类，没有 `chapter` 级别的标题；而对于 beamer 文档类，这些选项控制的是由 `\partpage`, `\sectionpage` 和 `\subsectionpage` 产生的标题样式，此时只有 `part`, `section` 和 `subsection` 这三层级别，并且 `runin`, `afterindent`, `fixskip`, `hang`, `break` 和 `tocline` 这六个选项无效。

多层选项之间用斜线分开，例如，part/name 选项设置 \part 标题的在数字前后的名称，而 section/number 选项设置 \section 标题的数字类型。

使用 \ctexset 设置多级选项时，还可以在同一个上级选项下设置多个下级选项。例如，同时设置 part 一级标题的 pagestyle 选项，chapter 一级标题的 format 与 pagestyle 选项和 section 一级标题的 name 与 number 选项：

例 14

```
\ctexset {
  part/pagestyle = empty,
  chapter = {
    format    = \raggedright,
    pagestyle = empty,
  },
  section = {
    name   = {第,节},
    number = \chinese{section},
  }
}
```

7.1　编号相关

part/numbering ⋆
chapter/numbering ⋆
section/numbering ⋆
subsection/numbering ⋆
subsubsection/numbering ⋆
paragraph/numbering ⋆
subparagraph/numbering ⋆

numbering = **true**|false

控制是否对不带星号的章节标题进行编号。各级标题的默认值均为 true。

LaTeX 标准的章节标题命令（如 \section）大体上完成四项工作：输出标题内容、对标题编号（计数器增加 1）、将标题列入目录（若调用了 hyperref 宏包还会添加 PDF 书签）、更新页眉页脚标记。带星号的章节标题命令（如 \section*）只简单地输出章节标题内容，但不对标题编号，不将标题列入目录或 PDF 书签，也不写入页眉页脚标记。与之不同的是，本选项仅仅是否对不带星号的章节标题进行编号。因此，当设置本选项为 false 时，除了不对标题编号以外，其余功能与正常标题一致：可以编入目录，并生成正确的 hyperref 目录超链接位置和页眉页脚标记。例如：

例 15

```
\documentclass{ctexbook}
\begin{document}
\tableofcontents
\chapter{A}
\chapter*{B}
\ctexset{chapter/numbering=false}
\chapter{C}
\end{document}
```

三章的标题分别为“第一章　A”、“B”和“C”，但在目录中则只出现“第一章　A”和“C”。

注意，章节标题是否编号还要受到 LaTeX 计数器 secnumdepth 的控制（可通过以下介绍的 secnumdepth 选项设置）。例如，对于 section 而言，其深度为 1。因此，section 会被编号，当且仅当 secnumdepth 不小于 1，并且 section/numbering 为 true，并且使用不带星号的章节标题命令（即 \section）。

secnumdepth ⋆

secnumdepth = ⟨整数或章节名称⟩

New: 2020-05-06

设置对章节标题进行编号的层次数。secnumdepth 的值可以是一个整数，也可以是 part，chapter 等名称。层次数与名称的对应关系见表 5。章节层次的默认设置见表 6。

本选项对 beamer/ctexbeamer 文档类无效。

表 5　章节层次

层次	名称	注
-1	part	book/report 类
0	chapter	book/report 类
0	part	article 类
1	section	
2	subsection	
3	subsubsection	
4	paragraph	
5	subparagraph	

表 6　章节层次的默认设置

文档类	secnumdepth	tocdepth
article	3 (subsubsection)	3 (subsubsection)
book/report	2 (subsection)	2 (subsection)
beamer	无效	3 (subsubsection)

如果没有特别说明, 以下将用“. . .”代表各级章节标题名。

.../name ⋆ `name = {⟨前名字⟩,⟨后名字⟩}`
`name = {⟨前名字⟩}`

设置章节的名字。所谓“章节的名字”, 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 16

```
\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。该选项的默认设置见表 7。

表 7　name 选项的默认设置

标题名	scheme = chinese	scheme = plain	注
part	{第,部分}	{\partname\space}	原 \partname 为 Part
chapter	{第,章}	{\chaptername\space}	原 \chaptername 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 \sectionname 为 \translate{Section}
section	同右	{}	
subsection (beamer)	{}	{\subsectionname\space}	原 \subsectionname 为 \translate{Subsection}
subsection	同右	{}	
subsubsection	同右	{}	
paragraph	同右	{}	
subparagraph	同右	{}	

.../number ⋆ number = {⟨数字输出命令⟩}

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

例 17

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

将会使 \section 的编号变为大写罗马数字（如 I、II 等）。

number 选项定义的同时将控制对章节计数器的交叉引用。在引用计数器时，记录在 LaTeX 辅助文件中的是 number 选项的定义。

但是， number 选项不会影响计数器本身的输出。即设置 section/number 不会影响 \thesection 的定义（但该选项会影响 \CTEXthesection 的定义，见后）。该选项的默认设置见表 8。

表 8　number 选项的默认设置

标题名	scheme = chinese	scheme = plain	原 \the⟨标题⟩ 等价定义
part (beamer)	\chinese{part}	\insertromanpartnumber	意义为 \Roman{part}
part	\chinese{part}	\thepart	\Roman{part}
chapter	\chinese{chapter}	\thechapter	\arabic{chapter}
section (beamer)	同右	\insertsectionnumber	意义为 \arabic{section}
section	同右	\thesection	\arabic{section}
subsection (beamer)	\arabic{section}.\arabic{subsection}	\insertsubsectionnumber	意义为 \arabic{subsection}
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}

7.2　格式相关

CTEX 宏集提供了 numberformat, nameformat, titleformat, format 这几个选项用来控制章节标题的格式。它们的作用范围如图 1 所示。具体用法见下文。

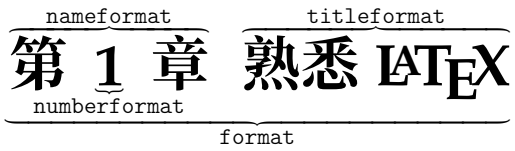

图 1　numberformat, nameformat, titleformat, format 几个选项的作用范围示意

.../format ⋆ format = {⟨格式命令⟩}
.../format+ ⋆ format+= {⟨格式命令⟩}

Updated: 2020-04-22

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+ 选项用于在已有的格式命令后附加内容。

format 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字、编号和标题内容（以及由 nameformat, numberformat, aftername, titleformat, aftertitle, indent 及 hang 选项设定的，应用于这些内容之上的格式），以实现特殊效果。

例如，设置章标题为无衬线字体左对齐、为节标题增加无衬线字体设置、为小节标题加框（\fbox 命令本身需带一个参数）：

例 18

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

format 选项的默认设置见表 9。

表 9　format 选项的默认设置

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

* 为了与 LaTeX 2ε 的默认效果保持一致，在 scheme = plain 时，part 和 chapter 的 nameformat 和 titleformat 并不一样，因此没有使用 format 选项统一设置名字和标题的格式。

.../nameformat ⋆ nameformat = {⟨格式命令⟩}
.../nameformat+ ⋆ nameformat+= {⟨格式命令⟩}

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。参见下面的 titleformat 选项。nameformat+ 用于在已有的章节名字格式命令后附加内容。

nameformat 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字和编号，以实现特殊效果（见例 24）。

nameformat 选项的默认设置见表 10。

表 10　nameformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\Large\bfseries
part (beamer)	同右	\usebeamerfont{part name} \usebeamercolor[fg]{part name}
part	{}	\huge\bfseries
chapter	{}	\huge\bfseries
section (beamer)	同右	\usebeamerfont{section name} \usebeamercolor[fg]{section name}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection name} \usebeamercolor[fg]{subsection name}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

.../numberformat ⋆ numberformat = {⟨格式命令⟩}
.../numberformat+ ⋆ numberformat+= {⟨格式命令⟩}

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, 当需要编号的格式和前后的章节名字不一样时可以使用。numberformat+ 用于在已有的编号格式命令后附加内容。

numberformat 选项的最后一个格式命令可以带有一个参数。这一参数用于接受编号数字。例如, 我们可以使用 numberformat 特别强调章标题中的数字:

例 19

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

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

numberformat 选项默认均设置为空, 故章节编号默认与章节名字使用相同的格式。

.../titleformat ⋆ titleformat = {⟨格式命令⟩}
.../titleformat+ ⋆ titleformat+= {⟨格式命令⟩}

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+ 选项用于在已有的标题格式命令后附加内容。

titleformat 选项的最后一个格式命令可以带有一个参数。这一参数用于接受标题内容。例如，实现多行标题的居中悬挂对齐：

例 20

```
\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

titleformat 选项的默认设置见表 11。

表 11　titleformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\huge\bfseries
part (beamer)	同右	\usebeamerfont{part title}
part	{}	\Huge\bfseries
chapter	{}	\Huge\bfseries
section (beamer)	同右	\usebeamerfont{section title}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection title}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

.../aftername ⋆ aftername = {⟨代码⟩}
.../aftername+ ⋆ aftername+= {⟨代码⟩}

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+ 用于在已有的代码后附加内容。该选项的默认设置见表 12。

.../aftertitle ⋆ aftertitle = {⟨代码⟩}
.../aftertitle+ ⋆ aftertitle+= {⟨代码⟩}

aftertitle 选项的参数 ⟨代码⟩ 将被插入到章节标题内容之后。aftertitle+ 用于在已有的代码后附加内容。该选项的默认设置见表 13。需注意，sub3section 或 sub4section 宏包选项（见 5.2 节）会影响 aftertitle 选项的默认值。

表 12　aftername 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	\quad	\par\nobreak
part (beamer)	同右	\vskip 1em \par
part	同右	\par\vskip 20pt
chapter	\quad	\par\nobreak\vskip 20pt
section (beamer)	同右	\vskip 1em \par
section	同右	\quad
subsection (beamer)	同右	\vskip 1em \par
subsection	同右	\quad
subsubsection	同右	\quad
paragraph	同右	\quad
subparagraph	同右	\quad

表 13　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

part/pagestyle ⋆
chapter/pagestyle ⋆

pagestyle = {⟨页面格式⟩}

设置 book/ctexbook 或 report/ctexrep 文档类中, \part 与 \chapter 标题所在页的页面格式 (page style)。该选项的默认设置见表 14。

表 14　pagestyle 选项的默认设置

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

7.3　间距、缩进相关

section/runin ⋆
subsection/runin ⋆
subsubsection/runin ⋆
paragraph/runin ⋆
subparagraph/runin ⋆

runin = true|false

runin 选项只对 \section 级以下标题有意义, 用于确定标题与随后的正文是否排在同一段之上。该选项的默认设置见表 15。

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

注意: section、subsection、subsubsection 的 aftertitle 默认值为 \@@par (分段命令),这会导致 runin 选项失效。若需要对这些层级使用 runin = true,必须同时将 aftertitle 设为空值:

```
\ctexset{section={runin=true, aftertitle={}}}
```

而 \paragraph、\subparagraph 的 aftertitle 默认值为 {}(空), 不存在此问题。

表 15　runin 选项的默认设置

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

.../hang ⋆ hang = true|false

Updated: 2020-04-23

hang 选项用于设置是否对章节标题实施悬挂缩进（缩进的宽度为名字宽度和 indent 选项设置的宽度之和）。

注意，当 hang = true 时，不恰当地设置选项 aftername 的值，可能会引发错误。这是因为当 hang = true 时，LaTeX 内部会构造一个 \hbox 而进入受限水平模式（restricted horizontal mode）。若在 aftername 中加入包含 \vskip 等会导致从受限水平模式切出的垂直命令（vertical command）时，就会报错。特别地，aftername 的默认值也可能导致这种情形（见表 12）。因此，当设置 hang = true 时，用户必须恰当地设置选项 aftername 的值。

本选项对 beamer/ctexbeamer 文档类无效。对于 \section 级以下标题，若设置了 runin 选项为 true，即标题与随后正文排在同一段，hang 选项没有意义。该选项的默认设置见表 16。

表 16　hang 选项的默认设置

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

表 17　indent 选项的默认设置

标题名	默认值
part	0pt
chapter	0pt
section	0pt
subsection	0pt
subsubsection	0pt
paragraph	0pt
subparagraph	\parindent
(sub3section)	0pt
(sub4section)	同上

.../indent ⋆ indent = {⟨缩进间距⟩}

Updated: 2020-04-23

indent 选项用于设置章节标题本身的首行缩进。该选项的默认设置见表 17。

如果 indent 的值是以 em、ex 或 \ccwd 为单位，那么缩进间距的大小是相对于 format 中指定的字号大小。例如，设置 \part 标题缩进三个字、\section 标题缩进 20 pt：

例 21

```
\ctexset{
  part = {
    format += \raggedright,
    indent  = 3\ccwd,
  },
  section = {
    format  = \Large\bfseries,
    indent  = 20pt,
  }
}
\part{首行缩进的标题}
\noindent 无缩进的正文。
\section{首行缩进的标题}
\noindent 无缩进的正文。
```

.../beforeskip ⋆ beforeskip = {⟨弹性间距⟩}

beforeskip 选项用于设置章节标题前的垂直间距。该选项的默认设置见表 18。

注意：若加载了 titlesec 宏包（且未使用其 loadonly 选项），titlesec 将完全接管章节标题命令。此时 beforeskip 选项不再生效，应改用 titlesec 的 \titlespacing 命令来设置标题前间距。

.../afterskip ⋆ afterskip = {⟨弹性间距⟩}

afterskip 选项控制章节标题下方或后面的间距。

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

该选项的默认设置见表 19。注意 sub3section 或 sub4section 宏包选项（见 5.2 节）会影响 aftertitle 选项的默认值。

注意：若加载了 titlesec 宏包（且未使用其 loadonly 选项），titlesec 将完全接管章节标题命令。此时 afterskip 选项不再生效，应改用 titlesec 的 \titlespacing 命令来设置标题后间距。

.../fixskip ⋆ fixskip = true|**false**

默认情况下，article、book 和 report 类的标题与正文的距离除了由 beforeskip 和 afterskip 选项设置的垂直间距外，还会有一些多余的间距。fixskip 选项用于抑制这些多余间距。该选项默认不开启。

表 18　beforeskip 选项的默认设置

标题名	默认值
part (article)	4ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	50pt
section (beamer)	0pt
section	3.5ex plus 1ex minus .2ex
subsection (beamer)	0pt
subsection	3.25ex plus 1ex minus .2ex
subsubsection	3.25ex plus 1ex minus .2ex
paragraph	3.25ex plus 1ex minus .2ex
subparagraph	3.25ex plus 1ex minus .2ex

表 19　afterskip 选项的默认设置

标题名	默认值
part (article)	3ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	40pt
section (beamer)	0pt
section	2.3ex plus .2ex
subsection (beamer)	0pt
subsection	1.5ex plus .2ex
subsubsection	1.5ex plus .2ex
paragraph	1em
(sub3section)	1ex plus .2ex
(sub4section)	同上
subparagraph	1em
(sub4section)	1ex plus .2ex

`.../break` ⋆ `break = {⟨格式命令⟩}`
`.../break+` ⋆ `break+= {⟨格式命令⟩}`

break 选项用于控制章节标题与之前正文的分隔关系。一般用于设置是否在标题之前分页或者设置行间罚点。break+ 用于在已有的格式命令后附加内容。

例如，若当前页剩余高度小于正文高度的一半时，则另起一页输出 \section 标题：

例 22

```
\usepackage{needspace}
\ctexset{section/break = \Needspace{.5\textheight}}
```

该选项的默认设置见表 20。

表 20　break 选项的默认设置

标题名	默认值
part (article)	{}
part	\if@openright\cleardoublepage\else\clearpage\fi
chapter	同上
section	\addpenalty{\@secpenalty}
subsection	同上
subsubsection	同上
paragraph	同上
subparagraph	同上

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

afterindent 选项用于设置章节标题后首段的缩进。

book 和 report 类的 \part 标题被单独排在一页之上，afterindent 选项没有意义。对于 \section 级以下标题，若设置了 runin 选项为 true，即标题与随后正文排在同一段，afterindent 选项也就没有了意义。

该选项的默认设置见表 21。

表 21　afterindent 选项的默认设置

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

7.4　目录、附录相关

tocdepth ⋆ tocdepth = ⟨整数或章节名称⟩

New: 2020-05-06

设置对章节标题编入目录的层次数。tocdepth 的值可以是一个整数，也可以是 part，chapter 等名称。层次数与名称的对应关系见表 5。章节层次的默认设置见表 6。

.../tocline ⋆ tocline = {⟨格式定义⟩}

tocline 选项用于定义章节标题在目录文件（.toc）中的格式。⟨格式定义⟩有两个参数：参数 #1 是 part、chapter 等名字，参数 #2 是标题内容。该选项的默认设置见表 22。

表 22　tocline 选项的默认设置

标题名	默认值
part	\CTEXifname{\CTEXthepart\hspace{1em}}{}#2
chapter (chinese)	\CTEXifname{\protect\numberline{\CTEXthechapter\hspace{.3em}}}{}#2
chapter (plain)	\CTEXnumberline{#1}#2
section	\CTEXnumberline{#1}#2
subsection	同上
subsubsection	同上
paragraph	同上
subparagraph	同上

这里 \CTEXnumberline 的意义是，若标题 #1 没有名字，则不输出 \numberline{\CTEXthe#1} 等编号：

```
\CTEXifname{\protect\numberline{\csname CTEXthe#1\endcsname}}{}
```

其中，\CTEXifname 的定义见 7.5 小节。

chapter/lofskip ⋆ lofskip = {⟨弹性间距⟩}
chapter/lotskip ⋆ lotskip = {⟨弹性间距⟩}

lofskip 选项控制插图目录（.lof）中，章之间的插图标题的距离。同样，lotskip 选项控制表格目录（.lot）中，章之间的表格标题的距离。

目前，这两个选项只在 chapter 标题下有定义。它们的默认值，在 scheme 选项的不同取值下都为 10 pt。

appendix/numbering ⋆ numbering = **true**|false

控制是否对附录章（对应 book 与 report）或附录节（对应 article）进行编号，用法与普通章节对应的 numbering 选项相同。该选项默认值为 true。

appendix/name ⋆
name = {⟨前名字⟩,⟨后名字⟩}
name = {⟨前名字⟩}

设置附录章(对应 book 与 report)或附录节(对应 article)的名字,用法与普通章节对应的 name 选项相同。

注意该选项与 appendixname 选项 (6.2 节) 在意义上有些重叠, 但不完全相同。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。

该选项的默认设置见表 23。

表 23　appendix/name 选项的默认设置

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

appendix/number ⋆
number = {⟨数字输出命令⟩}

设置附录章(对 book 与 report)或附录节(对 article)编号的数字输出格式,用法与普通章节对应的 number 选项相同。

appendix/number 选项同时也会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。

该选项的默认设置见表 24。

表 24　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}

7.5　辅助命令

CTEX 宏集还提供了一些辅助命令(宏),用于存储章节标题格式,或进行一些条件判断。

\CTEXthepart
\CTEXthechapter
\CTEXthesection
\CTEXthesubsection
\CTEXthesubsubsection
\CTEXtheparagraph
\CTEXthesubparagraph

以 \CTEXthe 开头的这组宏给出结合了 name 与 number 选项的章节编号输出格式。例如在 scheme = chinese 时,默认章编号输出格式就是 \CTEXthechapter,形如“第一章”。

这组宏在 CTEX 文档类中将代替 \thechapter 等宏的作用, 在章节中引用本章节的完整编号。例如用于帮助定义自定义的目录格式、页眉格式等。

\CTEXifname　\CTEXifname {⟨有名字时的内容⟩} {⟨无名字时的内容⟩}

\CTEXifname 会根据当前章节有无名字展开得到不同内容（通常是格式命令）。由于章节名字总是与编号一起出现，章节有无名字通常也表达为“章节是否编号”。在 LaTeX 中，后者取决于以下几个方面：章节深度是否不大于计数器 secnumdepth 的值，章节标题是否使用不带星号的命令。在 CTEX 宏集中，后者还取决于 .../numbering 是否为 true。

\CTEXifname 可用于 format, titleformat, aftertitle, afterskip, indent 这五个选项和 \chapter 标题 beforeskip 选项的格式设置之中。也可用于帮助定义自定义的目录格式、页眉格式等。

例如，设置章的标题有名字时左对齐，无名字时居中对齐，并且在标题后画一条横线。

例 23

```
\ctexset{
  chapter/format     = \CTEXifname{\raggedright}{\centering},
  chapter/aftertitle = \par\CTEXifname{}{\hrule},
}
```

7.6　示例

我们最后举一个稍微复杂的例子，来看看上述选项的综合应用。

例 24

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

本例的设置效果大致如下：

第 1 章

熟悉 LaTeX

第 8 节　实用命令

8.1　字号与间距

\zihao　\zihao {⟨字号⟩}

用于调整字号大小。其中 ⟨字号⟩ 的有效值共有 16 个，如表 25 所示。使用 \zihao 命令调整字体大小时，西文字号大小会始终和中文字号保持一致。

表 25　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

表 25 中的字号大小基于默认的 word 字号系统。当启用 experiment/font-size-system = letterpress 或用户自定义字号系统时，部分字号的 bp 数值会发生变化，详见 5.1 节。

\ziju　\ziju {⟨中文字符宽度的倍数⟩}

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

这个命令会影响 \ccwd 的值，但不会影响英文字距。

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

8.2　中文数字转换

CTEX 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。

\chinese　\chinese {⟨*counter*⟩}
\pagenumbering {chinese}

\chinese 命令与 \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。

\zhnumber　\zhnumber {⟨*number*⟩}

以中文格式输出数字。这里的数字可以是整数、小数和分数。

\zhdigits　\zhdigits {⟨*number*⟩}

将阿拉伯数字转换为中文数字串。

\CTEXnumber　\CTEXnumber \⟨*macro*⟩ {⟨*number*⟩}

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

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

\CTEXdigits　\CTEXdigits \⟨*macro*⟩ {⟨*number*⟩}

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

8.3　杂项

\CTeX　用于显示 CTEX 标志。

第 9 节 LuaLaTeX 下的中文支持方式

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

9.1 LuaLaTeX 下替代字体的设置

AlternateFont
Updated: 2020-04-30

```
\setCJKfamilyfont {⟨family⟩} {⟨base font name⟩}
  [
    ⟨base font features⟩ ,
    AlternateFont =
      {
        {⟨character range1⟩} {⟨alternate font name1⟩} ,
        {⟨character range2⟩} {⟨alternate font name2⟩} {⟨alternate font features2⟩} ,
        ......
      }
  ]
```

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⟩} {⟨alternate font name⟩}
  [
    CharRange = {⟨character range⟩} ,
    ⟨alternate font features⟩
  ]
```

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 25

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

declarecharrange ⋆
New: 2020-04-30

```
\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 name⟩}
⟨name⟩ = {⟨alternate font name⟩} {⟨alternate font features⟩}
```

⟨*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 ⋆
resetalternatefont ⋆

```
\ctexset
  {
    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 10 节　CTEX 宏集的配置文件

CTEX 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 CTEX 宏集的默认行为。

在多数情况下，并不需要修改配置文件，CTEX 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 CTEX 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 TEX 文件，但对方的操作系统或默认设置与本机不同。

与 CTEX 宏集的源代码一样，配置文件采用 LATEX3 的语法编写。

CTEX 宏集的配置文件随宏包其他文件一起安装在 TEX 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 CTEX 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 TEX Live，系统自带的配置文件就在 TEX 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 ~/texmf/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。

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

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

10.1　修改宏包默认选项

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

例 26

```
% 系统自带 ctexopts.cfg 注释中的示例语句，固定默认字体集为 windows。
% 该设置可以用在安装了 Windows 字体的非 Windows 系统中。
```

```
\ctex_set:nn { option } { fontset = windows }
```

如上例所示，宏包选项通常使用 LaTeX3 的 `\ctex_set:nn` 命令完成键值设置，第一个参数是固定的子模块 `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。

10.2　宏包载入后的配置

配置文件 `ctex.cfg` 将在宏包的末尾被载入生效。可以用它完成任意的设置，或是覆盖已有的定义。随系统安装的配置文件除版本信息外没有实际内容，注意配置文件中也使用 LaTeX3 语法。

例 27

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

例 28

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

10.3　配置标题中文翻译

由于 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。

10.4　自定义字体集

4.3 节介绍的用于 `fontset` 选项的自定义字库文件，类似于 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 11 节　第三方宏包兼容

CTEX 宏集为使 `\labelformat` 功能与章节标题设置兼容，对 cleveref 等宏包进行了补丁。由于 cleveref 长期未维护（CTAN 上仍为 0.21.4），与新版 LaTeX 内核和 hyperref 存在兼容问题，CTEX 提供了选项以控制补丁行为。

patch/cleveref ⋆ `\ctexset | patch/cleveref = true ‖ false |`

New: 2026-04-24

控制是否对 cleveref 宏包应用兼容补丁。默认为 `true`。该补丁在 cleveref 的 `\refstepcounter` 变体中注入 `\expandafter`，以确保 `\labelformat` 格式化后的标签能正确展开。

当该补丁与上游 LaTeX 内核或 hyperref 的更新产生冲突时，可以将此选项设为 `false` 以关闭补丁。

注意：当同时使用 hyperref 和 cleveref 时，`\appendix` 之后的交叉引用类型可能不正确（例如，`appendix` 退化为 `chapter`）。这是上游 LaTeX 内核 `firstaid` 对 cleveref 补丁不完整所致，与 CTEX 无关。可以使用以下方法规避：

例 29

```
\AddToHook{cmd/appendix/before}{%
  \crefalias{chapter}{appendix}%
  \crefalias{section}{subappendix}%
  \crefalias{subsection}{subsubappendix}%
  \crefalias{subsubsection}{subsubsubappendix}%
}
```

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　CTEX 0.9–CTEX 1.0d

在 2009 年在 ctex-kit 项目成立后，新增了 XeTeX 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 CTEX 宏包相对 1.02d 版本（2014/06/09）的兼容性。

第 2 版的 CTEX 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 CTEX 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。

CTEX 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 CTEX 宏包第 2 版中则完全不再支持 CCT。

下面这些是在旧版本 CTEX 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。

cs4size
c5size

分别相当于 `zihao=-4` 和 `zihao=5`，过时选项。

CCT
CCTfont

相关选项已删除。

indent
noindent

`indent` 和 `noindent` 什么也不做，过时选项。

在中文版式下，ctex 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开。CTEX 文档类的相关功能由章节标题的 `afterindent` 选项的值来确定。

zhmap
nozhmap

zhmap 宏包选项增加了参数，扩充了功能，除了支持真假值参数外，还支持选择 zhmCJK 作为底层中文处理宏包。(4.3 节)

nozhmap 选项相当于 zhmap=false。过时选项。

winfonts
adobefonts
nofonts

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 CTEX 宏包的默认字体不再是 Windows 系统字体，而是根据检测到的操作系统选择使用 Windows、Mac 的系统字体还是 Fandol 字体(4.3 节)。

punct
nopunct

旧版本中宏包 punct 选项没有参数，现在可以用参数设定标点风格(5.3 节)。原有无参形式的 punct 选项相当于 punct=quanjiao。

旧版宏包中 nopunct 选项的效果大致相当于 punct=plain。过时选项，不推荐使用。

cap
nocap

原有的 cap 和 nocap 选项由新的 scheme 选项代替。(5.3 节)

cap 选项相当于 scheme = chinese，nocap 选项相当于 scheme = plain。它们均已过时，仅因兼容性而保留。

space
nospace

新版本宏包 space 选项增加真假值参数。(5.3 节)

nospace 选项相当于 space=false，成为过时选项。

fancyhdr

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

fancyhdr 选项过时，因兼容性保留，功能是载入 fancyhdr 宏包。

hyperref

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

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

fntef

旧版本的 fntef 选项用于统一 CCTfntef 与 CJKfntef 的界面，新版本 CTEX 宏集不再支持 CCT，也不再自动载入 CJKfntef 或 xeCJKfntef 宏包，而仅在其末尾做适当格式调整。

fntef 选项过时，因兼容性保留，功能是根据引擎载入 CJKfntef (pdfTEX) 或 xeCJKfntef (XETEX) 宏包。

\CTEXunderdot
\CTEXunderline
\CTEXunderdblline
\CTEXunderwave
\CTEXsout
\CTEXxout
CTEXfilltwosides

在调用 fntef 宏包选项的同时，旧版本 CTEX 宏包由于需要支持 CCT 系统，会将以 \CJK 开头的 \CJKunderline 等宏换名为以 \CTEX 开头的 \CTEXunderline 等宏。此功能在新版本的 CTEX 宏集中已失去意义。此外，在 pdfTEX 引擎下，用于设置格式的 \CJKunderdotbasesep 等宏也被更名为 \CTEXunderdotbasesep 等宏。

在新版本中，上述由 fntef 衍生的相关命令和环境均被移除。

\CTEXsetfont

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

\CTEXindent

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

\CTEXnoindent

设置 \parindent=0pt。过时命令。

\CTEXsetup　\CTEXsetup[⟨选项⟩]{⟨标题⟩}

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

\CTEXoptions　\CTEXoptions[⟨选项⟩]

相当于设置了 \ctexset{⟨选项⟩}。过时命令。

\Chinese　\Chinese{⟨*counter*⟩}

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

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

可以使用 caption 宏包的 labelsep 选项来完成同样的功能。

例 30

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

12.3　CTEX 1.02c 以后的 SVN 开发版

CTEX 宏包在 1.02c 版本(2011/03/11)之后在 Google code 上的 SVN 开发版本, 内部版本号一直升到 1.11 版, 但从未正式发布。SVN 开发版在 1.02c 版本的基础上新增的功能在第 2 版中大多继承了过来, 但新增的命令与选项都不再保持兼容。

CTEX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

12.4　CTEX 2.2 之前的版本

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

在 CTEX 2.2 之前的版本中, beforeskip 选项的符号还用于确定章节标题后首段的缩进。当 beforeskip 是负值时, 章节标题后的第一段按英文文档的排版习惯, 没有首行缩进, 否则保留首行缩进。

这一特性在 2.2 版和后续版本中不再保留, 相应的功能通过新的 afterindent 选项来设置。如果原先设置 beforeskip 为负值, 在新版本中需要改为正值, 并设置相应的 afterindent 选项为 false。

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

在 CTEX 2.2 之前的版本中, 对于 \section 级以下标题, afterskip 选项的符号用于确定标题与随后正文是否排在同一段。如果是正值, 则正文另起一段, 否则标题与随后正文排在同一段, afterskip 的绝对值给出水平间距。

这一特性在 2.2 版和后续版本中不再保留, 相应的功能通过新的 runin 选项来设置。如果原先设置 afterskip 为负值, 在新版本中需要改为正值, 并设置相应的 runin 选项为 true。

12.5　CTEX 2.4.1 和 2.4.2

part/fixbeforeskip
chapter/fixbeforeskip

这两个选项已经被删除, 相应功能由新的选项 fixskip 提供。

12.6　CTEX 2.5 之前的版本

CTEX 2.5 有一些比较大的变动。

`UTF8` `GBK` (pdf)LATEX 格式下，文档编码初始值统一设置成 UTF-8。因此，仍旧使用 GBK 编码的文档，需要在文档类或宏包选项中显式指定 `GBK`。

`\CTEXunderdot` `\CTEXunderline` `\CTEXunderdblline` `\CTEXunderwave` `\CTEXsout` `\CTEXxout` `CTEXfilltwosides` 不再默认载入 CJKfntef 或 xeCJKfntef 宏包，同步移除有关命令和环境。若需使用相关宏包，建议用户使用 `\usepackage` 命令主动载入。

`fntef` 作为兼容性保留，会视编译引擎载入相应宏包。

`windows` 不再支持 Windows XP 系统，默认要求 Windows 系统有微软雅黑字体。建议 Windows XP 系统的用户及时更新操作系统。若一定要在 Windows XP 中使用，请使用 2.5 以前的版本。

`windowsold` `windowsnew` 过时字库选项，作为兼容性保留，功能是载入 `windows` 字库。

`ubuntu` 改用思源（Noto CJK）和文鼎字库。该字库不再支持 pdfLATEX编译。

`AlternateFont` 不再支持将替代字体的可选项放在字体名之前的方括号中，新的语法是将可选项放在字体名之后的花括号之内。

除了以上列出的选项以外，当用户使用 CTEX 系列文档类，且使用 LATEX 或 upLATEX 编译时，若用户没有在文档类选项中显式指定 `dvips`/`dvipdfmx`/`dvisvgm` 等驱动选项，则文档类指定默认驱动为 DVIPDFMx。

第 13 节　宏集依赖情况与手工安装方法

本节介绍 CTEX 宏集的依赖情况，并介绍手工编译安装的具体方法。通常用户只需参照第 2.2 节介绍的方法，使用发行版自带的包管理器安装本宏集。

CTEX 宏集有两个源文件：`ctex.dtx`、`ctexpunct.spa`。使用不同的编译方式时，CTEX 依赖的宏包略有不同。在手工安装 CTEX 宏集之前，请确保你的 TEX 发行版中已经正确安装了这些宏包。CTEX 依赖宏包的详情叙述如下：

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 宏集。
- indentfirst 宏包，属于 tools 宏集。
- zhnumber 宏包。

➥ 以上是各种编译方式都必需的依赖项。

- CJK 宏集。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包，它还依赖
  - iftex 宏包。

  - ltxcmds 宏包。
  - kvoptions 宏包。
  - kvsetkeys 宏包。
  - keyval 宏包, graphics 宏集。

➥ 以上是使用 pdfLaTeX 或 LaTeX + DVIPDFMx 的编译方式所需要的依赖项, 其中 zhmCJK 是可选的。

- xeCJK 宏集, 它还依赖
  - xtemplate 宏包, 它属于 l3packages 宏集。
  - fontspec 宏包。

➥ 以上是使用 XeLaTeX 编译时的依赖项。

- luatexja 宏包, 它还依赖
  - adobemapping 宏包。
  - luaotfload 宏包, 它还依赖 lualibs 宏包。
  - luatexbase 宏包, 它还依赖 ctablestack 宏包。
  - atbegshi 宏包。
  - etoolbox 宏包。
  - iftex 宏包。
  - infwarerr 宏包。
  - ltxcmds 宏包。
  - pdftexcmds 宏包。
  - xkeyval 宏包。
  - chinese-jfm 宏包。
- fontspec 宏包。

➥ 以上是使用 LuaLaTeX 编译时的依赖项。

- zhmetrics-uptex 宏包。

➥ 以上是使用 upLaTeX 编译时的依赖项。

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

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

其他细节, 可参照其 宏包手册 中第 3 节的指导。

CTeX 宏集已被 TeX Live 和 MiKTeX 收录, 若无特别理由, 我们强烈建议用户使用包管理器安装本宏集。

若要手工安装, 请遵循如下步骤:

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

第 14 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)
- 曾祥东 (xdzeng96@gmail.com)
- 李泽平 (zepinglee@gmail.com)
- 周宇恺 (muzimuzhi@gmail.com)
- 张瑞熹 (ruixizhang42@gmail.com)

参考文献

[1] Donald Ervin Knuth. *The TEXbook*, *Computers & Typesetting*, volume A. Addison-Wesley, 1986

[2] Frank Mittelbach and Michel Goossens. *The LATEX Companion*. Tools and Techniques for Computer Typesetting. Boston: Addison-Wesley, second edition, 2004

第 15 节　代码实现

```
⟨@@=ctex⟩

⟨*class|style⟩
\cs_if_exist:NF \NewDocumentCommand
  { \RequirePackage { xparse } }
⟨class⟩   \prop_gput:Nnn \g_msg_module_type_prop { ctex } { Class }
⟨article⟩ \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexart }
⟨book⟩    \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexbook }
⟨report⟩  \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexrep }
⟨beamer⟩  \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexbeamer }
⟨ctexsize⟩ \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexsize }
⟨ctexheading⟩ \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexheading }
⟨/class|style⟩

⟨*class|ctex⟩
```

检查 expl3 的版本。

```
\msg_new:nnnn { ctex } { l3-too-old }
  { Support~package~`#1'~too~old. }
  {
    Please~update~an~up-to-date~version~of~the~bundles\\\\
    `l3kernel'~and~`l3packages'\\\\
    using~your~TeX~package~manager~or~from~CTAN.
  }
\@ifpackagelater { expl3 } { 2025/10/09 } { }
  { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
```

`\c__ctex_engine_str` `\c__ctex_engine_file_str` 引擎检查。目前 LaTeX3 将 ApTeX 识别为 upTeX。

```
\str_const:Ne \c__ctex_engine_str
  { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }
\msg_new:nnnn { ctex } { engine-not-supported }
  { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }
  { You~can~switch~to~xelatex,~lualatex,~pdflatex,~uplatex,~or~aplatex. }
\file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }
  {
    \str_const:Ne \c__ctex_engine_file_str
      { ctex-engine- \c__ctex_engine_str .def }
  }
  { \msg_critical:nne { ctex } { engine-not-supported } { \c__ctex_engine_str } }

⟨/class|ctex⟩

⟨*class|ctex|ctexheading|ctexsize⟩
```

ctexsize 也要载入 fix-cm 包解决传统 cm 字体字号缺失的问题。

```
⟨!ctexsize⟩   \RequirePackage { ctexhook , ctexpatch }
⟨!ctexheading⟩ \RequirePackage { fix-cm }
```

宏包载入检查。

```
⟨*class|ctex⟩
⟨*class⟩
\ctex_disable_package:n { ctex }
\ctex_disable_package:n { ctexcap }
⟨/class⟩
\ctex_disable_package:n { ctexsize }
\ctex_disable_package:n { ctexheading }
⟨/class|ctex⟩
```

`\c__ctex_everysel_loaded_bool` LaTeX 2021-06-01 以后的版本内建了 everysel 包的功能。

```
⟨*!ctexsize⟩
```

```
⟨*!ctexheading⟩
\ctex_if_format_at_least:nTF { 2021/06/01 }
  { \bool_const:Nn \c__ctex_everysel_loaded_bool { \c_false_bool } }
  {
    \RequirePackage { everysel }
    \bool_const:Nn \c__ctex_everysel_loaded_bool { \c_true_bool }
  }
⟨/!ctexheading⟩
```

15.1　内部函数与变量

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_box
\l__ctex_tmp_dim
⟨!ctexheading⟩

临时变量。

```
\tl_clear_new:N \l__ctex_tmp_tl
\int_new:N \l__ctex_tmp_int
\box_new:N \l__ctex_tmp_box
\dim_new:N \l__ctex_tmp_dim
```

\ctex_define_option:n
\ctex_define:n
\ctex_set:n
\ctex_set:nn

在宏包内部使用的键值选项定义、设置命令。

```
⟨/!ctexsize⟩
\cs_new_protected:Npn \ctex_define_option:n
  { \keys_define:nn { ctex / option } }
⟨*!ctexsize⟩
\cs_new_protected:Npn \ctex_define:n
  { \keys_define:nn { ctex } }
\cs_new_protected:Npn \ctex_set:n
  { \keys_set:nn { ctex } }
\cs_new_protected:Npn \ctex_set:nn #1
  { \keys_set:nn { ctex / #1 } }
```

\ctex_scheme_input:n

输入 scheme 文件。先查找当前文档类下的 ⟨*scheme*⟩, 找不到再查找一般的文件。

```
\cs_new_protected:Npn \ctex_scheme_input:n #1
  {
    \ctex_push_file:
      \tl_if_exist:NTF \c__ctex_class_tl
        {
          \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
            { \file_input:n  { ctex-scheme- #1 .def } }
        }
        { \file_input:n  { ctex-scheme- #1 .def } }
    \ctex_pop_file:
  }
\cs_generate_variant:Nn \ctex_scheme_input:n { o }
```

\g__ctex_section_depth_int

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。

```
⟨*!beamer⟩
\int_new:N \g__ctex_section_depth_int
\int_gset:Nn \g__ctex_section_depth_int { 2 }
⟨/!beamer⟩

⟨/!ctexsize⟩
⟨/class|ctex|ctexheading|ctexsize⟩
⟨*class|ctex⟩
```

对旧版本的宏包给出错误信息。

```
\msg_new:nnnn { ctex } { package-too-old }
  { Support~package~`#1'~too~old. }
  {
```

```
    Please~update~an~up-to-date~version~of~the~package~`#1'\\
    using~your~TeX~package~manager~or~from~CTAN.
  }
```

\ifctexpdf 在 zhmetrics 映射文件中使用。

```
\sys_if_output_pdf:TF
  { \cs_new_eq:NN \ifctexpdf \if_true: }
  { \cs_new_eq:NN \ifctexpdf \if_false: }
```

\ctex_if_preamble:TF 测试是否在 LaTeX 2_ε 的导言区。在宏包内部初始为真，文档最开始位置再设置为假。注意，钩子 \ctex_after_end_preamble:n 在 \AtBeginDocument 之后执行，可以与 \@onlypreamble 的行为一致。

```
\cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
\ctex_after_end_preamble:n { \cs_set_eq:NN \ctex_if_preamble:TF \use_ii:nn }
```

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

```
\cs_new_protected:Npn \ctex_set_default_ccwd:Nn #1#2
  { \tl_set:Ne #1 { \__ctex_default_ccwd_aux:n {#2} } }
\cs_new:Npn \__ctex_default_ccwd_aux:n #1
  {
    \exp_not:n {#1}
    \exp_after:wN \__ctex_default_ccwd_aux:w
      \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop
  }
\exp_last_unbraced:NNNNo
  \cs_new:Npn \__ctex_default_ccwd_aux:w #1 { \tl_to_str:n { pt } } #2 \q_stop
    { \tl_if_empty:nT {#2} { \ccwd } }
```

\g__ctex_encoding_tl 所有引擎下默认编码均设为 UTF-8，初始值为空，\ProcessKeysOptions 再判断。

```
\tl_new:N \g__ctex_encoding_tl
```

\g__ctex_zhmCJK_bool 是否使用 zhmCJK 宏包。

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

\l__ctex_autoindent_tl 保存 autoindent 选项的值，空值表示不自动调整首行缩进。

```
\tl_new:N \l__ctex_autoindent_tl
```

\ctex_if_autoindent_touched:F 检查 autoindent 选项是否被用户设置。

```
\cs_new_eq:NN \ctex_if_autoindent_touched:F \use:n
```

\ctex_zhmap_case:nnn 参数 #1 是 zhmCJK 的内容，#2 是 zhmetrics。

```
\cs_new_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
```

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

```
⟨class⟩ \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfClass }
⟨ctex⟩  \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfPackage }
```

\ctex_load_std_class:n 保存传递给标准文档类的选项。使用 \PassOptionsToClass 是为了预防可能存在的选项冲突，
\g__ctex_std_options_clist 选项列表展开后再传递。

```
⟨*class⟩
\cs_new_protected:Npn \ctex_load_std_class:n #1
```

```
  {
    \tl_const:Nn \c__ctex_class_tl {#1}
    \exp_args:No \PassOptionsToClass
      { \g__ctex_std_options_clist }
      {#1}
    \LoadClass {#1}
  }
\clist_new:N \g__ctex_std_options_clist
```
⟨/class⟩

对无效选项给出警告。

```
\msg_new:nnn { ctex } { invalid-option }
  { Option~`\l_keys_key_str'~is~invalid~in~current~mode. }
\msg_new:nnn { ctex } { invalid-value }
  { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_str'. }
\msg_new:nnn { ctex } { CJKecglue-unsupported }
  { `experiment/CJKecglue'~is~not~supported~by~the~current~engine. }
```

\ctex_deprecated_option:nn
\ctex_set_deprecated_option:n
\ctex_deprecated_command:Nn

对过时选项或命令给出警告。

```
\cs_new_protected:Npn \ctex_deprecated_option:n
  { \msg_warning:nnn { ctex } { deprecated-option } }
\cs_new_protected:Npn \ctex_set_deprecated_option:n #1
  {
    \ctex_deprecated_option:n { Option~`#1'~is~set. }
    \ctex_set:nn { option } {#1}
  }
\cs_new_protected:Npn \ctex_deprecated_command:Nn #1#2
  {
    \msg_warning:nnee { ctex } { deprecated-command }
      { \token_to_str:N #1 } { \exp_not:n {#2} }
  }
\msg_new:nnn { ctex } { deprecated-option }
  { Option~`\l_keys_key_str'~is~deprecated.\\ #1 }
\msg_new:nnn { ctex } { deprecated-command }
  { Command~`#1'~is~deprecated.\\ #2 }
```
⟨/class|ctex⟩

\g__ctex_font_size_int

0 表示修改默认字体大小为五号, 1 为小四号, 大于 1 则不作修改。初始值 -1 表示 zihao 选项未初始化, 会在将来根据文档类决定初值。

```
⟨*class|ctex|ctexsize⟩
\int_new:N \g__ctex_font_size_int
\int_gset:Nn \g__ctex_font_size_int { -1 }
⟨/class|ctex|ctexsize⟩
```

15.2　宏包选项

```
⟨*class|style⟩
\ctex_define_option:n
  {
⟨/class|style⟩
```

experiment/font-size-system

选择字号系统。预设有 word 和 letterpress 两种字号系统, 用户也可以自定义。

```
⟨*class|ctex|ctexsize⟩
    experiment / font-size-system .tl_gset:N = \g__ctex_font_size_system_tl ,
    experiment / font-size-system .initial:n = { word } ,
    experiment / font-size-system .value_required:n = true ,
⟨/class|ctex|ctexsize⟩
```

zihao
```
⟨*class|ctex|ctexsize⟩
    zihao .choice: ,
    zihao .value_required:n = true ,
    zihao /     5  .code:n = { \int_gset:Nn \g__ctex_font_size_int { 0 } } ,
    zihao /    -4  .code:n = { \int_gset:Nn \g__ctex_font_size_int { 1 } } ,
    zihao / false  .code:n = { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
⟨ctexsize⟩  }
⟨/class|ctex|ctexsize⟩
⟨*class|ctex⟩
    c5size  .code:n = { \ctex_set_deprecated_option:n { zihao =  5 } } ,
    cs4size .code:n = { \ctex_set_deprecated_option:n { zihao = -4 } } ,
    c5size  .value_forbidden:n = true ,
    cs4size .value_forbidden:n = true ,
```

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

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

autoindent 自动调整段落的首行缩进功能。

```
    autoindent .choice: ,
    autoindent .default:n = { true } ,
    autoindent / true    .code:n =
      {
        \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
        \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
      } ,
    autoindent / false   .code:n =
      {
        \tl_clear:N \l__ctex_autoindent_tl
        \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
      } ,
    autoindent / unknown .code:n =
      {
        \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
        \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
      } ,
```

indent 仅为兼容性保留，已过时。

```
    indent .code:n =
      {
        \ctex_deprecated_option:n
          {
            The~functionality~has~been~removed.\\
            It's~better~to~set~the~heading~styles~via~`afterindent'~option.
          }
      } ,
    indent .value_forbidden:n = true ,
    noindent .code:n =
      {
        \ctex_deprecated_option:n
          {
            The~functionality~has~been~removed.\\
            It's~better~to~set~the~heading~styles~via~`afterindent'~option.
          }
      } ,
    noindent .value_forbidden:n = true ,
```

GBK
UTF8 文档编码，默认为 UTF-8。

```
    GBK  .code:n =
```

```
    {
      \sys_if_engine_pdftex:TF
        { \tl_gset:Nn \g__ctex_encoding_tl { GBK } }
        {
          \msg_warning:nn { ctex } { invalid-option }
          \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
        }
    } ,
  UTF8 .code:n = { \tl_gset:Nn \g__ctex_encoding_tl { UTF8 } } ,
  GBK  .value_forbidden:n = true ,
  UTF8 .value_forbidden:n = true ,
```

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

```
  fontset    .tl_gset:N = \g__ctex_fontset_tl ,
  nofonts    .code:n = { \ctex_set_deprecated_option:n { fontset = none } } ,
  adobefonts .code:n = { \ctex_set_deprecated_option:n { fontset = adobe } } ,
  winfonts   .code:n = { \ctex_set_deprecated_option:n { fontset = windows } } ,
  nofonts    .value_forbidden:n = true ,
  winfonts   .value_forbidden:n = true ,
  adobefonts .value_forbidden:n = true ,
```

zhmap

```
  zhmap .choice: ,
  zhmap .default:n = { true } ,
  zhmap / zhmCJK .code:n =
    {
      \bool_gset_true:N \g__ctex_zhmCJK_bool
      \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
    } ,
  zhmap / true   .code:n =
    {
      \bool_gset_false:N \g__ctex_zhmCJK_bool
      \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
    } ,
  zhmap / false  .code:n =
    {
      \bool_gset_false:N \g__ctex_zhmCJK_bool
      \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
    } ,
  nozhmap   .code:n =
    { \ctex_set_deprecated_option:n { zhmap = false } } ,
  nozhmap   .value_forbidden:n = true ,
```

punct 设置标点符号输出格式。

```
  punct   .tl_set:N = \l__ctex_punct_tl ,
  punct  .default:n = { quanjiao } ,
  punct  .initial:n = { quanjiao } ,
  nopunct   .code:n = \ctex_set_deprecated_option:n { punct = plain } ,
  nopunct   .value_forbidden:n = true ,
```

space

```
  space .choices:nn =
    { true , auto , false }
    { \ctex_at_end:n { \ctex_set:n { space = #1 } } } ,
  space  .default:n = { true } ,
  nospace   .code:n = { \ctex_deprecated_option:nn { space = false } } ,
  nospace   .value_forbidden:n = true ,
```

heading

```
  heading .bool_set:N = \l__ctex_heading_bool ,
```

```
⟨/class|ctex⟩
⟨*class|ctex|ctexheading⟩
```

sub3section
sub4section
```
⟨*!beamer⟩
    sub3section .code:n =
      { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
    sub4section .code:n =
      { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
    sub3section .value_forbidden:n = true ,
    sub4section .value_forbidden:n = true ,
⟨/!beamer⟩
```

scheme
```
    scheme .tl_set:N  = \l__ctex_scheme_tl ,
⟨*ctexheading⟩
    scheme .default:n = { plain } ,
    scheme .initial:n = { plain }
  }
⟨/ctexheading⟩
⟨*!ctexheading⟩
    scheme .default:n = { chinese } ,
    scheme .initial:n = { chinese } ,
⟨/!ctexheading⟩

⟨/class|ctex|ctexheading⟩
⟨*class|ctex⟩
```

cap
nocap

cap 和 nocap 是过时选项。

```
    cap    .code:n = { \ctex_set_deprecated_option:n { scheme = chinese } } ,
    nocap  .code:n = { \ctex_set_deprecated_option:n { scheme = plain } } ,
    cap    .value_forbidden:n = true ,
    nocap  .value_forbidden:n = true ,
```

以下三项都是过时的兼容选项, 它们会载入有关宏包。

fntef
```
    fntef .code:n =
      {
        \sys_if_engine_xetex:TF
          {
            \ctex_deprecated_option:n { `xeCJKfntef'~package~is~loaded. }
            \ctex_at_end:n { \RequirePackage { xeCJKfntef } }
          }
          {
            \sys_if_engine_pdftex:TF
              {
                \ctex_deprecated_option:n { `CJKfntef'~package~is~loaded. }
                \ctex_at_end:n { \RequirePackage { CJKfntef } }
              }
              {
                \ctex_deprecated_option:n
                  { Furthermore,~option~`fntef'~is~invalid~in~current~mode. }
              }
          }
      } ,
```

fancyhdr
```
    fancyhdr .code:n =
      {
        \ctex_deprecated_option:n { `fancyhdr'~package~is~loaded. }
        \ctex_at_end:n { \RequirePackage { fancyhdr } }
      } ,
```

hyperref
```
    hyperref .code:n =
      {
        \ctex_deprecated_option:n { `hyperref'~package~will~be~loaded. }
        \ctex_at_end:n
          {
            \cs_if_exist:NF \hypersetup
              { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
          }
```

```
        \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
      } ,
  }
```

```
⟨/class|ctex⟩
⟨*class|ctex|ctexsize⟩
```

10pt 11pt 12pt 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传给标准文档类。

```
\tl_clear_new:N \l__ctex_tmp_tl
\clist_map_inline:nn
  {
    10pt , 11pt , 12pt ,
     8pt ,  9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
  }
  {
    \tl_put_right:Nn \l__ctex_tmp_tl
      {
        #1 .code:n =
⟨*!class⟩
          { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
⟨/!class⟩
⟨*class⟩
          {
            \int_gset:Nn \g__ctex_font_size_int { 2 }
            \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
          } ,
⟨/class⟩
        #1 .value_forbidden:n = true ,
      }
  }
\exp_args:No \ctex_define_option:n { \l__ctex_tmp_tl }
\tl_clear:N \l__ctex_tmp_tl
```

将未知选项传给标准文档类。

```
⟨*class⟩
\ctex_define_option:n
  {
    unknown .code:n =
      { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
  }
⟨/class⟩
```

载入选项配置文件。

```
⟨!ctexsize⟩ \ctex_file_input:n { ctexopts.cfg }
⟨/class|ctex|ctexsize⟩
```

处理宏包选项。

```
⟨*class|style⟩
\cs_if_exist:NTF \ProcessKeyOptions
  { \ProcessKeyOptions [ ctex / option ] }
  {
    \RequirePackage { l3keys2e }
    \ProcessKeysOptions { ctex / option }
  }
⟨/class|style⟩
```

pdfLaTeX 下, 如果没有显式指定编码为 UTF8, 则给出警告信息。

```
⟨*class|ctex⟩
\msg_new:nnn { ctex } { pdftex-utf8 }
```

```
  { UTF8~will~be~used~as~the~default~encoding. }
\tl_if_empty:NT \g__ctex_encoding_tl
  {
    \sys_if_engine_pdftex:T
      { \msg_warning:nn { ctex } { pdftex-utf8 } }
    \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
  }
⟨/class|ctex⟩

⟨*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。

```
\int_case:nn { \g__ctex_font_size_int }
  {
    { 0 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt } }
    { 1 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt } }
  }
```

载入标准文档类。

```
⟨*article⟩
\ctex_load_std_class:n { article }
⟨/article⟩
⟨*book⟩
\ctex_load_std_class:n { book }
⟨/book⟩
⟨*report⟩
\ctex_load_std_class:n { report }
⟨/report⟩
⟨*beamer⟩
\ctex_load_std_class:n { beamer }
⟨/beamer⟩

⟨/class⟩
```

15.3　特定引擎支持与设置

15.3.1　ctexbackend.cfg

对于 XeLaTeX/pdfLaTeX/LuaLaTeX 等默认直接输出 PDF 的编译方式，用户无需为涉及驱动的宏包指定驱动选项。对于 LaTeX 和 upLaTeX 等默认不直接输出 PDF 的编译方式，用户则需要指定驱动选项。

由于历史遗留问题，在使用 LaTeX 或 upLaTeX 等编译时，大多数涉及驱动的宏包选定的默认输出驱动都是 Dvips。考虑当前实际使用频率，以及考虑到 CTeX 宏集对中文支持的默认方式，我们在用户使用 CTeX 系列文档类时，将默认的输出驱动改为 DVIPDFMx。

具体来说，如果 dvips，dvipdfmx，dvisvgm 等驱动没有在文档类的全局选项中被明确指定，我们就在 \@classoptionslist 开头加入 dvipdfmx。

本段代码只在 ctexart 等文档类开头载入，不在 ctex 中使用。并且需要放在 expl3 之前载入，保证它载入正确的 backend 文件。

```
⟨*backend⟩
\begingroup
\expandafter\ifx\csname Umathchardef\endcsname\relax
\else\expandafter\endgroup\expandafter\endinput\fi
  \ifodd
    \expandafter\ifx\csname      pdfoutput\endcsname\relax
    \expandafter\ifx\csname enablecjktoken\endcsname\relax 0\else 1\fi
                                    \else\ifnum\pdfoutput>0 0\else 1\fi\fi\space
    \def\x#1{%
```

```
      \if\relax\detokenize{#1}\relax
        \gdef\@classoptionslist{dvipdfmx}%
      \else
        \let\CTEX@add\@ne
        \@tfor\x:={dvips}{dvipdfmx}{dvisvgm}\do{%
          \expandafter\in@\expandafter{\expandafter,\x,}{,#1,}%
          \ifin@ \let\CTEX@add\tw@ \@break@tfor \fi}%
        \ifodd\CTEX@add \gdef\@classoptionslist{dvipdfmx,#1}\fi
      \fi}
    \expandafter\x\expandafter{\@classoptionslist}
  \fi
\endgroup
⟨/backend⟩
```

15.3.2 ctex-engine-pdftex.def

\ctex_set_zhmap:n 设置 upTEX 字体映射, 同时作用于 \AtBeginDvi 与 \AtBeginShipoutFirst。该宏对 pdfTEX 和 upTEX 均有用。\AtBeginDvi 直接将 \special 保存到盒子中, \AtBeginShipoutFirst 是保存到到宏中, 并且不展开参数。

可以使用 LATEX 2020/10/01 的钩子机制来统一设置。

```
⟨*pdftex|uptex|aptex⟩
\cs_new_protected:Npn \ctex_set_zhmap:n
  { \tl_gput_right:Ne \g__ctex_zhmap_tl }
\cs_new_protected:Npn \ctex_use_zhmap:
  { \tl_use:N \g__ctex_zhmap_tl }
\cs_if_exist:NTF \ctex_gadd_ltxhook:nn
  {
    \cs_new_protected:Npn \ctex_at_shipout_first:n
      { \ctex_gadd_ltxhook:nn { shipout/firstpage } }
    \ctex_at_shipout_first:n { \ctex_use_zhmap: }
  }
  {
    \cs_new_protected:Npn \ctex_add_dvi_zhmap:
      { \AtBeginDvi { \ctex_use_zhmap: } }
    \ctex_after_end_preamble:n { \ctex_add_dvi_zhmap: }
    \ctex_at_end_package:nn { atbegshi }
      {
        \cs_new_protected:Npn \ctex_at_shipout_first:n
          { \AtBeginShipoutFirst }
        \ctex_at_shipout_first:n { \ctex_use_zhmap: }
        \cs_gset_eq:NN \ctex_add_dvi_zhmap: \prg_do_nothing:
      }
  }
\tl_new:N \g__ctex_zhmap_tl
\@onlypreamble \ctex_set_zhmap:n
⟨/pdftex|uptex|aptex⟩

⟨*pdftex⟩
```

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

```
\seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
  { C19 , C10 , C00 , C09 , C40 , C60 }
```

\ctex_family_cmap:nn 在 \DeclareFontFamily 的 ⟨*loading-settings*⟩ 中给 CJK 字体族加上 CMap。

```
\cs_new_protected:Npn \ctex_family_cmap:nn #1#2
  {
    \cs_if_free:cF { #1 + #2 }
      {
        \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
          { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
```

```
      }
  }
\cs_generate_variant:Nn \ctex_family_cmap:nn { ee }
\cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:ee
```

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

```
\cs_new_protected:Npn \ctex_add_cmap:n #1
  {
    \cs_if_free:NF \CJK@plane
      { \exp_args:Ne \__ctex_add_cmap_auxi:n { #1 \CJK@plane } }
  }
\cs_new_protected:Npn \__ctex_add_cmap_auxi:n #1
  { \exp_args:Nc \__ctex_add_cmap_auxii:Nn { __ctex_add_cmap_ #1 : } {#1} }
\cs_new_protected:Npn \__ctex_add_cmap_auxii:Nn #1#2
  {
    \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
    #1
  }
\cs_new_protected:Npn \__ctex_save_cmap:Nn #1#2
  {
    \exp_args:Ne \file_get_full_name:nNTF
      { \str_lowercase:n {#2} .cmap } \l__ctex_cmap_file_tl
      {
        \pdf_object_unnamed_write:ne
          { fstream }
          { { } { \l__ctex_cmap_file_tl } }
        \cs_new_protected:Npx #1
          {
            \pdfnobuiltintounicode \tex_font:D
            \tex_pdffontattr:D \tex_font:D
              { /ToUnicode ~ \pdf_object_ref_last: }
          }
      }
      { \cs_new_eq:NN #1 \prg_do_nothing: }
  }
\tl_new:N \l__ctex_cmap_file_tl
```

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

```
\sys_if_output_pdf:T
  {
    \exp_args:Nno \use:n
      { \cs_gset_protected:Npn \DeclareFontFamily #1#2#3 }
      {
        \DeclareFontFamily {#1} {#2} {#3}
        \CTEX@Family@CMap  {#1} {#2}
      }
    \ctex_at_end_package:nn { cmap }
      {
        \cs_gset_eq:NN \ctex_add_cmap:n \use_none:n
        \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn
      }
  }
```

首先检查选项, 决定是否载入 zhmCJK 宏包。

```
\bool_if:NTF \g__ctex_zhmCJK_bool
  {
    \exp_args:Ne \PassOptionsToPackage
      { encoding = \g__ctex_encoding_tl }
      { zhmCJK }
    \RequirePackage { zhmCJK }
  }
```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```
{
  \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
    { \RequirePackage { CJK } }
    { \RequirePackage { CJKutf8 } }
  \RequirePackage { CJKpunct , CJKspace }
```

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

```
  \cs_new_protected:Npn \ctex_load_zhmap:nnnn #1#2#3#4
    {
      \tl_set:Nn \CJKrmdefault {#1}
      \tl_set:Nn \CJKsfdefault {#2}
      \tl_set:Nn \CJKttdefault {#3}
      \ctex_set_zhmap:n { \ctex_file_input:n { ctex-zhmap- #4 .tex } }
    }
  \@onlypreamble \ctex_load_zhmap:nnnn
}
```

\ctex_undeclare_unicode_character:n LaTeX 2022-06-01 将通常用作中文书名号的 U+3008 和 U+3009 分别定义为 \textlangle 和 \textrangle, 这个定义会被 CJKutf8 包优先使用。我们总是使用中文书名号, 需要取消 LaTeX 的定义, 作用是局部的, 不使用 \cs_undefine:N。

```
\cs_new_protected:Npn \ctex_undeclare_unicode_character:n #1
  { \cs_set_eq:cN { u8 : \codepoint_str_generate:n {#1} } \tex_undefined:D }
\CJKaddEncHook { UTF8 }
  {
    \ctex_undeclare_unicode_character:n { "3008 }
    \ctex_undeclare_unicode_character:n { "3009 }
  }
```

\ctex_CJK_input:n
\CJK@input 载入 CJK 包的 .enc 和 .bdg 等文件时, 需要设置 \endlinechar 为 −1。

```
\cs_new_protected:Npn \ctex_CJK_input:n #1
  {
    \ctex_push_file:
      \int_set:Nn \tex_endlinechar:D { -1 }
      \file_input:n {#1}
    \ctex_pop_file:
  }
\cs_set_eq:NN \CJK@input \ctex_CJK_input:n
```

\ctex_plane_to_utfxvibe:Nn
\CJK@surr fancyhdr 宏包的 \nouppercase 会将 \uppercase 定义为 \relax, 而 \CJK@surr 需要用它将 \CJK@plane 转化成大写字母, 这就造成了冲突[18]。我们在这里给出 \CJK@surr 的一个不依赖 \uppercase 的实现。

```
\cs_if_free:NF \CJK@surr
  {
    \cs_new_protected:Npn \ctex_plane_to_utfxvibe:Nn #1#2
      {
        \int_set:Nn \l__ctex_tmp_int
          { \exp_args:Ne \int_from_hex:n {#2} }
        \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
          { \tl_gset:Ne #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
          {
            \int_sub:Nn \l__ctex_tmp_int { 256 }
            \tl_gset:Ne #1
              {
                \int_to_Hex:n
                  { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
```

[18] https://github.com/CTeX-org/ctex-kit/issues/146

```
                \int_to_Hex:n
                  { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
              }
          }
      }
    \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
  }
```

\CJK@addcmap LaTeX 2021-06-01 默认载入 glyphtounicode.tex, 我们对 CJK 字体禁用这一内建设置。

```
\cs_if_free:NF \CJK@addcmap
  {
    \ctex_patch_cmd:Nnn \CJK@addcmap
      { \pdffontattr \font@name }
      {
        \pdfnobuiltintounicode \font@name
        \pdffontattr \font@name
      }
  }
```

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

```
\AtBeginDocument
  {
    \use:e
      {
        \str_if_eq:nnF { \l__ctex_punct_tl } { quanjiao }
          { \exp_not:N \punctstyle { \l__ctex_punct_tl } }
      }
  }
```

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

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

从 LaTeX 2018-04-01 开始, 传统引擎的默认编码被设置成 UTF-8, 汉字的首字节已经被设置成活动字符, 我们无需再使用 \CJK@makeActive 设置。\CJK@envStart 的定义是

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

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

```
\bool_if:NF \g__ctex_zhmCJK_bool
```

```
  {
    \str_if_eq:onTF { \g__ctex_encoding_tl } { UTF8 }
      { \CJK@loadBinding { UTF8 } }
      { \CJK@loadBinding { standard } }
    \exp_args:Ne \ctex_at_end_preamble:n
      {
        \exp_not:N \CJK@envStart
          { } { \g__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
        \exp_not:N \CJKtilde
      }
  }
```

\ctex_auto_ignorespaces: 保存 \CJK@@ignorespaces 的定义, 方便使用。

```
\cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces
```

\CJK@X
\CJK@XX
\CJK@XXX
\CJK@XXXX

最新 LaTeX 中的 \MakeUppercase 等大小写转化命令被定义为 LaTeX3 中对应的 \text_uppercase:n 等函数。它会先将参数用 \text_expand:n 逐步展开,这与 CJK 包的定义不兼容。可以有多种方法解决这个冲突, 最简单直接的方法是将首字节的活动字符用 \protected 来定义, 但可能有一定的兼容性问题。保险起见, 我们采用将 \CJK@X 等内部宏转化为 \protected 宏的方式, 并将参数提前由字符转化为数字, 避免在移动参数中被进一步展开。

```
\cs_new:Npn \__ctex_char:N #1
  { \exp_args:Ne \CTEX@char@n { \int_value:w `#1 } }
\cs_new:Npn \__ctex_char:NN #1#2
  {
    \token_if_eq_meaning:NNTF #2 \protect
      { \__ctex_char_aux:NN #1 }
      { \__ctex_char_aux:NN #1#2 }
  }
\cs_new:Npn \__ctex_char_aux:NN #1#2
  {
    \exp_last_unbraced:Ne \CTEX@char@nn
      {
        { \int_value:w `#1 }
        { \int_value:w `#2 }
      }
  }
\cs_new:Npn \__ctex_char:NNN #1#2#3
  {
    \token_if_eq_meaning:NNTF #2 \protect
      { \__ctex_char_auxi:NNNN #1#3 }
      { \__ctex_char_auxii:NNN #1#2#3 }
  }
\cs_new:Npn \__ctex_char_auxi:NNNN #1#2#3#4
  { \__ctex_char_auxii:NNN #1#2#4 }
\cs_new:Npn \__ctex_char_auxii:NNN #1#2#3
  {
    \exp_last_unbraced:Ne \CTEX@char@nnn
      {
        { \int_value:w `#1 }
        { \int_value:w `#2 }
        { \int_value:w `#3 }
      }
  }
\cs_new:Npn \__ctex_char:NNNN #1#2#3#4
  {
    \token_if_eq_meaning:NNTF #2 \protect
      { \__ctex_char_auxi:NNNNN #1#3 }
      { \__ctex_char_auxii:NNNN #1#2#3#4 }
  }
\cs_new:Npn \__ctex_char_auxi:NNNNN #1#2#3#4#5
  { \__ctex_char_auxii:NNNN #1#2#3#5 }
\cs_new:Npn \__ctex_char_auxii:NNNN #1#2#3#4
```

```
  {
    \exp_last_unbraced:Ne \CTEX@char@nnnn
      {
        { \int_value:w `#1 }
        { \int_value:w `#2 }
        { \int_value:w `#3 }
        { \int_value:w `#4 }
      }
  }
\cs_gset_eq:NN \CJK@X    \__ctex_char:N
\cs_gset_eq:NN \CJK@XX   \__ctex_char:NN
\cs_gset_eq:NN \CJK@XXX  \__ctex_char:NNN
\cs_gset_eq:NN \CJK@XXXX \__ctex_char:NNNN
```

CJKutf8 包优先使用 `\u8:xxx`/`\u8:xxxx` 的 LaTeX 定义。

```
\str_if_eq:onF { \g__ctex_encoding_tl } { GBK }
  {
    \exp_args:Nne \use:n
      { \cs_gset:Npn \__ctex_char_aux:NN #1#2 }
      {
        \exp_not:N \cs_if_exist_use:cF
          { u8: \exp_not:N \tl_to_str:n { #1#2 } }
          { \exp_not:o { \__ctex_char_aux:NN {#1} {#2} } }
      }
    \exp_args:Nne \use:n
      { \cs_gset:Npn \__ctex_char_auxii:NNN #1#2#3 }
      {
        \exp_not:N \cs_if_exist_use:cF
          { u8: \exp_not:N \tl_to_str:n { #1#2#3 } }
          { \exp_not:o { \__ctex_char_auxii:NNN {#1} {#2} {#3} } }
      }
    \exp_args:Nne \use:n
      { \cs_gset:Npn \__ctex_char_auxii:NNNN #1#2#3#4 }
      {
        \exp_not:N \cs_if_exist_use:cF
          { u8: \exp_not:N \tl_to_str:n { #1#2#3#4 } }
          { \exp_not:o { \__ctex_char_auxii:NNNN {#1} {#2} {#3} {#4} } }
      }
  }
```

`\mule@arg` 处理文档被 `extconv` 等工具预处理过的情况, 只考虑最简单的情形。

```
\group_begin:
\char_set_catcode_active:n { "7F }
\cs_new:Npn \__ctex_char_preproc:w #1 ^^7f #2 ^^7f
  { \exp_args:No \CTEX@char@nn { \int_value:w `#1 } {#2} }
\cs_gset_eq:NN \mule@arg \__ctex_char_preproc:w
\group_end:
\exp_args:Ne \ctex_at_end:n
  {
    \char_set_catcode:nn
      { \int_value:w "7F }
      { \char_value_catcode:n { "7F } }
  }
```

`\CTEX@char@n` `\CTEX@char@nn` `\CTEX@char@nnn` `\CTEX@char@nnnn` 额外加入 `\mode_leave_vertical:` 是为了解决某些历史遗留问题。

```
\cs_new_protected:Npn \CTEX@char@n #1
  {
    \mode_leave_vertical:
    \use:c { CJK@ #1 }
    \CJK@ignorespaces
  }
\cs_new_protected:Npn \CTEX@char@nn #1#2
  {
```

```
    \mode_leave_vertical:
    \use:c { CJK@ #1 } {#2}
    \CJK@ignorespaces
  }
\cs_new_protected:Npn \CTEX@char@nnn  #1#2#3
  {
    \mode_leave_vertical:
    \use:c { CJK@ #1 } {#2} {#3}
    \CJK@ignorespaces
  }
\cs_new_protected:Npn \CTEX@char@nnnn #1#2#3#4
  {
    \mode_leave_vertical:
    \use:c { CJK@ #1 } {#2} {#3} {#4}
    \CJK@ignorespaces
  }
```

\CTEX@char@nn@n
\CTEX@char@nnn@n
\CTEX@char@nnnn@n

以下给 \CTEX@char@nn 等定义在 \text_expand:n 中的等价形式, 将它们的参数用只有一个参数的 \CTEX@char@nn@n 等包装起来, 并将 \CTEX@char@nn@n 等放入排除列表中, 避免可能的大小写转化。

```
\cs_new_protected:Npn \CTEX@char@nn@n   #1 { \CTEX@char@nn   #1 }
\cs_new_protected:Npn \CTEX@char@nnn@n  #1 { \CTEX@char@nnn  #1 }
\cs_new_protected:Npn \CTEX@char@nnnn@n #1 { \CTEX@char@nnnn #1 }
\cs_new:Npn \__ctex_char_wrap:nn #1#2
  { \CTEX@char@nn@n   { {#1} {#2} } }
\cs_new:Npn \__ctex_char_wrap:nnn #1#2#3
  { \CTEX@char@nnn@n  { {#1} {#2} {#3} } }
\cs_new:Npn \__ctex_char_wrap:nnnn #1#2#3#4
  { \CTEX@char@nnnn@n { {#1} {#2} {#3} {#4} } }
\tl_put_right:Nn \l_text_case_exclude_arg_tl
  {
    \CTEX@char@n
    \CTEX@char@nn@n
    \CTEX@char@nnn@n
    \CTEX@char@nnnn@n
  }
\text_declare_expand_equivalent:Nn \CTEX@char@nn   { \__ctex_char_wrap:nn }
\text_declare_expand_equivalent:Nn \CTEX@char@nnn  { \__ctex_char_wrap:nnn }
\text_declare_expand_equivalent:Nn \CTEX@char@nnnn { \__ctex_char_wrap:nnnn }
```

以下给 \CTEX@char@nn@n 等定义在 \text_purify:n 中的等价形式, 将参数中的数字转化回字符。

```
\cs_new:Npn \__ctex_char_raw:n #1
  { \char_generate:nn {#1} { 12 } }
\cs_new:Npn \__ctex_char_raw:nn #1#2
  {
    \char_generate:nn {#1} { 12 }
    \char_generate:nn {#2} { 12 }
  }
\cs_new:Npn \__ctex_char_raw:nnn #1#2#3
  {
    \char_generate:nn {#1} { 12 }
    \char_generate:nn {#2} { 12 }
    \char_generate:nn {#3} { 12 }
  }
\cs_new:Npn \__ctex_char_raw:nnnn #1#2#3#4
  {
    \char_generate:nn {#1} { 12 }
    \char_generate:nn {#2} { 12 }
    \char_generate:nn {#3} { 12 }
    \char_generate:nn {#4} { 12 }
  }
\cs_new:Npn \__ctex_char_raw_nn:n   #1 { \use:e { \__ctex_char_raw:nn   #1 } }
\cs_new:Npn \__ctex_char_raw_nnn:n  #1 { \use:e { \__ctex_char_raw:nnn  #1 } }
```

```
\cs_new:Npn \__ctex_char_raw_nnnn:n #1 { \use:e { \__ctex_char_raw:nnnn #1 } }
\text_declare_purify_equivalent:Nn \CTEX@char@n      { \__ctex_char_raw:n }
\text_declare_purify_equivalent:Nn \CTEX@char@nn@n   { \__ctex_char_raw_nn:n }
\text_declare_purify_equivalent:Nn \CTEX@char@nnn@n  { \__ctex_char_raw_nnn:n }
\text_declare_purify_equivalent:Nn \CTEX@char@nnnn@n { \__ctex_char_raw_nnnn:n }
```

\ctex_ignorespaces_case:N
\ctex_set_ignorespaces:

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

```
\cs_new_protected:Npn \ctex_ignorespaces_case:N #1
  {
    \cs_set_protected:Npn \ctex_set_ignorespaces:
      { \cs_set_eq:NN \CJK@ignorespaces #1 }
    \ctex_set_ignorespaces:
  }
\cs_new_protected:Npn \ctex_set_ignorespaces:
  { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }
```

\CJKhook

CJK 和 CJK* 环境都会重新定义 \CJK@ignorespaces。我们在 CJK 宏包提供的 \CJKhook 里重新设置它，让这两个环境忽略空格的方式都受 space 选项的控制。这对 zhmCJK 是必要的。

```
\tl_if_exist:NF \CJKhook { \tl_new:N \CJKhook }
\tl_gput_right:Nn \CJKhook { \ctex_set_ignorespaces: }
```

\ctex_punct_set:n

设置 CJK 族对应到实际的字体。#1 是 fontset 的名字。

```
\cs_new_protected:Npn \ctex_punct_set:n #1
  {
    \clist_map_inline:Nn \c__ctex_punct_family_clist
      {
        \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
          {
            \cs_set_eq:cc
              { CJKpunct@ ##1 @spaces }
              { c__ctex_ #1 ##1 _punct_spaces_tl }
          }
      }
  }
\clist_const:Nn \c__ctex_punct_family_clist
  {
    zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
    zhsongb , zhheil , zhheib , zhyoub ,
    zhyahei , zhyaheib , zhpf , zhpfb
  }
```

\ctex_punct_map_family:nn

CJK 族 #1 使用族 #2 的边界信息。

```
\cs_new_protected:Npn \ctex_punct_map_family:nn #1#2
  {
    \cs_if_free:cF { CJKpunct@ #2 @spaces }
      { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
  }
```

\ctex_punct_map_bfseries:nn

CJK 族 #1 的 \bfseries 使用族 #2 的边界信息。

```
\cs_new_protected:Npn \ctex_punct_map_bfseries:nn #1#2
  {
    \clist_map_inline:nn {#1}
      {
        \ctex_punct_map_series:nnn { ##1 } { b } {#2}
        \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
      }
  }
\cs_new_protected:Npn \ctex_punct_map_series:nnn #1#2#3
  {
```

```
    \CJKpunctmapfamily { C19 } {#1} {#2} { m }  {#3}
    \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
    \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
    \CJKpunctmapfamily { C70 } {#1} {#2} { m }  {#3}
    \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
    \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
  }
```

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

```
\cs_new_protected:Npn \ctex_punct_map_itshape:nn #1#2
  {
    \CJKpunctmapfamily { C19 } {#1} { m }  { it } {#2}
    \CJKpunctmapfamily { C19 } {#1} { b }  { it } {#2}
    \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
    \CJKpunctmapfamily { C70 } {#1} { m }  { it } {#2}
    \CJKpunctmapfamily { C70 } {#1} { b }  { it } {#2}
    \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
  }
```

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

```
\cs_new_protected:Npn \ctex_punct_space:nn #1#2
  { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
\cs_new_eq:NN \ctexspadef \ctex_punct_space:nn
```

载入边界信息文件。

```
\ctex_file_input:n { ctexspa.def }
```

```
⟨/pdftex⟩
```

15.3.3　ctex-engine-xetex.def

```
⟨*xetex⟩
```

```
\RequirePackage { xeCJK }
\exp_args:Ne \xeCJKsetup
  {
    LoadFandol   = false ,
    PunctStyle   = \l__ctex_punct_tl
  }
```

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。

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

```
⟨/xetex⟩
```

15.3.4　ctex-engine-luatex.def

```
⟨*luatex⟩
```

LuaTeX-ja 为了兼容 pLaTeX 的使用习惯, 对 LaTeX 2$_\varepsilon$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。

```
\msg_new:nnn { ctex } { luatexja-loaded }
  {
    Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
    Loading~file~`#1'~will~abort!
  }
\@ifpackageloaded { luatexja }
  { \msg_critical:nne { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
  {
```

```
    \ctex_at_begin_package:nn { luatexja }
      { \msg_redirect_name:nnn { ctexhook } { disable-package } { info } }
    \ctex_at_end_package:nn { luatexja }
      { \msg_redirect_name:nnn { ctexhook } { disable-package } { } }
    \ctex_disable_package:n { ltj-latex }
  }
\RequirePackage { luatexja }
\@ifpackagelater { luatexja } { 2020/04/12 } { }
  { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
\RequirePackage { fontspec }
\@ifpackagelater { fontspec } { 2020/02/21 } { }
  { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
```

引擎文件是通过 `\ctex_file_input:n` 载入的，其中的 `\catcodetable` 机制会完整恢复文件载入之前的 `\catcode` 状态，在引擎文件中的 `\catcode` 设置都无效。因此，对 `\ltjlineendcomment` 的设置要放到文件之外进行。

```
\ctex_at_end:n { \char_set_catcode_comment:n { \ltjlineendcomment } }
```

15.3.4.1 LuaTeX-ja 的默认设置

```
\ExplSyntaxOff
```

以下设置抄录自 `lltjdefs.sty`, 略有改动。

U+2460–U+24FF (Enclosed Alphanumerics) 原属于字符范围 6, 是 JAchar, 我们把它们归入字符范围 3, 改成 ALchar。

```
\ltjdefcharrange{1}{"80-"36F, "1E00-"1EFF}
\ltjdefcharrange{2}{"370-"4FF, "1F00-"1FFF}
\ltjdefcharrange{3}{%
  "2000-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
\ltjdefcharrange{4}{%
   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
  "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A95F, "A980-"ABFF, "E000-"F8FF,
  "FB00-"FE0F, "FE20-"FE2F, "FE70-"FEFF, "10000-"1AFFF, "1B170-"1F0FF,
  "1F300-"1FFFF, "2000-"206F}
\ltjdefcharrange{5}{"D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
\ltjdefcharrange{6}{%
  "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
  "4E00-"9FFF, "F900-"FAFF, "FE10-"FE1F, "FE30-"FE6F, "FF00-"FFEF,
  "1B000-"1B16F, "1F100-"1F2FF, "20000-"3FFFF, "E0100-"E01EF}
\ltjdefcharrange{7}{%
  "1100-"11FF, "2F00-"2FFF, "3100-"318F, "31A0-"31EF, "A000-"A4CF,
  "A960-"A97F, "AC00-"D7FF}
\ltjdefcharrange{8}{"A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
```

将间隔号、引号、破折号等中西文公用的标点符号归入字符范围 9, 将他们设置为 JAchar。

```
\ltjdefcharrange{9}{%
  "00B7, "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}
```

LuaTeX-ja 默认把字符范围 2 和 3 设置为 JAchar, 我们这里把它们都改成 ALchar。

```
\ltjsetparameter{jacharrange={-1, -2, -3, -4, -5, +6, +7, -8, +9}}
\directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}
```

以下设置抄录自 2020/08/08 之前的 `ltj-latex.sty`。自 2020/08/08 开始, LuaTeX-ja 引入新的缓存机制, 此段设置被整合进 `luatexja.lua`。

```
\@ifpackagelater{luatexja}{2020/08/08}
  { \ltjsetparameter { autospacing, autoxspacing, differentjfm = paverage } }
  {
    \directlua{
      local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
      luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
    }
```

```
    \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
      xkanjiskip=.25\zw plus 1pt minus 1pt,
      autospacing, autoxspacing, jacharrange={-1},
      yalbaselineshift=\z@, yjabaselineshift=\z@,
      jcharwidowpenalty=500, differentjfm=paverage
    }
  }
\ExplSyntaxOn
```

15.3.4.2 LuaTeX-ja 的补丁

```
⟨@@=ctex_ltj⟩
```

\CTEX@alchar
\CTEX@beginallalchar
\CTEX@endallalchar

分组中的字符都是 ALchar 类。

```
\cs_new_protected:Npn \CTEX@alchar #1
  { \CTEX@beginallalchar #1 \CTEX@endallalchar }
\cs_new_protected:Npn \CTEX@beginallalchar
  {
    \group_begin:
      \ctex_ltj_zero_globaldefs:
      \ltj@allalchar
  }
\cs_new_protected:Npn \CTEX@endallalchar
  { \group_end: }
```

\CTEX@chardef@text@cmd

补丁 \chardef@text@cmd, 应用于 \DeclareTextSymbol, 使其定义的符号都是 ALchar。

```
\cs_new_protected:Npn \CTEX@chardef@text@cmd #1
  {
    \cs_set_eq:NN \@ifdefinable \@@ifdefinable
    \tl_set:Nn \l__ctex_ltj_cmd_tl {#1}
    \tex_afterassignment:D \__ctex_ltj_chardef_text_cmd:
    \tex_chardef:D #1
  }
\tl_new:N \l__ctex_ltj_cmd_tl
\cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd:
  { \exp_after:wN \__ctex_ltj_chardef_text_cmd_aux:N \l__ctex_ltj_cmd_tl }
\cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd_aux:N #1
  {
    \int_compare:nNnF {#1} < { "80 }
      { \cs_set_protected:Npx #1 { \CTEX@alchar { \tex_Uchar:D #1 } } }
  }
```

\CTEX@text@composite@x

\@text@composite@x 的重定义, 应用于 \DeclareUnicodeComposite 等。

```
\cs_new_protected:Npn \CTEX@text@composite@x #1#2
  {
    \CTEX@beginallalchar
      \cs_if_exist_use:NF #1 {#2}
    \CTEX@endallalchar
  }
```

\CTEX@add@unicode@accent

\add@unicode@accent 的重定义, 应用于 \DeclareUnicodeAccent。

```
\cs_new_protected:Npx \CTEX@add@unicode@accent #1#2
  {
    \CTEX@beginallalchar
      \exp_not:N \tl_if_blank:nTF {#2} { \tex_Uchar:D "A0 ~ } {#2}
      \exp_not:N \tex_Uchar:D \tex_numexpr:D #1 \scan_stop:
    \CTEX@endallalchar
  }
```

\CTEX@patch@text@cmd

单独补丁由 \DeclareTextCommand 定义的命令。

```
\cs_new_protected:Npn \CTEX@patch@text@cmd #1
```

```
  {
    \exp_args:NNc \__ctex_ltj_patch_text_cmd:NN #1
      { \UnicodeEncodingName \token_to_str:N #1 }
  }
\cs_new_protected:Npn \__ctex_ltj_patch_text_cmd:NN #1#2
  {
    \cs_set_eq:NN \CTEX@textcmd #2
    \ctex_preto_cmd:NnnTF \CTEX@textcmd
      { \ExplSyntaxOff \makeatletter }
      { \CTEX@beginallalchar }
      {
        \ctex_appto_cmd:NnnTF \CTEX@textcmd
          { \ExplSyntaxOff \makeatletter }
          { \CTEX@endallalchar }
          { \cs_set_eq:NN #2 \CTEX@textcmd }
          { \ctex_patch_failure:N #1 }
      }
      { \ctex_patch_failure:N #1 }
  }
```

\CTEX@patch@tunec 重新载入 tunec.def, 使补丁生效。

```
\cs_new_protected:Npn \CTEX@patch@tunec
  {
    \cs_set_eq:NN \chardef@text@cmd   \CTEX@chardef@text@cmd
    \ctex_file_input:n { tuenc.def }
    \cs_set_eq:NN \@text@composite@x  \CTEX@text@composite@x
    \cs_set_eq:NN \add@unicode@accent \CTEX@add@unicode@accent
    \CTEX@patch@text@cmd \textasteriskcentered
  }
\@ifpackageloaded { xunicode }
  { }
  { \CTEX@patch@tunec }
```

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

```
\ctex_at_end_package:nn { xunicode }
  {
    \RequirePackage { xunicode-addon }
    \AtBeginUTFCommand { \CTEX@beginallalchar }
    \AtEndUTFCommand   { \CTEX@endallalchar }
  }
```

对 listings 的补丁是让代码环境支持 JAchar 类。

```
\ctex_at_end_package:nn { listings }
  { \RequirePackage { lltjp-listings } }
```

15.3.4.3 Lua 函数

直接用 token.set_lua 定义, 不是传统意义上的 TeX 宏。

```
\group_begin:
\char_set_catcode_space:n { 32 }
\lua_now:e
  {
```

ctex.newluacmd 定义新的 Lua 函数。

```
    ctex = ctex or { }
    local ctex = ctex
    local functions = lua.get_functions_table()
```

```
  local new_luafunction = luatexbase.new_luafunction
  local create, set_lua = token.create, token.set_lua
  local lua_cmds = {
    lua_call            = true ,
    lua_expandable_call = true ,
  }
  local newluacmd = function (name, func, ...)
    local tok = create(name)
    local id = lua_cmds[tok.cmdname] and tok.index
    local id = id or new_luafunction(name)
    set_lua(name, id, ...)
    functions[id] = func
  end
  ctex.newluacmd = newluacmd

  local ltjfont = luatexja.jfont
  local getattribute = tex.getattribute
  local tex_set, sprint = tex.set, tex.sprint
  local scan_int, scan_arg = token.scan_int, token.scan_argument
```

`\ctex_ltj_add_kyenc:n` 保存 jfont 的编码, 用于判断。

```
  newluacmd("ctex_ltj_add_kyenc:n", ltjfont.add_kyenc_list, "global", "protected")
```

`\ctex_ltj_is_kenc:n` 判断编码是否属于 jfont。

```
  newluacmd("ctex_ltj_is_kenc:n", ltjfont.is_kenc, "global")
```

`\ctex_ltj_patch_external_font:n` 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 luatexja.jfont.print_aftl_address()
  return ';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。

```
  newluacmd("ctex_ltj_patch_external_font:n", function ()
    local s = scan_arg()
    local is_braced, is_quoted
    if s:sub(1,1) == '{' and s:sub(-1)=='}' then
      is_braced = true; s = s:sub(2,-2)
    end
    if s:sub(1,1) == '"' and s:sub(-1) == '"' then
      is_quoted = true; s = s:sub(2,-2)
    end
    s = s .. ltjfont.print_aftl_address()
    if     is_braced then s = '{'..s..'}'
    elseif is_quoted then s = '"'..s..'"'
    end
    sprint(-2, s)
  end, "global")
```

`\ctex_ltj_set_alt_font:nnnn` #1 和 #2 分别是字符区间的首末, #3 是基础字体, #4 是替代字体。

```
  newluacmd("ctex_ltj_set_alt_font:nnnn", function ()
    local b = tonumber(scan_arg())
    local e = tonumber(scan_arg())
    local alt  = scan_arg()
    local base = scan_arg()
    ltjfont.set_alt_font_latex(b, e, alt, base)
  end, "global", "protected")
```

`\ctex_ltj_clear_alt_font:n` 清除 #1 的替代字体。

```
  newluacmd("ctex_ltj_clear_alt_font:n", function ()
```

```
      local base = scan_arg()
      ltjfont.clear_alt_font_latex(base)
    end, "global", "protected")
```

\ctex_ltj_pickup_alt_font:nn 定义 #1 的替代字体, #2 是字体大小。
会在内部执行 \ltj@pickup@altfont@auxy 和 \ltj@pickup@altfont@copy。

```
  newluacmd("ctex_ltj_pickup_alt_font:nn", function ()
    local base = scan_arg()
    local size = scan_arg()
    ltjfont.output_alt_font_cmd("y", base)
    ltjfont.pickup_alt_font_a(size)
  end, "global", "protected")
```

__ctex_ltj_pickup_alt_font:Nn #1 是 font.id, #2 是字体名称。在 \ltj@pickup@altfont@copy 之中使用。

```
  newluacmd("__ctex_ltj_pickup_alt_font:Nn", function ()
    local num  = scan_int()
    local base = scan_arg()
    ltjfont.pickup_alt_font_b(num, base)
  end, "global", "protected")
```

__ctex_ltj_if_alt_set:nT 判断是否存在替代字体。会设置变量 aftl_base 和返回 \@firstofone 或 \@gobble。

```
  newluacmd("__ctex_ltj_if_alt_set:nT", ltjfont.does_alt_set, "global")
```

\ctex_ltj_zero_globaldefs: 设置 \globaldefs 为 0, 避免全局设置, 应当在分组中使用。这里通过 Lua 设置, 可以不受外部 TEX 环境中的 \globaldefs 的影响。

```
  newluacmd("ctex_ltj_zero_globaldefs:", function ()
    tex_set("globaldefs", 0)
  end, "global", "protected")
}
\group_end:
```

15.3.4.4　字体切换方式

\ctex_ltj_select_font:
\CJK@family　\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。

```
\cs_new_protected:Npn \ctex_ltj_select_font:
  {
    \group_begin: \exp_args:NNc \group_end:
    \cs_if_exist_use:NF { \l__ctex_ltj_current_font_tl }
      { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
  }
\tl_new:N \CJK@family
\tl_new:N \l__ctex_ltj_current_font_tl
\tl_set:Nn \l__ctex_ltj_current_font_tl
  { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }
```

__ctex_ltj_select_font_aux: 使用 \pickup@font 取得字体名称前, 总需要先设置 \font@name。在这里将 \f@family 换成 CJK 字体族, 并确保编码正确。

```
\cs_new_protected:Npn \__ctex_ltj_select_font_aux:
  {
    \group_begin:
      \tl_set_eq:NN \f@encoding \CJK@encoding
      \tl_set_eq:NN \f@family \CJK@family
      \cs_set_eq:NN \pickup@font \ctex_ltj_pickup_font:
      \__ctex_ltj_push_fontname:n { \curr@fontshape / \f@size }
      \ctex_ltj_pickup_font:
    \group_end:
    \font@name
    \__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。

```
    \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
      { \__ctex_ltj_select_font_aux: }
  }
\cs_new_protected:Npn \__ctex_ltj_push_fontname:n #1
  {
    \seq_gpush:No \g__ctex_ltj_fontname_seq { \font@name }
    \tl_gset:Ne \font@name { \exp_not:c {#1} }
  }
\cs_new_protected:Npn \__ctex_ltj_pop_fontname:
  {
    \seq_gpop:NNT \g__ctex_ltj_fontname_seq \l__ctex_ltj_tmp_tl
      { \tl_gset_eq:NN \font@name \l__ctex_ltj_tmp_tl }
  }
\seq_new:N \g__ctex_ltj_fontname_seq
```

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

```
\cs_new_protected:Npn \ctex_ltj_pickup_font:
  {
    \exp_after:wN \cs_if_exist:NF \font@name
      {
        \group_begin:
          \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
          \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
          \define@newfont
        \group_end:
      }
  }
\cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:
```

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

```
%%%%%%%% \jfont\CS={...:...;jfm=metric;...}, \globaljfont
\protected\def\jfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX
    (false, 'yoko','\luatexluaescapestring{\noexpand#1}')}}
\protected\def\globaljfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX
    (true,  'yoko','\luatexluaescapestring{\noexpand#1}')}}
\newluafunction\ltj@@jfont@inner
\directlua{
  local t = lua.get_functions_table()
  t[\the\ltj@@jfont@inner] = luatexja.jfont.jfontdefY
}
\def\ltj@@jfont{\luafunction\ltj@@jfont@inner}
```

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

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

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

```
\cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
  {
    \get@external@font
    \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
      {
```

```
        \tl_set:Ne \external@font
          { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
      }
    \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。

```
    \font@name
    \ctex_ltj_use_jfont:
    \use:c { \f@encoding + \f@family }
    \use:c { \curr@fontshape }
  }
```

\ctex_ltj_use_jfont: 使用 jfont，确保当前的 \font 是 jfont。

```
\cs_new_protected_nopar:Npn \ctex_ltj_use_jfont:
  { \tex_setfontid:D \ltj@curjfnt }
```

\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。

```
\cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
  {
    \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
      {
        \group_begin:
        \tl_set_eq:NN \CJK@family \f@family
        \cs_if_exist:cF { \l__ctex_ltj_current_font_tl  }
          {
            \cs_gset_protected_nopar:Npx \subst@correction
              {
                \cs_new_eq:NN
                  \exp_not:c { \l__ctex_ltj_current_font_tl }
                  \font@name
              }
            \group_insert_after:N \group_insert_after:N
            \group_insert_after:N \subst@correction
          }
        \group_end:
      }
  }
```

\ctex_ltj_if_alternate_shape_exist:n*TF* 即 LuaTeX-ja 中的 \ltj@@does@alt@set，判断是否存在替代字体。

```
\prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
  {
    \__ctex_ltj_if_alt_set:nT {#1} { \prg_return_true: \use_none:n }
    \prg_return_false:
  }
```

__ctex_ltj_patch_external_font:w
```
\cs_new:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
  { \ctex_ltj_patch_external_font:n {#1} ~ at }
```

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

```
\cs_new_protected:Npn \ctex_ltj_select_alternate_font:
  {
    \ctex_ltj_if_alternate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
      {
        \ctex_ltj_pickup_alt_font:nn
          { \l__ctex_ltj_current_shape_tl } { \f@size }
      }
  }
\tl_new:N \l__ctex_ltj_current_shape_tl
\tl_set:Nn \l__ctex_ltj_current_shape_tl
  { \CJK@encoding / \CJK@family / \f@series / \f@shape }
```

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

```
\cs_new_protected:Npn \ltj@pickup@altfont@auxy #1
  {
    \cs_if_exist:cF { #1/\f@size }
      {
        \group_begin:
          \use:e { \exp_not:N \split@name #1 / \f@size } \@nil
          \__ctex_ltj_push_fontname:n { \curr@fontshape / \f@size }
          \ctex_ltj_pickup_font:
        \group_end:
        \__ctex_ltj_pop_fontname:
      }
  }
```

\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。

```
\cs_new_protected:Npn \ltj@pickup@altfont@copy #1#2
  {
    \ltj@@getjfontnumber #1
    \__ctex_ltj_pickup_alt_font:Nn \ltj@tempcntc {#2}
  }
```

15.3.4.5　数学字体族

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

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

\⟨*encoding*⟩/⟨*family*⟩/⟨*series*⟩/⟨*shape*⟩

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

```
\cs_new:Npn \ctex_ltj_if_jfont:nTF #1
  {
    \ctex_ltj_is_kenc:n { \__ctex_ltj_ltj_if_jfont:w #1 / \q_stop }
    \legacy_if:nTF { in@ }
  }
\cs_new:Npn \__ctex_ltj_ltj_if_jfont:w #1 / #2 \q_stop
  {#1}
```

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

```
\cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
```

```
  { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
\group_begin:
  \char_set_catcode_other:N M
  \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
    { \ctex_ltj_if_jfont:nTF {#3} }
\group_end:
```

\getanddefine@fonts
\ctex_ltj_get_and_define_fonts:nN

在使用的场合，\escapechar 已经被设置成 −1，使用 \token_to_str:N 就可以得到名字，不必使用 \cs_to_str:N。

```
\cs_new_protected:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
  {
    \exp_args:No \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
      { \ctex_ltj_get_and_define_fonts_ja:nN }
      { \ctex_ltj_get_and_define_fonts_al:nN }
      {#1} #2
  }
\cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
\cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
\cs_new_protected:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
  {
    \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
    \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
    \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
    \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
    \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
    \ctex_ltj_pickup_font:
    \tl_put_right:Ne \math@fonts
      {
        \ltj@setpar@global
        \ltj@@set@stackfont #1 , \textfont@name   \c_colon_str { MJT }
        \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
        \ltj@@set@stackfont #1 , \font@name       \c_colon_str { MJSS }
      }
  }
```

\use@mathgroup
\ctex_ltj_use_math_group:Nn

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

```
\cs_new_protected:Npn \ctex_ltj_use_math_group:Nn #1#2
  {
    \mode_if_math:T
      {
        \math@bgroup
          \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
          \ctex_ltj_math_group_hook:
          \ctex_ltj_if_jfont_math:NTF #1
            { \jfam } { \mathgroup } #2 \scan_stop:
        \math@egroup
      }
  }
\cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
\cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
```

对 unicode-math 的补丁主要是将 unicode-math-table.tex 中的数学符号设置为 luatexja 中的数学字母。本段代码应放在 \ctex_ltj_math_group_hook: 的定义之后，避免因宏包载入顺序而造成的编译错误。

```
\cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
  {
    \group_begin:
      \cs_set_protected:Npn #1 ##1##2##3
        { \ltjsetmathletter { ##1 } }
      #2
    \group_end:
  }
```

```
\ctex_at_end_package:nn { unicode-math }
  {
    \cs_if_exist:NTF \um_input_math_symbol_table:
      {
        \ctex_ltj_set_math_letter:NN
          \um_sym:nnn
          \um_input_math_symbol_table:
      }
      {
        \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
        \cs_set_protected:Npn \ctex_ltj_math_group_hook:
          { \__um_switchto_literal: }
        \ctex_ltj_set_math_letter:NN
          \__um_sym:nnn
          \__um_input_math_symbol_table:
      }
  }
```

15.3.4.6　字体族的定义与使用

\ctex_set_jfm:n
\l__ctex_ltj_jfm_tl
\l__ctex_ltj_redirect_jfm_prop

设置 JFM, 需要进行一些重定向操作。

```
\cs_new_protected:Npn \ctex_set_jfm:n #1
  {
    \prop_get:NnNF \l__ctex_ltj_redirect_jfm_prop {#1} \l__ctex_ltj_jfm_tl
      { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
  }
\cs_generate_variant:Nn \ctex_set_jfm:n { o }
\prop_new:N \l__ctex_ltj_redirect_jfm_prop
\prop_set_from_keyval:Nn \l__ctex_ltj_redirect_jfm_prop
  {
    plain    = mono ,
    quanjiao = zh_CN / quanjiao ,
    banjiao  = zh_CN / banjiao ,
    kaiming  = zh_CN / kaiming
  }
\keys_define:nn { ctex_ltj / fontspec }
  {
    JFM           .code:n = \ctex_set_jfm:n {#1} ,
    JFM .value_required:n = true
  }
\tl_new:N \l__ctex_ltj_jfm_tl
```

注意这里的 `@@=ctex_ltj`。

```
\ctex_set_jfm:o { \l__ctex_punct_tl }
```

\CJK@encoding

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

```
\tl_const:Nn \CJK@encoding { LTJY3 }
\DeclareFontEncoding { \CJK@encoding } { } { }
\use:e
  {
    \exp_not:N \DeclareFontSubstitution
      { \CJK@encoding } { song } { \mddefault } { \shapedefault }
  }
\ctex_ltj_add_kyenc:n { \CJK@encoding }
\DeclareFontFamily { \CJK@encoding } { song } { }
\DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \shapedefault }
  { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
\DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
```

```
  { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
\tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
\DeclareSymbolFont { \c__ctex_ltj_math_tl }
  { \CJK@encoding } { song } { \mddefault } { \shapedefault }
\SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
  { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
\int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
\jfam \c__ctex_ltj_math_fam_int
```

这是 luatexja-fontspec 中新增的一些字体选项。

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

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

```
\keys_define:nn { fontspec-preparse-external }
  {
    NoEmbed .code:n =
      { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
  }
\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。

```
\cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
  {
    \group_begin:
    \clist_clear:N \l__ctex_ltj_char_range_clist
    \prop_clear:N \l__ctex_ltj_alternate_prop
    \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
    \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
    \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
    \ctex_ltj_set_alternate_family:nnF {#1} {#3}
      {
        \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
        \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
          {#1} { \l__ctex_ltj_font_options_clist }
        \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
        \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
        \__ctex_ltj_gset_family_cs:nn {#1} {#3}
      }
    \group_end:
  }
\tl_new:N \l__ctex_ltj_base_CJKfamily_tl
\clist_new:N \l__ctex_ltj_font_options_clist
```

`\__ctex_ltj_use_global_options:N` 应用默认字体选项, 并总是设置 `JFM` 和 `NFSSEncoding`。

```
\cs_new_protected:Npn \__ctex_ltj_use_global_options:N #1
  {
    \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
    \clist_put_left:Ne #1
      { NFSSEncoding = \CJK@encoding , JFM = \l__ctex_ltj_jfm_tl }
  }
```

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

```
\prop_new:N \g__ctex_ltj_family_name_prop
\prop_new:N \g__ctex_ltj_family_font_name_prop
\prop_new:N \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。

```
\cs_new_protected:Npn \__ctex_ltj_check_family:n #1
```

```
  {
    \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
      {
        \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
        \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
        \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
          {
            \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
            \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
            \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
            \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
          }
        \msg_warning:nnee { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
      }
  }
\tl_new:N \l__ctex_ltj_tmp_tl
\msg_new:nnn { ctex } { redefine-family }
  { Redefining~CJKfamily~`\__ctex_ltj_msg_family_map:n {#1}'~(#2). }
```

__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。

```
\cs_new_protected:Npn \__ctex_ltj_gset_family_cs:nn #1#2
  {
    \cs_gset_protected:cpx { \__ctex_ltj_family_csname:n {#1} }
      {
        \group_begin:
        \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
        \exp_not:n { \fontspec_gset_family:Nnn \g__ctex_ltj_fontspec_family_tl }
          { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
        \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
          { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
        \__ctex_ltj_set_alternate_family:n {#1}
        \group_end:
      }
  }
\tl_new:N \l__ctex_ltj_base_family_tl
\tl_new:N \g__ctex_ltj_fontspec_family_tl
\cs_new:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
\cs_new_protected:Npn \__ctex_ltj_set_alternate_family:n #1
  {
    \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
    \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
    \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
    \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
  }
\cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }
```

\CJKfamily 切换字体。

```
\NewDocumentCommand \CJKfamily { m }
  { \ctex_ltj_switch_family:e {#1} \tex_ignorespaces:D }
\cs_new_protected:Npn \ctex_ltj_switch_family:n #1
  {
    \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
      {
        \tl_set:Nn \l_ctex_ltj_family_tl {#1}
        \selectfont
      }
      { \__ctex_ltj_family_unknown_warning:n {#1} }
  }
\tl_new:N \l_ctex_ltj_family_tl
\cs_generate_variant:Nn \ctex_ltj_switch_family:n { e }
```

\ctex_ltj_family_if_exist:nN*TF* 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。

```
\prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
```

```
    {
      \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
        { \prg_return_true: }
        {
          \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
            {
              \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
              \prg_return_true:
            }
            { \prg_return_false: }
        }
    }
\prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { e } { T , F , TF }

\__ctex_ltj_family_unknown_warning:n
\cs_new_protected:Npn \__ctex_ltj_family_unknown_warning:n #1
    {
      \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
        {
          \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
            {
              \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
              \msg_warning:nnn { ctex } { family-unknown } {#1}
            }
        }
    }
\seq_new:N \g__ctex_ltj_unknown_family_seq
\msg_new:nnn { ctex } { family-unknown }
    {
      Unknown~CJK~family~`\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
      Try~to~use~`\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
    }
\cs_new:Npn \__ctex_ltj_msg_def_family_map:n #1
    {
      \str_case_e:nnF {#1}
        {
          \CJKrmdefault { \token_to_str:N \setCJKmainfont }
          \CJKsfdefault { \token_to_str:N \setCJKsansfont }
          \CJKttdefault { \token_to_str:N \setCJKmonofont }
        }
        { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
      [...]\{...\}
    }
\cs_new:Npn \__ctex_ltj_msg_family_map:n #1
    {
      \str_case_e:nnF {#1}
        {
          \CJKrmdefault { \token_to_str:N \CJKrmdefault }
          \CJKsfdefault { \token_to_str:N \CJKsfdefault }
          \CJKttdefault { \token_to_str:N \CJKttdefault }
        }
        {#1}
    }

\ctex_ltj_fontspec:nn
\cs_new_protected:Npn \ctex_ltj_fontspec:nn #1#2
    {
      \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
        { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
        { \ctex_ltj_switch_family:e { \l_ctex_ltj_family_tl } }
        {
          \int_gincr:N \g__ctex_ltj_family_int
          \__ctex_ltj_fontspec:enn
            { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
            {#1} {#2}
        }
    }
\cs_new_protected:Npn \ctex_ltj_fontspec:ee #1#2
    { \use:e { \ctex_ltj_fontspec:nn {#1} {#2} } }
```

```
\cs_new_protected:Npn \__ctex_ltj_fontspec:nnn #1#2#3
  {
    \bool_if:NT \l__ctex_ltj_add_alternate_bool
      {
        \cs_if_free:cF
          { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
          {
            \cs_gset_eq:cc
              { \__ctex_ltj_alternate_cs:n { reset / #1 } }
              { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
            \cs_gset_eq:cc
              { \__ctex_ltj_alternate_cs:n { clear / #1 } }
              { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
          }
        \bool_set_false:N \l__ctex_ltj_add_alternate_bool
      }
    \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
    \ctex_ltj_set_family:nnn {#1} {#2} {#3}
    \ctex_ltj_switch_family:n {#1}
  }
\cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { e }
\prop_new:N \g__ctex_ltj_fontspec_prop
```

\ctex_ltj_add_font_features:n
\ctex_ltj_add_font_features:nn

```
\cs_new_protected:Npn \ctex_ltj_add_font_features:n #1
  { \ctex_ltj_add_font_features:en { \l_ctex_ltj_family_tl } {#1} }
\cs_new_protected:Npn \ctex_ltj_add_font_features:nn #1#2
  {
    \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
      {#1} \l__ctex_ltj_tmp_tl
      {
        \prop_get:NnN \g__ctex_ltj_family_font_options_prop
          {#1} \l__ctex_ltj_font_options_clist
        \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
        \bool_set_true:N \l__ctex_ltj_add_alternate_bool
        \ctex_ltj_fontspec:ee
          { \exp_not:o { \l__ctex_ltj_font_options_clist } }
          { \exp_not:o { \l__ctex_ltj_tmp_tl } }
      }
      { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
  }
\bool_new:N \l__ctex_ltj_add_alternate_bool
\cs_generate_variant:Nn \ctex_ltj_add_font_features:n  { e }
\cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { e }
\msg_new:nnn { ctex } { addCJKfontfeature-ignored }
  {
    \token_to_str:N \addCJKfontfeature (s)~ignored.\\
    It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
  }
```

__ctex_ltj_pass_args:nnnn 为了支持字体属性可选项在前在后两种语法, 给出两个辅助工具, 自带展开功能。

```
\cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
  {
    \tl_if_novalue:nTF {#2}
      { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
      {
        \use:e { #1 {#2} {#3} }
        #4
      }
  }
\NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
  {
    \use:e { #1 {#4} {#2} }
    #3
  }
```

\setCJKfamilyfont
\newCJKfontfamily
\CJKfontspec
\addCJKfontfeatures

```
\NewDocumentCommand \setCJKfamilyfont { m o m }
```

```
  {
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
      { }
  }
\msg_new:nnn { ctex } { command-already-defined }
  { Control~sequence~#1~is~already~defined. }
\NewDocumentCommand \newCJKfontfamily { o m o m }
  {
    \tl_set:Ne \l__ctex_ltj_tmp_tl
      { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
    \cs_if_exist:NTF #2
      { \msg_error:nne { ctex } { command-already-defined }
          { \token_to_str:N #2 } }
      {
        \cs_set_protected:Npx #2
          { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
      }
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
      { }
  }
\NewDocumentCommand \CJKfontspec { o m }
  {
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_fontspec:nn } {#1} {#2}
      { \tex_ignorespaces:D }
  }
\NewDocumentCommand \addCJKfontfeatures { m }
  {
    \ctex_ltj_add_font_features:e {#1}
    \tex_ignorespaces:D
  }
\cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\NewDocumentCommand \setCJKmainfont { o m }
  {
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
      { \normalfont }
  }
\cs_new_eq:NN \setCJKromanfont \setCJKmainfont
\NewDocumentCommand \setCJKsansfont { o m }
  {
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
      { \normalfont }
  }
\NewDocumentCommand \setCJKmonofont { o m }
  {
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
      { \normalfont }
  }
\NewDocumentCommand \setCJKmathfont { o m }
  {
    \__ctex_ltj_pass_args:nnnn
      { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
      { }
  }
\NewDocumentCommand \defaultCJKfontfeatures { m }
  { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
\clist_new:N \g__ctex_ltj_default_features_clist
\@onlypreamble \setCJKmainfont
\@onlypreamble \setCJKsansfont
\@onlypreamble \setCJKmonofont
\@onlypreamble \setCJKmathfont
```

```
\@onlypreamble \setCJKromanfont
\@onlypreamble \defaultCJKfontfeatures
```

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

```
\cs_new_protected:Npn \ctex_ltj_ensure_default_family:
  {
    \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
      {
        \ctex_ltj_family_if_exist:eNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
          {
            \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
              { \use:n }
              {
                \ctex_ltj_family_if_exist:eNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
                  { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
                  { \use:n }
              }
              {
                \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
                  {
                    \prop_map_break:n
                      { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
                  }
              }
          }
        \normalfont
        \ctex_ltj_update_mathfont:
      }
  }
```

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

```
\cs_new_protected:Npn \ctex_ltj_update_mathfont:
  {
    \ctex_ltj_family_if_exist:eNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
      { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
      {
        \ctex_ltj_family_if_exist:eNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
          { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
      }
  }
\cs_new_protected:Npn \ctex_ltj_update_mathfont:n #1
  {
    \tl_const:Ne \c__ctex_ltj_math_family_tl {#1}
    \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
      { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
    \cs_if_free:cTF
      { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\shapedefault }
      {
        \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
          { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
      }
      {
        \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
          { \c__ctex_ltj_math_family_tl } { \bfdefault } { \shapedefault }
      }
  }
```

15.3.4.7　替代字体的设置

AlternateFont CharRange 设置替代字体的选项。

```
\keys_define:nn { ctex_ltj / fontspec }
  {
```

```
    AlternateFont  .code:n = \ctex_ltj_set_alternate_prop:n {#1} ,
    AlternateFont  .value_required:n = true ,
    CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
    CharRange .value_required:n = true
  }
```

\ctex_ltj_set_alternate_prop:n 保存替代字体序列。

```
\cs_new_protected:Npn \ctex_ltj_set_alternate_prop:n #1
  { \clist_map_function:nN {#1} \__ctex_ltj_push_alternate_prop:n }
\cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:n #1
  {
    \clist_set:Ne \l__ctex_ltj_tmp_clist { \tl_head:n {#1} }
    \tl_remove_all:Nn \l__ctex_ltj_tmp_clist { ~ }
    \exp_args:No \__ctex_ltj_push_alternate_prop:nn
      { \l__ctex_ltj_tmp_clist } {#1}
  }
\cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:nn #1
  {
    \prop_remove:Nn \l__ctex_ltj_alternate_prop {#1}
    \prop_put:Nnn \l__ctex_ltj_alternate_prop {#1}
  }
\clist_new:N \l__ctex_ltj_tmp_clist
\prop_new:N \l__ctex_ltj_alternate_prop
```

\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。

```
\cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnF
  {
    \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
      { \__ctex_ltj_set_family_aux:nnn }
      { \__ctex_ltj_set_alternate_family_aux:nnn }
  }
\cs_new_protected:Npn \__ctex_ltj_set_family_aux:nnn #1#2#3
  {
    \__ctex_ltj_check_family:n {#1}
    \prop_if_empty:NF \l__ctex_ltj_alternate_prop
      { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
    #3
  }
\cs_new_protected:Npn \__ctex_ltj_set_alternate_family_aux:nnn #1#2#3
  { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
```

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

```
\cs_new_protected:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
  {
    \prop_map_inline:Nn \l__ctex_ltj_alternate_prop
      { \__ctex_ltj_save_alternate_auxi:w ##2 { } \q_mark #1 {#2} }
  }
\cs_new_protected:Npn \__ctex_ltj_save_alternate_auxi:w #1#2#
  {
    \tl_if_blank:nTF {#2}
      { \__ctex_ltj_save_alternate_auxii:w {#1} }
      { \__ctex_ltj_save_alternate_auxii:w {#1} {#2} }
  }
\cs_new_protected:Npn \__ctex_ltj_save_alternate_auxii:w #1#2#3 #4 \q_mark #5#6
  {
    \clist_set:Nn \l__ctex_ltj_char_range_clist {#1}
    \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#3}
    \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
    \tl_if_blank:nTF {#2}
      { \tl_set:Nn \l__ctex_ltj_tmp_tl {#6} }
      {
        \tl_set:Ne \l__ctex_ltj_tmp_tl { \tl_trim_spaces:n {#2} }
        \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#6}
```

```
      }
    \use:e
      {
        \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #5
          { \exp_not:o { \l__ctex_ltj_char_range_clist } }
          { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
          { \exp_not:o { \l__ctex_ltj_tmp_tl } }
      }
  }
\clist_new:N \l__ctex_ltj_alternate_options_clist
\cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
```

\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。

```
\cs_new_protected:Npn \ctex_ltj_set_alternate_family:nn #1#2
  {
    \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
    \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
    \ctex_ltj_set_alternate_family:coonn
      { \__ctex_ltj_alternate_cs:n {#1} }
      { \l__ctex_ltj_char_range_clist }
      { \l__ctex_ltj_font_options_clist } {#2} {#1}
  }
\cs_new_protected:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
  {
    \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
      { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
    \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
  }
\cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }
```

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

```
\cs_new_protected:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
  {
    \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
    \cs_gset_protected:Npx #1
      { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
  }
```

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

```
\cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
  {
    \group_begin:
    \cs_set_eq:NN \CJKfamily \use_none:n
    \ctex_ltj_swap_cs:NN
      \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
    \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
    \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
    \group_end:
  }
\tl_new:N \l__ctex_ltj_alternate_family_tl
```

\ctex_ltj_swap_cs:NN 交换两个控制序列的意义。

```
\cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
  {
    \cs_set_eq:NN \__ctex_ltj_tmp:w #1
    \cs_set_eq:NN #1 #2
    \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
    \cs_undefine:N \__ctex_ltj_tmp:w
  }
```

LTJFONTUID
__ctex_ltj_update_family_uid:N

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

```
\keys_define:nn { fontspec } { LTJFONTUID .code:n = }
\cs_new_protected:Npn \__ctex_ltj_update_family_uid:N #1
  {
    \int_gincr:N \g__ctex_ltj_family_int
    \clist_put_right:Ne #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
  }
\int_new:N \g__ctex_ltj_family_int
```

\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*⟩。

```
\cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
  {
    \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
    \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
      { \l__ctex_ltj_base_family_tl } {#3} {#4} {#2} {#3} {#4}
  }
```

\ctex_ltj_set_alternate_shape:Nnnnnnn

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

```
\cs_new_protected:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
  {
    \clist_map_inline:Nn #1
      {
        \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
          {
            \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
              \l__ctex_ltj_char_range_tl
          }
          { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
      }
    \__ctex_ltj_save_alternate_shape:cn
      { \__ctex_ltj_alternate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
      { \ctex_ltj_clear_alt_font:n { \CJK@encoding/#2/#3/#4 } }
  }
```

\ctex_ltj_set_alternate_shape:nnn

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

```
\NewDocumentCommand \ctex_ltj_set_alternate_shape:nnn
  { m m > { \SplitArgument { 1 } { -> } } m }
  { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
\cs_new_protected:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
  {
    \ctex_ltj_set_alternate_shape:e
      {
        \__ctex_ltj_range_normalization:nn {#3} {#4}
        { \CJK@encoding / \exp_not:n {#2} }
        { \CJK@encoding / \exp_not:n {#1} }
      }
  }
\cs_new_protected:Npn \ctex_ltj_set_alternate_shape:n #1
  {
    \ctex_ltj_set_alt_font:nnnn #1
    \__ctex_ltj_save_alternate_shape:cn
      { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
      { \ctex_ltj_set_alt_font:nnnn #1 }
  }
\cs_generate_variant:Nn \ctex_ltj_set_alternate_shape:n { e }
```

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

```
\cs_new_protected:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
  {
    \tl_map_inline:Nn #3
      {
        \ctex_ltj_set_alternate_shape:n
          {
            ##1
            { \CJK@encoding/#2 }
            { \CJK@encoding/#1 }
          }
      }
  }
```

__ctex_ltj_save_alternate_shape:Nn 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。

```
\cs_new_protected:Npn \__ctex_ltj_save_alternate_shape:Nn #1#2
  {
    \group_begin:
      \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
      \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
      \cs_gset_protected:Npx #1 { \exp_not:o {#1} #2 }
    \group_end:
  }
\cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nn { c }
```

clearalternatefont
resetalternatefont 清除和重置操作总是全局的。

```
\ctex_define:n
  {
    clearalternatefont    .code:n =
      { \clist_map_function:eN {#1} \ctex_ltj_clear_alternate_font:n } ,
    resetalternatefont    .code:n =
      { \clist_map_function:eN {#1} \ctex_ltj_reset_alternate_font:n } ,
    clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
    resetalternatefont .default:n = \l_ctex_ltj_family_tl
  }
\cs_new_protected:Npn \ctex_ltj_clear_alternate_font:n #1
  {
    \group_begin:
      \ctex_ltj_family_if_exist:eNTF {#1} \l__ctex_ltj_base_family_tl
        {
          \cs_if_exist_use:cT
            { \__ctex_ltj_alternate_cs:n { clear / #1 } }
            {
              \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
                {#1} { \l__ctex_ltj_base_family_tl }
              \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
              \selectfont
            }
        }
        { \__ctex_ltj_family_unknown_warning:n {#1} }
    \group_end:
  }
\cs_new_protected:Npn \ctex_ltj_reset_alternate_font:n #1
  {
    \group_begin:
      \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
        {
          \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
          \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
          \selectfont
        }
```

```
    \group_end:
  }
\prop_new:N \g__ctex_ltj_reset_alternate_prop
\cs_generate_variant:Nn \clist_map_function:nN { e }
```

declarecharrange 预先声明字符范围。

```
\ctex_define:n
  {
    declarecharrange .code:n = \ctex_ltj_declare_char_range:e {#1} ,
    declarecharrange .value_required:n = true
  }
\cs_new_protected:Npn \ctex_ltj_declare_char_range:n #1
  { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
\cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { e }
\cs_new_protected:Npn \__ctex_ltj_declare_char_range:nn #1
  { \tl_trim_spaces_apply:nN {#1} \ctex_ltj_declare_char_range:nn }
```

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

```
\cs_new_protected:Npn \ctex_ltj_declare_char_range:nn #1#2
  {
    \tl_clear:N \l__ctex_ltj_char_range_tl
    \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
    \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
    \ctex_ltj_def_char_range_key:n {#1}
    \tl_clear:N \l__ctex_ltj_char_range_tl
  }
\tl_new:N \l__ctex_ltj_char_range_tl
\prop_new:N \g__ctex_ltj_char_range_prop
```

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

```
\NewDocumentCommand \ctex_ltj_save_char_range:n
  { > { \SplitArgument { 1 } { -> } } m }
  { \ctex_ltj_save_char_range:nn #1 }
\cs_new_protected:Npn \ctex_ltj_save_char_range:nn #1#2
  {
    \tl_put_right:Ne \l__ctex_ltj_char_range_tl
      { { \__ctex_ltj_range_normalization:nn {#1} {#2} } }
  }
\cs_new:Npn \__ctex_ltj_range_normalization:nn #1#2
  {
    \tl_if_novalue:nTF {#2}
      {
        { \int_eval:n {#1} }
        { \int_eval:n {#1} }
      }
      {
        { \int_eval:n { \tl_if_blank:nTF {#1} { \c__ctex_ltj_range_min_int } {#1} } }
        { \int_eval:n { \tl_if_blank:nTF {#2} { \c__ctex_ltj_range_max_int } {#2} } }
      }
  }
\int_const:Nn \c__ctex_ltj_range_min_int { "80 }
\int_const:Nn \c__ctex_ltj_range_max_int { \c_max_char_int }
```

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

```
\cs_new_protected:Npn \ctex_ltj_def_char_range_key:n #1
  {
    \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
      {
        \keys_define:nn { ctex_ltj / fontspec }
          { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
      }
  }
```

\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。

```
\cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
  {
    \tl_if_blank:nTF {#2}
      { \clist_set:Nn \l__ctex_ltj_char_range_clist {#1} }
      { \__ctex_ltj_push_alternate_prop:nn {#1} { {#1} #2 } }
  }
```

15.3.4.8　其他设置

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

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

```
\AtBeginDocument
  {
    \ctex_appto_cmd:NnnTF \verbatim@font
      { \char_set_catcode_letter:n { 64 } }
      { \CTEX@verbatim@font@hook }
      { }
      { \ctex_patch_failure:N \verbatim@font }
  }
\cs_new_protected:Npn \CTEX@verbatim@font@hook
  { \ltjsetparameter { autospacing = false , autoxspacing = false } }
```

LuaTeX-ja 的 lltjcore.sty 会将 LaTeX 的 \verb 中 \null （即 \hbox{}） 替换为 \vadjust{}, 因为空 \hbox{} 会阻断 xkanjiskip 的自动插入。由于 ctex 禁用了 ltj-latex, 这里需要自行复制该修复。

```
\@namedef { ver@lltjcore.sty } { }
\if@compatibility \else
  \def \verb
    {
      \relax \ifmmode \hbox \else \leavevmode \vadjust { } \fi
      \bgroup
        \verb@eol@error \let \do \@makeother \dospecials
        \verbatim@font \@noligs
        \language \l@nohyphenation
        \@ifstar \@sverb \@verb
    }
\fi
\patchcmd { \do@noligs } { \kern \z@ } { \vadjust { } } { } { }
```

\@@italiccorr LaTeX 的倾斜校正也要重新定义。

```
\cs_set_eq:NN \@@italiccorr \/
```

\ctex_ltj_set_kanjiskip:N
\ctex_ltj_set_xkanjiskip:N \ltjsetkanjiskip 和 \ltjsetxkanjiskip 是相应的 \ltjsetparameter 的快捷方式, 在使用他们时, 要注意先使用 \ltj@setpar@global。

```
\cs_new_protected:Npn \ctex_ltj_set_kanjiskip:N
```

```
  { \ltj@setpar@global \ltjsetkanjiskip }
\cs_new_protected:Npn \ctex_ltj_set_xkanjiskip:N
  { \ltj@setpar@global \ltjsetxkanjiskip }

⟨@@=ctex⟩
⟨/luatex⟩
```

15.3.5 ctex-engine-uptex.def

```
⟨*uptex|aptex⟩
```

按 CJK 的命名习惯模拟 `\CJKfamily`。

```
\NewDocumentCommand \CJKfamily { m }
  { \kanjifamily {#1} \selectfont }
```

将 upLaTeX 的默认字体由 `mc` 改为 `zhrm`, 并启用 `\jfam`。

```
\DeclareErrorKanjiFont    {JY2}{zhrm}{m}{n}{10}
\DeclareKanjiSubstitution {JY2}{zhrm}{m}{n}
\DeclareKanjiSubstitution {JT2}{zhrm}{m}{n}
\DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
\SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
\jfam \symmincho
```

`\em` 取消 upLaTeX 对 `\em` 使用 `\mcfamily`、`\gtfamily` 命令的重定义, 恢复 LaTeX 2_ε 对 `\em` 的原始定义。如果用户已经重定义了 `\em`, 则新定义保持不变。upLaTeX 2016/05/07u00 的定义有所变化, 这一行为可以由用户通过 platexrelease 包改变, 需要分支处理。

```
\ctex_patch_cmd_once:NnnnTF \em
  { \ExplSyntaxOff }
  { \eminnershape \else \gtfamily \itshape }
  { \eminnershape \else \itshape }
  { }
  {
    \ctex_patch_cmd:Nnn \em
      { \mcfamily \upshape \else \gtfamily \itshape }
      { \eminnershape \else \itshape }
  }
\cs_set_nopar:Npn \eminnershape { \upshape }
```

`\ctex_set_upfamily:nnn` 将 NFSS 字体族 `#1` 设置为 JFM 字体名 `#2`, 粗体形式字体名 `#3`。其中字体名形如 `upzhserif`, 不包括表示方向的后缀 `-h` 与 `-v`。粗体字体名为空时不设置该字形。本命令不设置字体映射, 需要复用已有的字体映射或另行设置。

```
\cs_new_protected:Npn \ctex_set_upfamily:nnn #1#2#3
  {
    \DeclareKanjiFamily{JY2}{#1}{}
    \DeclareKanjiFamily{JT2}{#1}{}
    \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
    \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
    \tl_if_empty:nF {#3}
      {
        \DeclareFontShape{JY2}{#1}{b}{n}{<->~ #3-h}{}
        \DeclareFontShape{JT2}{#1}{b}{n}{<->~ #3-v}{}
        \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
        \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
      }
  }
```

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

```
\cs_new_protected:Npn \ctex_set_upmap:nnn #1#2#3
```

```
  {
    \ctex_set_zhmap:n
      {
        \special { pdf:mapline~#1-h~UniGB-UTF16-H~#2 }
        \special { pdf:mapline~#1-v~UniGB-UTF16-V~#2 }
        \tl_if_empty:nF {#3}
          {
            \special { pdf:mapline~#1b-h~UniGB-UTF16-H~#3 }
            \special { pdf:mapline~#1b-v~UniGB-UTF16-V~#3 }
          }
      }
  }
```

\ctex_set_upmap_unicode:nnn 设置 upTEX 字体映射, 使用 unicode CMap。参数同上。

```
\cs_new_protected:Npn \ctex_set_upmap_unicode:nnn #1#2#3
  {
    \ctex_set_zhmap:n
      {
        \special { pdf:mapline~#1-h~unicode~#2 }
        \special { pdf:mapline~#1-v~unicode~#2 }
        \tl_if_empty:nF {#3}
          {
            \special { pdf:mapline~#1b-h~unicode~#3 }
            \special { pdf:mapline~#1b-v~unicode~#3 }
          }
      }
  }
```

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

```
\cs_new_protected:Npn \ctex_set_upfonts:nnnnnn #1#2#3#4#5#6
  {
    \ctex_set_upmap:nnn { upserif   } {#1} {#2}
    \ctex_set_upmap:nnn { upserifit } {#3} {}
    \ctex_set_upmap:nnn { upsans    } {#4} {#5}
    \ctex_set_upmap:nnn { upmono    } {#6} {}
  }
```

以下命令只能在导言区使用。

```
\@onlypreamble \ctex_set_upfamily:nnn
\@onlypreamble \ctex_set_upmap:nnn
\@onlypreamble \ctex_set_upmap_unicode:nnn
\@onlypreamble \ctex_set_upfonts:nnnnnn
```

everysel 宏包(2011/10/28)未考虑 upLATEX 对 \selectfont 的修改,需要引入 pxeverysel 宏包。

```
\bool_if:NT \c__ctex_everysel_loaded_bool
  { \RequirePackage { pxeverysel } }
```

```
⟨/uptex|aptex⟩
```

15.3.6　修改主要字体命令

修改 \rmfamily 等主要字体命令, 使得中文字体能随西文主要字体更新。使用 LATEX 2020/10/01 提供的 rmfamily 等 NFSS 钩子。

xeCJK 和 zhmCJK 已经有相同的工作, 本段代码不需要对他们使用。

```
⟨*pdftex|luatex|uptex|aptex⟩
⟨pdftex⟩ \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
```

\ctex_provide_font_hook:NNN
\CTEX@rmfamilyhook

给 \rmfamily 等字体命令加钩子，钩子名字统一为 \CTEX@rmfamilyhook 等。

```
\cs_new_protected:Npn \ctex_provide_font_hook:NNN #1#2
  {
    \exp_args:Nc \__ctex_provide_font_hook_aux:NNNN
      { CTEX \cs_to_str:N #2 } #1#2
  }
\cs_new_protected:Npn \__ctex_provide_font_hook_aux:NNNN #1#2#3#4
  {
    \tl_new:N #1
    \exp_args:Ne \ctex_gadd_ltxhook:nn { \cs_to_str:N #2 } {#1}
  }
\ctex_provide_font_hook:NNN \rmfamily \@rmfamilyhook \selectfont
\ctex_provide_font_hook:NNN \sffamily \@sffamilyhook \selectfont
\ctex_provide_font_hook:NNN \ttfamily \@ttfamilyhook \selectfont
```

⟨pdftex|luatex⟩

```
\ctex_provide_font_hook:NNN \normalfont \@defaultfamilyhook \usefont
```

按 CJK 的命名习惯模拟部分命令，并设置默认字体。

```
\tl_if_exist:NF \CJKfamilydefault
  { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
⟨*pdftex|luatex⟩
\tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
\tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
\tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
\tl_gput_right:Nn \CTEX@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
\tl_gput_right:Nn \CTEX@sffamilyhook { \CJKfamily { \CJKsfdefault } }
\tl_gput_right:Nn \CTEX@ttfamilyhook { \CJKfamily { \CJKttdefault } }
\tl_gput_right:Nn \CTEX@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }
⟨/pdftex|luatex⟩
```

upLaTeX 不需要补丁 \normalfont，只需要修改 \kanjifamilydefault。

```
⟨*uptex|aptex⟩
\tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { zhrm } }
\tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { zhsf } }
\tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { zhtt } }
\tl_gput_right:Nn \CTEX@rmfamilyhook { \kanjifamily { \CJKrmdefault } }
\tl_gput_right:Nn \CTEX@sffamilyhook { \kanjifamily { \CJKsfdefault } }
\tl_gput_right:Nn \CTEX@ttfamilyhook { \kanjifamily { \CJKttdefault } }
\tl_gset:Nn \kanjifamilydefault { \CJKfamilydefault }
⟨/uptex|aptex⟩
```

zhmCJK 判断结束。

⟨pdftex⟩

```
\fi:
```

使修改立刻生效，保证导言区字体族正确。

```
\normalfont
```

在导言区末尾更新 \CJKfamilydefault，pdfTeX已经在之前使用过此处代码。

⟨!pdftex⟩

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

\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。

```
\cs_new_protected:Npn \ctex_update_default_family:
  {
    \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
      {
        \group_begin:
          \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
          \tl_gset:Ne \CJKfamilydefault
```

```
            {
              \str_case:onF { \familydefault }
                {
                  { \rmdefault } { \exp_not:N \CJKrmdefault }
                  { \sfdefault } { \exp_not:N \CJKsfdefault }
                  { \ttdefault } { \exp_not:N \CJKttdefault }
                }
                { \CJKfamilydefault }
            }
          \group_end:
        }
```

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

```
⟨luatex⟩    \ctex_ltj_ensure_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。

```
\tl_new:N \l__ctex_family_default_init_tl
\cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
\tl_set:Ne \l__ctex_family_default_init_tl
  {
    \exp_not:N \__ctex_family_default_wrap:n
      { \exp_not:o { \CJKfamilydefault } }
  }
\tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl

⟨/pdftex|luatex|uptex|aptex⟩
```

15.3.7　hyperref 兼容性处理

现在处理各个引擎下的 PDF 中文书签问题。根据编译引擎与文件编码的不同，ctex 向 hyperref 传递适当的参数，完成中文书签的正确设置。用户仍需要自己载入 hyperref 宏包。

`\ctex_hypersetup:n` 如果已经载入 hyperref 宏包，则直接使用其定义设置选项；否则 `\ctex_hypersetup:n` 的效果与 `\PassOptionsToPackage` 一致，只传递宏包参数。如果用户不载入 hyperref 宏包，相关参数即被丢弃。

```
⟨*class|ctex⟩
\@ifpackageloaded { hyperref }
  {
    \cs_new_protected:Npn \ctex_hypersetup:n #1
      { \hypersetup {#1} }
  }
  {
    \cs_new_protected:Npn \ctex_hypersetup:n #1
      { \PassOptionsToPackage {#1} { hyperref } }
  }
⟨/class|ctex⟩
```

在 pdfTeX 下使用 GBK 编码，DVIPDFMx 驱动可以直接用它的 `\special` 命令，其他模式用 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。hyperref 2021-02-04 版开始默认设置 `unicode` 为 `true`，对于 DVIPDFMx 驱动，我们需要禁用这个设置，为此设置 `pdfencoding` 为 `pdfdoc`。

`driverfallback` 是 hyperref 的加载选项，加载后即被禁用，不能通过 `\hypersetup` 设置。因此当 hyperref 已被其他宏包（如 beamer）加载时，只能跳过此选项。

```
⟨*pdftex⟩
```

```
\@ifpackageloaded { hyperref }
  { }
  { \PassOptionsToPackage { driverfallback = dvipdfmx } { hyperref } }
\str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
  {
    \ctex_hypersetup:n { CJKbookmarks = true }
    \sys_if_output_pdf:TF
      { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
      {
        \ctex_hypersetup:n { pdfencoding = pdfdoc }
        \ctex_at_end_package:nn { hyperref }
          {
            \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
              {
                \ctex_at_shipout_first:n
                  { \special { pdf:tounicode~GBK-EUC-UCS2 } }
              }
              { \RequirePackage { xCJK2uni } }
          }
      }
  }
  { \ctex_hypersetup:n { pdfencoding = unicode } }
⟨/pdftex⟩
```

XeTeX 和 LuaTeX统一设置 pdfencoding 为 unicode。

```
⟨*xetex|luatex⟩
\ctex_hypersetup:n { pdfencoding = unicode }
⟨/xetex|luatex⟩
```

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

```
⟨*uptex|aptex⟩
\@ifpackageloaded { hyperref }
  { \ctex_hypersetup:n { pdfencoding = pdfdoc } }
  {
    \ctex_hypersetup:n
      {
        driverfallback = dvipdfmx ,
        pdfencoding    = pdfdoc
      }
  }
\ctex_at_end_package:nn { hyperref }
  {
    \ctex_at_shipout_first:n
      { \special { pdf:tounicode~UTF8-UTF16 } }
  }
⟨/uptex|aptex⟩

⟨*pdftex|xetex|luatex|uptex|aptex⟩
```

15.3.8　CJKfntef、xeCJKfntef 相关设置

对 pdfTeX 与 XeTeX 引擎, 分别在 CJKfntef、xeCJKfntef 宏包的末尾关闭彩色显式等多余格式。

```
⟨*pdftex⟩
\ctex_at_end_package:nn { CJKfntef }
  {
    \normalem
    \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
      { \tl_clear:c { CJK#1color } }
⟨/pdftex⟩
⟨*xetex⟩
```

```
\ctex_at_end_package:nn { xeCJKfntef }
  {
    \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
      {
        \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
          { \xeCJKsetup { #1 / format = { } } }
      }
      {
        \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
          { \tl_clear:c { CJK#1color } }
      }
⟨/xetex⟩
⟨*pdftex|xetex⟩
    \clist_map_inline:nn
      { underdot , underline , underdblline , underwave , sout , xout }
      { \__ctex_clear_fntef_color:n {#1} }
  }
⟨/pdftex|xetex⟩
```

15.3.9　\ccwd 的更新

\ctex_update_ccwd:
\ccwd

```
\cs_new_protected:Npn \ctex_update_ccwd:
⟨*pdftex|xetex⟩
  {
    \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
    \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
  }
⟨/pdftex|xetex⟩
⟨*luatex⟩
  { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
⟨/luatex⟩
⟨*uptex|aptex⟩
  { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
⟨/uptex|aptex⟩
\dim_new:N \ccwd
```

\ctex_update_ccglue:

更新字间距。

```
\cs_new_protected:Npn \ctex_update_ccglue:
⟨*pdftex|xetex⟩
  {
    \cs_set_protected:Npn \CJKglue
      { \skip_horizontal:N \l__ctex_ccglue_skip }
    \cs_if_exist:NT \__ctex_ccglue:
      { \cs_set_eq:NN \__ctex_ccglue: \CJKglue }
  }
⟨/pdftex|xetex⟩
⟨*luatex⟩
  { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
⟨/luatex⟩
⟨*uptex|aptex⟩
  { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
⟨/uptex|aptex⟩
\skip_new:N \l__ctex_ccglue_skip
```

\ctex_if_ccglue_touched_p:
\ctex_if_ccglue_touched:*TF*

检查用户是否修改过汉字间距。

```
\prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
  {
⟨*pdftex|xetex⟩
    \if_meaning:w \CJKglue \__ctex_ccglue:
      \prg_return_false: \else: \prg_return_true: \fi:
⟨/pdftex|xetex⟩
⟨*luatex⟩
    \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
```

```
      { \prg_return_false: } { \prg_return_true: }
〈/luatex〉
〈*uptex|aptex〉
    \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
      { \prg_return_false: } { \prg_return_true: }
〈/uptex|aptex〉
  }
〈pdftex|xetex〉\ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
〈*pdftex|xetex〉
\ctex_at_end:n
  {
    \cs_set_protected:Npn \__ctex_update_stretch_auxii:
      {
        \ctex_if_ccglue_touched:TF
          { \ctex_update_ccwd: }
          { \__ctex_update_stretch_auxiii: }
      }
  }
〈/pdftex|xetex〉
```

\ctex_update_em_unit: 将当前汉字的宽度保存到 \ccwd 中备用。不采用 1em，因为这时的 1em 实际上来自西文字体的信息，未必等于汉字的宽度，这似乎在传统的 .tfm 字体上表现更明显。在 pdfTEX 和 XETEX 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 XETEX 可以用 \fontcharwd 来改进。而在 pdfTEX 下，若使用 zhmetrics 技术，所有的汉字共享同一个 .tfm，\fontcharwd 也就没有意义。在 LuaTEX 下，LuaTeX-ja 总是按照 JFM 中的设置输出汉字的宽度，可以直接用 \zw 作为汉字宽度。upTEX 可以直接使用原生的长度单位 zw。

```
\cs_new_protected:Npn \ctex_update_em_unit:
〈pdftex|xetex〉  { \dim_set:Nn \ccwd { \f@size \p@ } }
〈luatex〉  { \dim_set:Nn \ccwd { \zw } }
〈uptex|aptex〉  { \dim_set:Nn \ccwd { 1zw } }
```

15.3.10　其他

\ctex_add_to_selectfont:n
\CTEX@selectfont@hook

```
\cs_new_protected:Npn \ctex_add_to_selectfont:n #1
  {
    \cs_set_protected:Npx \CTEX@selectfont@hook
      { \exp_not:o { \CTEX@selectfont@hook #1 } }
  }
\cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
```

使用 everysel 包的情况。\EverySelectfont 直到文档开始时才有效。为了 \ccwd 和 LuaTeX-ja 的字体设置在导言区也可用，我们还需要在这里手工修改 \selectfont。everysel 宏包会用 \CheckCommand 来检查 \selectfont 是否为标准定义。我们修改了 \selectfont，所以会给出一个警告。为了消除这个警告，在它检查之前，还原本来定义。pxeverysel 宏包取消了检查，但也需要恢复定义，避免重复使用钩子。KOMA-Script 宏包也会进行检查，我们需要小心处理。

```
\bool_if:NTF \c__ctex_everysel_loaded_bool
  {
    \cs_if_free:NF \@EverySelectfont@Init
      {
        \group_begin:
        \cs_set:Npn \__ctex_tmp:N #1
          {
            \tl_set:Nn \l__ctex_tmp_tl {#1}
            \cs_new_eq:NN \CTEX@selectfont@save #1
            \cs_new_protected:Npn \__ctex_restore_selectfont:
              {
                \cs_if_free:NF \scr@new@selectfont
                  {
```

CJK 直接修改 \selectfont 和 pxeverysel 的补丁，会使 KOMA-Script 的 \par@update 失效。

```
⟨*pdftex|uptex|aptex⟩
                    \cs_if_free:NF \par@update
⟨*uptex|aptex⟩
                      {
                        \tl_put_right:Nn \@EverySelectfont@Init
                          { \tl_put_right:Nn #1 { \par@update } }
                      }
⟨/uptex|aptex⟩
⟨*pdftex⟩
                      { \tl_put_right:Nn #1 { \par@update } }
                    \cs_set_eq:NN \scr@selectfont \CTEX@selectfont@save
⟨/pdftex⟩
⟨/pdftex|uptex|aptex⟩
                    \cs_set_eq:NN \scr@new@selectfont #1
⟨!pdftex⟩           \cs_set_eq:NN \CTEX@selectfont@save \scr@selectfont
                  }
                \tl_put_left:Nn \@EverySelectfont@Init
                  { \cs_set_eq:NN #1 \CTEX@selectfont@save }
                \cs_undefine:N \__ctex_restore_selectfont:
              }
          }
        \ctex_parse_name:NN \__ctex_tmp:N \selectfont
      \exp_last_unbraced:NNo \group_end:
      \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
        { \ExplSyntaxOff }
        { \size@update }
        { \CTEX@selectfont@hook \size@update }
        { \__ctex_restore_selectfont: }
        { \ctex_patch_failure:N \selectfont }
      }
    \cs_new_protected:Npn \ctex_gadd_selectfont_hook:n
      { \EverySelectfont }
  }
```

使用 LaTeX 2021-06-01 的新钩子，不使用 everysel 包的情况。

```
  {
    \cs_new_protected:Npn \ctex_gadd_selectfont_hook:n
      { \ctex_gadd_ltxhook:nn { selectfont } }
  }
```

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

```
⟨*pdftex⟩
\ctex_gadd_selectfont_hook:n
  { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
⟨/pdftex⟩
⟨*xetex|luatex|uptex|aptex⟩
\ctex_gadd_selectfont_hook:n { \CTEX@selectfont@hook }
⟨/xetex|luatex|uptex|aptex⟩
```

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

```
⟨*luatex⟩
\ctex_add_to_selectfont:n
  {
    \ctex_ltj_select_font:
    \ctex_ltj_select_alternate_font:
  }
\tl_set:Nn \CJK@family { song } \selectfont
\tl_clear:N \CJK@family
⟨/luatex⟩
```

\ctex_update_xkanjiskip:
\l__ctex_xkanjiskip_skip

upTeX 和 LuaTeX-ja 对 \xkanjiskip 都是即时赋值。单位 zw 与字体相关，因此需要每次 \selectfont 的时候更新一次 \xkanjiskip。如果用户设置过 \xkanjiskip，就不更新。注意，

同 TEX 的 \baselineskip 一样, 如果在一个段落内多次设置了 \kanjiskip 或 \xkanjiskip, 只有最后的设置会影响全段。

```
⟨*luatex|uptex|aptex⟩
\cs_new_protected:Npn \ctex_update_xkanjiskip:
  {
    \skip_if_eq:nnT
      ⟨luatex⟩ { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
    ⟨uptex|aptex⟩ { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
      {
        \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
        ⟨luatex⟩ \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
        ⟨uptex|aptex⟩ \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
      }
  }
\tl_new:N \l__ctex_xkanjiskip_tl
\tl_set:Nn \l__ctex_xkanjiskip_tl
  ⟨luatex⟩ { .25\zw plus 1pt minus 1pt }
  ⟨uptex|aptex⟩ { .25zw plus 1pt minus 1pt }
\skip_new:N \l__ctex_xkanjiskip_skip
\skip_set:Nn \l__ctex_xkanjiskip_skip
  ⟨luatex⟩ { \ltjgetparameter { xkanjiskip } }
  ⟨uptex|aptex⟩ { \tex_xkanjiskip:D }

\ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
⟨/luatex|uptex|aptex⟩
```

\cht
\cdp
\cwd
\ctex_update_kanjisize:

分别从 .jfm 中读取字符高度、深度和宽度, 目前仅考虑横排的情况。

```
⟨*luatex⟩
\dim_new:N \cht
\dim_new:N \cdp
\dim_new:N \cwd
\group_begin:
\char_set_catcode_space:n { 32 }
\lua_now:e
  {
    local nulltable = { }
    local fmt = luatexja.jfont.font_metric_table
    local getattribute = tex.getattribute
    local setdimen = tex.setdimen
    ctex.newluacmd("ctex_update_kanjisize:", function ()
      local ft = fmt[getattribute("ltj@curjfnt")] or nulltable
      local ft = ft and ft.char_type or nulltable
      local fk = ft and ft[0] or nulltable
      setdimen("cht", fk.height or 0)
      setdimen("cdp", fk.depth or 0)
      setdimen("cwd", fk.width or ft.zw or 0)
    end, "global", "protected")
  }
\group_end:
\ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
⟨/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、upTEX 下是无效选项。

```
\ctex_define:n
  {
⟨*pdftex|xetex⟩
    space .choice: ,
    space / true  .code:n =
      ⟨pdftex⟩ { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
      ⟨xetex⟩ { \xeCJKsetup { CJKspace = true } } ,
    space / auto  .code:n =
      ⟨pdftex⟩ { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
```

```
〈xetex〉      { \xeCJKsetup { CJKspace = false } } ,
    space / false .code:n =
〈pdftex〉     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
〈xetex〉      { \xeCJKsetup { CJKspace = false } } ,
    space .default:n = { true } ,
    space .initial:n = { auto }
〈/pdftex|xetex〉
〈*luatex|uptex|aptex〉
    space .code:n =
      { \msg_warning:nn { ctex } { invalid-option } }
〈/luatex|uptex|aptex〉
  }
```

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。upTEX 暂时无效。

```
\ctex_define:n
  {
    punct .code:n =
      {
        \tl_set:Ne \l__ctex_punct_tl {#1}
〈pdftex〉       \punctstyle { \l__ctex_punct_tl }
〈xetex〉        \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
〈luatex〉       \ctex_set_jfm:o { \l__ctex_punct_tl }
〈uptex|aptex〉  \msg_warning:nn { ctex } { invalid-option }
      } ,
    punct .default:n = { quanjiao } ,
  }
```

experiment 实验性选项。experiment/CJKecglue 用于统一设置 CJK 文字与西文之间的间距。XeTEX 下转发给 xeCJK，LuaTEX 和 upTEX 下设置 xkanjiskip，pdfTEX 下不支持。

experiment/CJKecglue

```
\ctex_define:n
  { experiment .meta:nn = { ctex / experiment } {#1} }
\keys_define:nn { ctex / experiment }
  {
〈*xetex〉
    CJKecglue .code:n =
      { \xeCJKsetup { CJKecglue = { \skip_horizontal:n {#1} } } } ,
〈/xetex〉
〈*luatex|uptex|aptex〉
    CJKecglue .code:n =
      {
        \tl_set:Nn \l__ctex_xkanjiskip_tl {#1}
        \skip_set:Nn \l__ctex_xkanjiskip_skip {#1}
〈luatex〉       \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
〈uptex|aptex〉  \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
      } ,
〈/luatex|uptex|aptex〉
〈*pdftex〉
    CJKecglue .code:n =
      { \msg_warning:nn { ctex } { CJKecglue-unsupported } } ,
〈/pdftex〉
    CJKecglue .value_required:n = true
  }

〈/pdftex|xetex|luatex|uptex|aptex〉
```

15.3.11　载入引擎定义文件

最后载入各个编译引擎的定义文件。

```
〈class|ctex〉 \ctex_file_input:n { \c__ctex_engine_file_str }
```

15.4　用户设置接口

\ctexset
```
⟨*class|ctex|ctexheading⟩
\NewDocumentCommand \ctexset { } { \ctex_set:n }
⟨/class|ctex|ctexheading⟩
```

\CTEXsetup
\CTEXoptions
过时命令。

```
⟨*class|ctex⟩
\NewDocumentCommand \CTEXsetup { +O { } > { \TrimSpaces } m }
  {
    \tl_if_blank:nTF {#1}
      { \ctex_deprecated_command:Nn \CTEXsetup { } }
      {
        \ctex_deprecated_command:Nn \CTEXsetup
          { \ctexset {~#2~=~{~#1~}~}~is~set. }
        \ctex_set:nn {#2} {#1}
      }
  }
\NewDocumentCommand \CTEXoptions { +O { } }
  {
    \tl_if_blank:nTF {#1}
      { \ctex_deprecated_command:Nn \CTEXoptions { } }
      {
        \ctex_deprecated_command:Nn \CTEXoptions
          { \ctexset {~#1~}~is~set. }
        \ctex_set:n {#1}
      }
  }
```

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。

```
\ctex_define:n
  {
    autoindent .choice: ,
    autoindent .default:n = { true } ,
    autoindent / true    .code:n =
      {
        \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
        \ctex_select_size:
      } ,
    autoindent / false   .code:n =
      { \tl_clear:N \l__ctex_autoindent_tl } ,
    autoindent / unknown .code:n =
      {
        \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
        \ctex_select_size:
      }
  }
```

\CTEXsetfont
无论字体大小是否变化都更新相关信息。

```
\NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
\cs_new_protected:Npn \ctex_select_size:
  { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }
```

\ctex_update_size:
在字号变化时更新 \ccwd、\parindent 和汉字间距。字距为零则恢复正常设置。

```
\cs_new_protected:Npn \ctex_update_size:
  {
    \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
      {
```

```
        \ctex_update_stretch:
        \ctex_update_parindent:
      }
      { \ctex_update_ziju: }
  }
\tl_const:Ne \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
\tl_new:N \l__ctex_ziju_tl
\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。

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

linestretch 若行宽不是汉字宽度的整数倍, 自然要求伸展它们之间的差。这里设置的是在此基础上的额外伸展量。初始化为一个汉字的宽度。若设置为 \maxdimen, 则禁用此功能。参数的默认单位是汉字的宽度 \ccwd。

```
\ctex_define:n
  {
    linestretch .code:n =
      {
        \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
        \ctex_select_size:
      } ,
    linestretch .value_required:n = true
  }
\tl_new:N \l__ctex_line_stretch_tl
\tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }
```

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

```
\cs_new_protected:Npn \ctex_update_stretch:
  {
    \ctex_update_em_unit:
    \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
    \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
      { \__ctex_update_stretch_auxi: }
      { \__ctex_update_stretch_auxii: }
  }
\cs_new_protected:Npn \__ctex_update_stretch_auxi:
  {
    \ctex_if_ccglue_touched:TF
      { \ctex_update_ccwd: }
      {
        \dim_set:Nn \l__ctex_tmp_dim
          { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
        \skip_set:Nn \l__ctex_ccglue_skip
          { \c_zero_dim plus .08 \l__ctex_tmp_dim }
        \ctex_update_ccglue:
      }
  }
\cs_new_protected:Npn \__ctex_update_stretch_auxii:
  { \__ctex_update_stretch_auxiii: }
\cs_new_protected:Npn \__ctex_update_stretch_auxiii:
  {
    \int_set:Nn \l__ctex_tmp_int
```

```
      { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
    \tex_divide:D \l__ctex_tmp_int \ccwd
    \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
      {
        \skip_set:Nn \l__ctex_ccglue_skip
          {
            \c_zero_dim plus \dim_eval:n
              {
                ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
                \l__ctex_tmp_int
              }
          }
      }
      { \skip_zero:N \l__ctex_ccglue_skip }
    \ctex_update_ccglue:
  }
```

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

```
\cs_new_protected:Npn \ctex_update_parindent:
  {
    \tl_if_empty:NF \l__ctex_autoindent_tl
      {
        \dim_compare:nNnF \parindent = \c_zero_dim
          { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
      }
  }
```

\ziju 若参数为 0, 则恢复正常间距。

```
\NewDocumentCommand \ziju { m }
  { \exp_args:Ne \ctex_ziju:n {#1} \tex_ignorespaces:D }
\cs_new_protected:Npn \ctex_ziju:n #1
  {
    \tl_set:Ne \l__ctex_ziju_tl { \fp_eval:n {#1} }
    \ctex_select_size:
  }
```

\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。

```
\cs_new_protected:Npn \ctex_update_ziju:
  {
    \ctex_update_em_unit:
    \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
    \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
    \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 TEX 决定伸展还是收缩。

```
      {
        \dim_set:Nn \l__ctex_tmp_dim
          { \linewidth - \ccwd + \l__ctex_ziju_dim }
        \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
        \tex_divide:D \l__ctex_tmp_int \ccwd
        \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。

```
        \dim_compare:nNnF \parindent = \c_zero_dim
          {
            \int_compare:nNnF \l__ctex_tmp_int < 3
              { \int_sub:Nn \l__ctex_tmp_int { 2 } }
```

```
          }
        \skip_set:Nn \l__ctex_ccglue_skip
          {
            \l__ctex_ziju_dim
            plus  \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
            minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
              { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
          }
      }
      { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
    \ctex_update_ccglue:
```

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

```
    \ctex_update_parindent:
  }
\dim_new:N \l__ctex_ziju_dim
```

\CTEXindent
\CTEXnoindent

过时命令。

```
\NewDocumentCommand \CTEXindent { }
  {
    \ctex_deprecated_command:Nn \CTEXindent
      { \parindent is~set~to~2\ccwd. }
    \ctex_update_ccwd:
    \dim_set:Nn \parindent { 2 \ccwd }
  }
\NewDocumentCommand \CTEXnoindent { }
  {
    \ctex_deprecated_command:Nn \CTEXnoindent
      { \parindent is~set~to~0pt. }
    \dim_zero:N \parindent
  }
```

15.6　中文数字与日期

需要注意的是，ltkeys 设置的选项列表是 \@raw@opt@⟨\@currname⟩.⟨\@currext⟩，该列表不会将 \PassOptionsToPackage 传递的选项完全展开。

```
\exp_args:Ne \PassOptionsToPackage
  { encoding = \g__ctex_encoding_tl }
  { zhnumber }
\RequirePackage { zhnumber }
```

\chinese

```
\cs_new:Npn \chinese { \zhnum_counter:n }
\cs_new_eq:NN \@chinese \@zhnum
\cs_new_eq:NN \Chinese \chinese
\cs_new_eq:NN \CTEXcounter \use_none:n
```

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

```
\ctex_at_end_package:nn { enumitem }
  {
    \cs_if_free:NF \AddEnumerateCounter
      {
        \AddEnumerateCounter * { \zhnum }   { \@zhnum } { 1 }
        \AddEnumerateCounter * { \zhdig }   { \@zhdig } { 1 }
        \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
      }
  }
```

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

      today
\cs_set_eq:NN \CTEX@todayold \today
\ctex_define:n
  {
    today .choice: ,
    today / old     .code:n =
      { \cs_set_eq:NN \today \CTEX@todayold } ,
    today / small   .code:n =
      {
        \cs_set_eq:NN \today \zhtoday
        \zhnumsetup { time = Arabic }
      } ,
    today / big     .code:n =
      {
        \cs_set_eq:NN \today \zhtoday
        \zhnumsetup { time = Chinese }
      } ,
    today / unknown .code:n =
      { \msg_error:nne { ctex } { today-undef } {#1} }
  }
\msg_new:nnnn { ctex } { today-undef }
  { Today~format~`#1'~is~undefined. }
  { Available~today~formats~are~`old',~`small',~and~`big'. }
```

15.7　其他中文标题定义

\proofname　\proofname 未在标准文档类中定义，需要确保它非空。

```
\tl_if_exist:NF \proofname
  {
    \tl_new:N \proofname
    \tl_set:Nn \proofname { Proof }
  }

\ctex_define:n
  {
    contentsname    .tl_set:N = \contentsname ,
    listfigurename  .tl_set:N = \listfigurename ,
    listtablename   .tl_set:N = \listtablename ,
    figurename      .tl_set:N = \figurename ,
    tablename       .tl_set:N = \tablename ,
    abstractname    .tl_set:N = \abstractname ,
    indexname       .tl_set:N = \indexname ,
    appendixname    .tl_set:N = \appendixname ,
    proofname       .tl_set:N = \proofname ,
⟨article⟩
    bibname         .tl_set:N = \refname
⟨book|report⟩
    bibname         .tl_set:N = \bibname
⟨*beamer⟩
    algorithmname   .tl_set:N = \algorithmname ,
    bibname         .tl_set:N = \bibname ,
    refname         .tl_set:N = \refname ,
    continuation    .tl_set:N = \insertcontinuationtext
⟨/beamer⟩
  }

⟨*ctex⟩
\msg_new:nnn { ctex } { ctexbibname }
  {
    Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.\\
    The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
  }
\tl_if_exist:NTF \insertcontinuationtext
```

```
  {
    \ctex_define:n
      {
        algorithmname .tl_set:N = \algorithmname ,
        bibname       .tl_set:N = \bibname ,
        refname       .tl_set:N = \refname ,
        continuation  .tl_set:N = \insertcontinuationtext
      }
  }
  {
    \tl_if_exist:NTF \bibname
      { \ctex_define:n { bibname .tl_set:N = \bibname } }
      {
        \tl_if_exist:NTF \refname
          { \ctex_define:n { bibname .tl_set:N = \refname } }
          {
            \msg_warning:nn { ctex } { ctexbibname }
            \ctex_define:n { bibname .tl_set:N = \ctexbibname }
          }
      }
  }
⟨/ctex⟩

⟨/class|ctex⟩
```

15.8　中文化的标题结构

本节内容在 CTEX 文档类或打开 heading 选项下生效。

```
⟨*class|heading⟩
```

15.8.1　定义标题格式选项

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

```
⟨*article|book|report⟩
\seq_const_from_clist:Nn \c__ctex_section_headings_seq
  { section , subsection , subsubsection , paragraph , subparagraph }
⟨/article|book|report⟩
```

\c__ctex_headings_seq　保存各级标题名字。

```
⟨article|book|report|beamer⟩ \seq_const_from_clist:Nn \c__ctex_headings_seq
⟨*article|book|report⟩
  {
    part ,
⟨book|report⟩   chapter ,
    section , subsection , subsubsection , paragraph , subparagraph
  }
⟨/article|book|report⟩
⟨beamer⟩  { part , section , subsection }
```

__ctex_initial_heading:n

```
\cs_new_protected:Npn \__ctex_initial_heading:n #1
  {
    \tl_new:c { CTEX@pre#1 }
    \tl_new:c { CTEX@post#1 }
    \tl_const:ce { CTEXthe#1 }
      {
        \exp_not:c { CTEX@pre#1 }
        \exp_not:c { CTEX@the#1 }
        \exp_not:c { CTEX@post#1 }
      }
    \tl_const:ce { CTEX@#1name }
```

```
      {
        \group_begin:
          \exp_not:c { CTEX@#1@nameformat }
            {
              \exp_not:c { CTEX@pre#1 }
              \exp_not:N \tl_if_empty:NTF
              \exp_not:c { CTEX@#1@numberformat }
                { \exp_not:c { CTEX@the#1 } }
                {
                  \group_begin:
                    \exp_not:c { CTEX@#1@numberformat }
                    \exp_not:c { CTEX@the#1 }
                  \group_end:
                }
              \exp_not:c { CTEX@post#1 }
            }
        \group_end:
      }
  }
```

__ctex_def_heading_keys:n

```
\cs_new_protected:Npn \__ctex_def_heading_keys:n #1
  {
    \exp_args:NNe \tl_put_right:Nn \l__ctex_tmp_tl
      {
        #1                   .meta:nn = { ctex / #1 } { ##1 } ,
        #1 / name             .code:n =
          { \ctex_assign_heading_name:nn {#1} { ##1 } } ,
        #1 / number         .tl_set:N = \exp_not:c { CTEX@the#1 } ,
        #1 / beforeskip     .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
        #1 / afterskip      .tl_set:N = \exp_not:c { CTEX@#1@afterskip} ,
        #1 / indent         .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
        #1 / numbering    .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
        #1 / numbering     .initial:n = true ,
        #1 / beforeskip    .initial:n = \c_zero_skip ,
        #1 / afterskip     .initial:n = \c_zero_skip ,
        #1 / indent        .initial:n = \c_zero_dim ,
        #1 / beforeskip    .value_required:n = true ,
        #1 / afterskip     .value_required:n = true ,
        #1 / indent        .value_required:n = true ,
⟨*article|book|report⟩
        #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
        #1 / fixskip     .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
        #1 / hang        .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
        #1 / hang         .initial:n = true ,
        #1 / runin       .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
        #1 / tocline      .cs_set:Np = \exp_not:c { CTEX@#1@tocline } ##1##2 ,
        \__ctex_plus_key_aux:nn {#1} { break } ,
⟨/article|book|report⟩
        \__ctex_plus_key_aux:nn {#1} { format } ,
        \__ctex_plus_key_aux:nn {#1} { nameformat } ,
        \__ctex_plus_key_aux:nn {#1} { numberformat } ,
        \__ctex_plus_key_aux:nn {#1} { titleformat } ,
        \__ctex_plus_key_aux:nn {#1} { aftername } ,
        \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
      }
  }
\cs_new:Npn \__ctex_plus_key_aux:nn #1#2
  {
    #1 / #2   .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
    #1 / #2 +   .code:n =
      { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { ##1 } } ,
    #1 / #2 ~ + .code:n =
      { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { ##1 } }
  }
```

\ctex_assign_heading_name:nn
__ctex_assign_heading_name:nnn

name 的值是一个至多两个元素的逗号分隔列表。由于 LaTeX3 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。

```
\NewDocumentCommand \ctex_assign_heading_name:nn
  { m > { \SplitArgument { 1 } { , } } +m }
  { \__ctex_assign_heading_name:nnn {#1} #2 }
\cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
  {
    \tl_set:cn { CTEX@pre#1 } {#2}
    \tl_if_novalue:nTF {#3}
      { \tl_clear:c { CTEX@post#1 } }
      { \tl_set:cn { CTEX@post#1 } {#3} }
  }
```

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

只在 ctexbook 和 ctexrep 下有定义。

```
\group_begin:
⟨*book|report⟩
\tl_set:Nn \l__ctex_tmp_tl
  {
    part    / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
    chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
    chapter / lofskip   .tl_set:N = \CTEX@chapter@lofskip ,
    chapter / lotskip   .tl_set:N = \CTEX@chapter@lotskip ,
    chapter / lofskip  .initial:n = \c_zero_skip ,
    chapter / lotskip  .initial:n = \c_zero_skip ,
    chapter / lofskip  .value_required:n = true ,
    chapter / lotskip  .value_required:n = true ,
  }
⟨/book|report⟩
⟨*article|beamer⟩
\tl_clear:N \l__ctex_tmp_tl
⟨/article|beamer⟩
```

定义标题键值选项。

```
\seq_map_inline:Nn \c__ctex_headings_seq
  {
    \__ctex_initial_heading:n {#1}
    \__ctex_def_heading_keys:n {#1}
  }
\exp_args:NNo \group_end: \ctex_define:n { \l__ctex_tmp_tl }
```

\CTEX@heading@format@initial

标题格式的一些初始设置,包括恢复默认字体,并禁用自动调整首行缩进,禁止在标题中分页。同时用 \noindent 抑制首行缩进并进入水平模式。统一在各级标题的 format 选项之前使用。

```
\cs_new_protected:Npn \CTEX@heading@format@initial
  {
    \normalfont
    \tl_clear:N \l__ctex_autoindent_tl
    \int_set:Nn \tex_interlinepenalty:D { 10 000 }
    \tex_noindent:D
  }
```

\ctex_indent_box:n

设置 \parindent,并插入用于产生缩进的盒子,如果缩进为 0,就不插入。

```
\cs_new_protected:Npn \ctex_indent_box:n #1
  {
    \dim_set:Nn \tex_parindent:D {#1}
    \__ctex_insert_indent:
  }
\cs_new_protected:Npn \__ctex_insert_indent:
  {
```

```
    \dim_compare:nNnF \tex_parindent:D = \c_zero_dim
      { \tex_indent:D }
  }
\cs_new_eq:NN \CTEX@indentbox \ctex_indent_box:n
```

15.8.2　标准标题命令的修改

```
⟨*article|book|report⟩
```

\CTEX@fixtopskip 修正 book 和 report 类的 \part 和 \chapter 标题之前的多余空行。

```
⟨*book|report⟩
\cs_new_protected:Npn \CTEX@fixtopskip
  {
    \CTEX@fixheadingskip
    \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
      { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
  }
⟨/book|report⟩
```

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

```
\cs_new_protected:Npn \CTEX@fixheadingskip
  {
    \par
    \dim_set:Nn \tex_prevdepth:D { -1000pt }
    \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
  }
\skip_new:N \l__ctex_heading_skip
\cs_new_protected:Npn \CTEX@setheadingskip
  { \skip_set:Nn \l__ctex_heading_skip }
\cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip
```

\CTEX@setthispagestyle 将 \CTEX@(part|chapter)@pagestyle 展开后再传给 \thispagestyle。

```
⟨*book|report⟩
\cs_new_protected:Npn \CTEX@setthispagestyle #1
  { \exp_args:Ne \thispagestyle { \use:c { CTEX@#1@pagestyle } } }
⟨/book|report⟩
```

\partmark 提供 \partmark。

```
\ProvideDocumentCommand \partmark { m }
  { \markboth { } { } }
```

\CTEXifname 用于判断当前标题是否有编号。
\CTEX@ifnametrue
\CTEX@ifnamefalse

```
\cs_new_eq:NN \CTEXifname \use_ii:nn
\cs_new_protected:Npn \CTEX@ifnametrue
  { \cs_set_eq:NN \CTEXifname \use_i:nn }
\cs_new_protected:Npn \CTEX@ifnamefalse
  { \cs_set_eq:NN \CTEXifname \use_ii:nn }
```

\CTEX@addloflotskip 往插图和表格目录中加入额外间距。如果间距为零，则不加入。

```
⟨*book|report⟩
\cs_new_protected:Npn \CTEX@addloflotskip #1
  {
    \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
```

```
    \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
      {
        \addtocontents { lof }
          { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
      }
    \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
    \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
      {
        \addtocontents { lot }
          { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
      }
  }
⟨/book|report⟩
```

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

15.8.2.1　part 的标题

\part
```
⟨*article⟩
\renewcommand\part{%
  \if@noskipsec \leavevmode \fi
  \par
  \CTEX@part@break
% \addvspace{4ex}%
  \CTEX@setheadingskip \CTEX@part@beforeskip
  \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
  \addvspace \CTEX@headingskip
  \ifodd \CTEX@part@afterindent
   \@afterindenttrue
  \else
   \@afterindentfalse
  \fi
  \secdef\@part\@spart}
⟨/article⟩
⟨*book|report⟩
\renewcommand\part{%
% \if@openright
%   \cleardoublepage
% \else
%   \clearpage
% \fi
  \CTEX@part@break
% \thispagestyle{plain}%
  \CTEX@setthispagestyle{part}%
  \if@twocolumn
    \onecolumn
    \@tempswatrue
  \else
    \@tempswafalse
  \fi
% \null\vfil
  \CTEX@setheadingskip \CTEX@part@beforeskip
  \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
  \vspace*{\CTEX@headingskip}%
  \secdef\@part\@spart}
⟨/book|report⟩
```

\@part
```
⟨*article⟩
\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
    \ifodd \CTEX@part@numbering
      \CTEX@ifnametrue
      \refstepcounter{part}%
%     \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
```

```
    \else
      \CTEX@ifnamefalse
      \CTEX@makeanchor{part*}%
%     \addcontentsline{toc}{part}{#1}%
    \fi
  \else
    \CTEX@ifnamefalse
    \CTEX@makeanchor{part*}%
%   \addcontentsline{toc}{part}{#1}%
  \fi
  \CTEX@gettitle{#1}%
  \CTEX@addtocline{part}{#1}%
  \partmark{#1}%
  \begingroup
%   \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
    \CTEX@heading@format@initial
    \CTEX@part@format{%
%   \ifnum \c@secnumdepth >\m@ne
%     \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
%   \fi
      \CTEX@headinghang{part}%
        {\CTEXifname{\CTEX@partname\CTEX@part@aftername}{}}%
%   \huge\bfseries #2%
        \CTEX@part@titleformat{#2}%
%   \markboth{}{}\par
        \CTEX@part@aftertitle}\par
  \endgroup
  \nobreak
% \vskip 3ex
  \CTEX@setheadingskip \CTEX@part@afterskip
  \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
  \vskip \CTEX@headingskip
  \@afterheading}
⟨/article⟩
⟨*book|report⟩
\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >-2\relax
    \ifodd \CTEX@part@numbering
      \CTEX@ifnametrue
      \refstepcounter{part}%
%     \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \CTEX@ifnamefalse
      \CTEX@makeanchor{part*}%
%     \addcontentsline{toc}{part}{#1}%
    \fi
  \else
    \CTEX@ifnamefalse
    \CTEX@makeanchor{part*}%
%   \addcontentsline{toc}{part}{#1}%
  \fi
  \CTEX@gettitle{#1}%
  \CTEX@addtocline{part}{#1}%
%  \markboth{}{}%
  \partmark{#1}%
  \begingroup
%   \centering \interlinepenalty \@M \normalfont
    \CTEX@heading@format@initial
    \CTEX@part@format{%
%   \ifnum \c@secnumdepth >-2\relax
%     \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
%   \fi
      \CTEX@headinghang{part}%
        {\CTEXifname{\CTEX@partname\CTEX@part@aftername}{}}%
%   \Huge\bfseries #2\par
      \CTEX@part@titleformat{#2}%
      \CTEX@part@aftertitle}\par
```

```
    \endgroup
    \@endpart}
⟨/book|report⟩
```

\@spart
```
⟨*article⟩
\def\@spart#1{%
    \CTEX@ifnamefalse
    \CTEX@makeanchor@spart{part*}%
    \CTEX@gettitle{#1}%
    \begingroup
%     \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
      \CTEX@heading@format@initial
      \CTEX@part@format{%
        \CTEX@headinghang{part}{}%
%     \huge \bfseries #1\par
        \CTEX@part@titleformat{#1}%
        \CTEX@part@aftertitle}\par
    \endgroup
    \nobreak
% \vskip 3ex
    \CTEX@setheadingskip \CTEX@part@afterskip
    \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
    \vskip \CTEX@headingskip
    \@afterheading}
⟨/article⟩
⟨*book|report⟩
\def\@spart#1{%
    \CTEX@ifnamefalse
    \CTEX@makeanchor@spart{part*}%
    \CTEX@gettitle{#1}%
    \begingroup
%     \centering \interlinepenalty \@M \normalfont
      \CTEX@heading@format@initial
      \CTEX@part@format{%
        \CTEX@headinghang{part}{}%
%     \Huge \bfseries #1\par%
        \CTEX@part@titleformat{#1}%
        \CTEX@part@aftertitle}\par
    \endgroup
    \@endpart}
⟨/book|report⟩
```

\@endpart
```
⟨*book|report⟩
\def\@endpart{%
% \vfil
    \CTEX@setheadingskip \CTEX@part@afterskip
    \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
    \vskip \CTEX@headingskip
    \newpage
    \if@twoside
      \if@openright
        \null
        \thispagestyle{empty}%
        \newpage
      \fi
    \fi
    \if@tempswa
      \twocolumn
    \fi}
⟨/book|report⟩
```

15.8.2.2　chapter 的标题

```
⟨*book|report⟩
```

\chapter
```
\renewcommand\chapter{%
% \if@openright\cleardoublepage\else\clearpage\fi
```

```
% \thispagestyle{plain}%
  \CTEX@chapter@break
  \CTEX@setthispagestyle{chapter}%
  \global\@topnum\z@
% \@afterindentfalse
  \ifodd \CTEX@chapter@afterindent
    \@afterindenttrue
  \else
    \@afterindentfalse
  \fi
  \secdef\@chapter\@schapter}

\@chapter \def\@chapter[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
⟨*book⟩
    \if@mainmatter
⟨/book⟩
      \ifodd \CTEX@chapter@numbering
        \CTEX@ifnametrue
        \refstepcounter{chapter}%
%       \typeout{\@chapapp\space\thechapter.}%
        \typeout{\CTEXthechapter}%
%       \addcontentsline{toc}{chapter}
%         {\protect\numberline{\thechapter}#1}%
      \else
        \CTEX@ifnamefalse
        \CTEX@makeanchor{\Hy@chapapp*}%
%       \addcontentsline{toc}{chapter}{#1}%
      \fi
⟨*book⟩
    \else
      \CTEX@ifnamefalse
      \CTEX@makeanchor@chapter{\Hy@chapapp*}%
%     \addcontentsline{toc}{chapter}{#1}%
    \fi
⟨/book⟩
  \else
    \CTEX@ifnamefalse
    \CTEX@makeanchor@chapter{\Hy@chapapp*}%
%   \addcontentsline{toc}{chapter}{#1}%
  \fi
  \CTEX@gettitle{#1}%
  \CTEX@addtocline{chapter}{#1}%
  \chaptermark{#1}%
% \addtocontents{lof}{\protect\addvspace{10\p@}}%
% \addtocontents{lot}{\protect\addvspace{10\p@}}%
  \CTEX@addloflotskip{chapter}%
  \if@twocolumn
    \@topnewpage[\@makechapterhead{#2}]%
  \else
    \@makechapterhead{#2}%
  \@afterheading
  \fi}

\@schapter \def\@schapter#1{%
  \CTEX@ifnamefalse
  \CTEX@makeanchor@schapter{\Hy@chapapp*}%
  \CTEX@gettitle{#1}%
  \if@twocolumn
    \@topnewpage[\@makeschapterhead{#1}]%
  \else
    \@makeschapterhead{#1}%
    \@afterheading
  \fi}

\@makechapterhead \def\@makechapterhead#1{%
% \vspace*{50\p@}%
```

```
  \CTEX@setheadingskip \CTEX@chapter@beforeskip
  \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
  \vspace*{\CTEX@headingskip}%
  \begingroup
%   \parindent \z@ \raggedright \normalfont
    \CTEX@heading@format@initial
    \CTEX@chapter@format{%
%   \ifnum \c@secnumdepth >\m@ne
%     \if@mainmatter
%       \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
%     \fi
%   \fi
      \CTEX@headinghang{chapter}%
        {\CTEXifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
%   \Huge \bfseries #1\par\nobreak
      \CTEX@chapter@titleformat{#1}%
      \CTEX@chapter@aftertitle}\par
  \endgroup
  \nobreak
%  \vskip 40\p@
  \CTEX@setheadingskip \CTEX@chapter@afterskip
  \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
  \vskip \CTEX@headingskip}

\@makeschapterhead
\def\@makeschapterhead#1{%
% \vspace*{50\p@}%
  \CTEX@setheadingskip \CTEX@chapter@beforeskip
  \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
  \vspace*{\CTEX@headingskip}%
  \begingroup
% \parindent \z@ \raggedright \normalfont \interlinepenalty\@M
    \CTEX@heading@format@initial
    \CTEX@chapter@format{%
      \CTEX@headinghang{chapter}{}%
%     \Huge \bfseries  #1\par\nobreak
      \CTEX@chapter@titleformat{#1}%
      \CTEX@chapter@aftertitle}\par
  \endgroup
  \nobreak
% \vskip 40\p@
  \CTEX@setheadingskip \CTEX@chapter@afterskip
  \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
  \vskip \CTEX@headingskip}

⟨/book|report⟩
```

15.8.2.3　section 类的标题

\@startsection　LaTeX 的标准参数是：

{⟨*name*⟩}{⟨*level*⟩}{⟨*indent*⟩}{⟨*beforeskip*⟩}{⟨*afterskip*⟩}{⟨*style*⟩}*[⟨*altheading*⟩]{⟨*heading*⟩}

```
\def\@startsection#1#2#3#4#5#6{%
  \if@noskipsec \leavevmode \fi
  \par
% \@tempskipa #4\relax
% \@afterindenttrue
% \ifdim \@tempskipa <\z@
%   \@tempskipa -\@tempskipa \@afterindentfalse
% \fi
  \CTEX@update@sectionformat@n{#1}%
  \ifodd \CTEX@afterindent
    \@afterindenttrue
  \else
    \@afterindentfalse
  \fi
  \if@nobreak
```

```
3127      \everypar{}%
3128    \else
3129 %    \addpenalty\@secpenalty\addvspace\@tempskipa
3130      \csname CTEX@#1@break\endcsname
3131      \CTEX@setheadingskip{#4}%
3132      \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
3133      \addvspace \CTEX@headingskip
3134    \fi
3135    \@ifstar
3136      {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
3137      {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

\@seccntformat 3138 \def\@seccntformat#1{%
3139 % \csname the#1\endcsname\quad}%
3140    \csname CTEX@#1name\endcsname
3141    \csname CTEX@#1@aftername\endcsname}

      \@sect 3142 \def\@sect#1#2#3#4#5#6[#7]#8{%
3143    \ifnum #2>\c@secnumdepth
3144      \CTEX@ifnamefalse
3145      \CTEX@makeanchor@sect{#1*}%
3146      \let\@svsec\@empty
3147    \else
3148      \ifodd \csname CTEX@#1@numbering\endcsname
3149        \CTEX@ifnametrue
3150        \refstepcounter{#1}%
3151        \protected@edef\@svsec{\@seccntformat{#1}\relax}%
3152      \else
3153        \CTEX@ifnamefalse
3154        \CTEX@makeanchor{#1*}%
3155        \let\@svsec\@empty
3156      \fi
3157    \fi
3158    \CTEX@gettitle{#7}%
3159 % \@tempskipa #5\relax
3160 % \ifdim \@tempskipa>\z@
3161    \unless \ifodd \CTEX@runin
3162      \begingroup
3163        \CTEX@heading@format@initial
3164        #6{%
3165 %        \@hangfrom{\hskip #3\relax\@svsec}%
3166 %        \interlinepenalty \@M #8\@@par
3167          \CTEX@sectionhang{#3}{\@svsec}%
3168          \csname CTEX@#1@titleformat\endcsname{#8}%
3169          \csname CTEX@#1@aftertitle\endcsname}\par
3170      \endgroup
3171      \csname #1mark\endcsname{#7}%
3172 %    \addcontentsline{toc}{#1}{%
3173 %      \ifnum #2>\c@secnumdepth \else
3174 %        \protect\numberline{\csname the#1\endcsname}%
3175 %      \fi
3176 %      #7}%
3177      \CTEX@addtocline{#1}{#7}%
3178    \else
3179      \def\@svsechd{%
3180        #6{%
3181 %        \hskip #3\relax \@svsec #8
3182          {\CTEX@indentbox{#3}}\@svsec
3183          \csname CTEX@#1@titleformat\endcsname{#8}%
3184          \csname CTEX@#1@aftertitle\endcsname}%
3185        \csname #1mark\endcsname{#7}%
3186 %      \addcontentsline{toc}{#1}{%
3187 %        \ifnum #2>\c@secnumdepth \else
3188 %          \protect\numberline{\csname the#1\endcsname}%
3189 %        \fi
3190 %        #7}%
3191        \CTEX@addtocline{#1}{#7}}%
```

```
3192   \fi
3193   \@xsect{#5}}

\@ssect 3194 \def\@ssect#1#2#3#4#5{%
3195   \CTEX@ifnamefalse
3196   \CTEX@gettitle{#5}%
3197 % \@tempskipa #3\relax
3198 % \ifdim \@tempskipa>\z@
3199   \unless \ifodd \CTEX@runin
3200     \begingroup
3201       \CTEX@heading@format@initial
3202       #4{%
3203 %       \@hangfrom{\hskip #1}%
3204 %       \interlinepenalty \@M #5\__ctexpar
3205         \CTEX@sectionhang{#1}{}%
3206           \CTEX@titleformat@n{#5}%
3207           \CTEX@aftertitle}\par
3208     \endgroup
3209   \else
3210 %   \def\@svsechd{#4{\hskip #1\relax #5}}%
3211     \def\@svsechd{#4{{\CTEX@indentbox{#1}}%
3212       \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
3213   \fi
3214   \@xsect{#3}}

\@xsect 3215 \def\@xsect#1{%
3216 % \@tempskipa #1\relax
3217 % \ifdim \@tempskipa>\z@
3218   \unless \ifodd \CTEX@runin
3219     \par \nobreak
3220 %   \vskip \@tempskipa
3221     \CTEX@setheadingskip{#1}%
3222     \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
3223     \vskip \CTEX@headingskip
3224     \@afterheading
3225   \else
3226     \@nobreakfalse
3227     \global\@noskipsectrue
3228     \everypar{%
3229       \if@noskipsec
3230         \global\@noskipsecfalse
3231        {\setbox\z@\lastbox}%
3232         \clubpenalty\@M
3233         \begingroup \@svsechd \endgroup
3234         \unskip
3235 %       \@tempskipa #1\relax
3236 %       \hskip -\@tempskipa
3237         \CTEX@heading@glue{#1}%
3238       \else
3239         \clubpenalty \@clubpenalty
3240         \everypar{}%
3241       \fi}%
3242   \fi
3243   \ignorespaces}
```

\CTEX@headinghang
\CTEX@sectionhang

分别用于用于实现 \part/\chapter 和 \section 类标题的 indent 和 hang 选项。

```
3244 \cs_new_protected:Npn \CTEX@headinghang #1
3245   {
3246     \ctex_heading_hang:cnn
3247       { CTEX@#1@hang }
3248       { \use:c { CTEX@#1@indent } }
3249   }
3250 \cs_new_protected:Npn \CTEX@sectionhang
3251   { \ctex_heading_hang:Nnn \CTEX@hang }
```

\ctex_heading_hang:Nnn
\ctex_hang_from:n

hang 选项控制是否采用悬挂缩进, 同时设置 \parindent。

```
\cs_new_protected:Npn \ctex_heading_hang:Nnn #1#2#3
  {
    \dim_set:Nn \tex_parindent:D {#2}
    \bool_if:NTF #1
      { \ctex_hang_from:n }
      { \use:n }
      { \__ctex_insert_indent: #3 }
  }
\cs_new_protected:Npn \ctex_hang_from:n #1
  {
    \tex_noindent:D
    \hbox_set:Nn \l__ctex_tmp_box {#1}
    \tex_hangindent:D = \box_wd:N \l__ctex_tmp_box
    \box_use_drop:N \l__ctex_tmp_box
  }
\cs_generate_variant:Nn \ctex_heading_hang:Nnn { c }
```

\ctex_heading_glue:n
\CTEX@heading@glue

如果缩进 #1 长度为零, 就不插入水平间距。

```
\cs_new_protected:Npn \ctex_heading_glue:n #1
  {
    \group_begin:
      \skip_set:Nn \l__ctex_heading_skip {#1}
      \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
        { \skip_horizontal:N \l__ctex_heading_skip }
    \group_end:
  }
\cs_new_eq:NN \CTEX@heading@glue \ctex_heading_glue:n
```

\CTEX@update@sectionformat@n

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

```
\cs_new_protected:Npn \CTEX@update@sectionformat@n #1
  {
    \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
    \cs_set_eq:Nc \CTEX@aftertitle    { CTEX@#1@aftertitle }
    \cs_set_eq:Nc \CTEX@afterindent   { CTEX@#1@afterindent }
    \cs_set_eq:Nc \CTEX@fixskip       { CTEX@#1@fixskip }
    \cs_set_eq:Nc \CTEX@hang          { CTEX@#1@hang }
    \cs_set_eq:Nc \CTEX@runin         { CTEX@#1@runin }
  }
\cs_new_eq:NN \CTEX@titleformat@n \use:n
\cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
\cs_new_eq:NN \CTEX@afterindent \c_true_bool
\cs_new_eq:NN \CTEX@fixskip \c_false_bool
\cs_new_eq:NN \CTEX@hang \c_true_bool
\cs_new_eq:NN \CTEX@runin \c_false_bool
```

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

```
\cs_new:Npn \CTEX@part@tocline #1#2
  {
    \CTEXifname
      { \CTEXthepart \hspace { 1em } }
      { }
    #2
  }
⟨*book|report⟩
\cs_new:Npn \CTEX@chapter@tocline #1#2
  {
    \CTEXifname
      { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
      { }
    #2
  }
⟨/book|report⟩
```

\CTEXnumberline

```
\cs_new:Npn \CTEXnumberline #1
```

```
  {
    \CTEXifname
      { \protect \numberline { \use:c { CTEXthe #1 } } }
      { }
  }

\int_zero:N \l__ctex_tmp_int
\seq_map_inline:Nn \c__ctex_section_headings_seq
  {
    \int_incr:N \l__ctex_tmp_int
    \cs_gset_protected:cpx  {#1}
      {
        \exp_not:N \@startsection {#1}
          { \int_use:N \l__ctex_tmp_int }
          { \exp_not:c { CTEX@#1@indent } }
          { \exp_not:c { CTEX@#1@beforeskip } }
          { \exp_not:c { CTEX@#1@afterskip } }
          { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
      }
    \cs_new:cpn { CTEX@#1@tocline } ##1##2
      { \CTEXnumberline { ##1 } ##2 }
  }
```

15.8.2.4　附录标题

appendix/name
appendix/number
appendix/numbering

```
\ctex_define:n
  {
    appendix                .meta:nn = { ctex / appendix } {#1} ,
    appendix / name          .code:n =
      { \ctex_assign_heading_name:nn { appendix } {#1} } ,
    appendix / number      .tl_set:N = \CTEX@appendix@number ,
    appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
    appendix / numbering  .initial:n = true
  }
\tl_new:N \CTEX@preappendix
\tl_new:N \CTEX@postappendix
```

\appendix

```
\cs_new_eq:NN \CTEX@save@appendix \appendix
\cs_gset_protected:Npn \appendix
  {
    \CTEX@save@appendix
⟨*article⟩
    \gdef \CTEX@presection { \CTEX@preappendix }
    \gdef \CTEX@thesection { \CTEX@appendix@number }
    \gdef \CTEX@postsection { \CTEX@postappendix }
    \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
⟨/article⟩
⟨*book|report⟩
    \gdef \CTEX@prechapter { \CTEX@preappendix }
    \gdef \CTEX@thechapter { \CTEX@appendix@number }
    \gdef \CTEX@postchapter { \CTEX@postappendix }
    \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
⟨/book|report⟩
  }
```

15.8.2.5　设置 hyperref 宏包的标题锚点

\CTEX@makeanchor 设置超链接跳转锚点, 在 hyperref 载入后才有意义。

```
\cs_new_protected:Npn \CTEX@makeanchor #1
  { }
```

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

```
\seq_const_from_clist:Nn \c__ctex_headings_cs_seq
```

```
⟨article⟩ 3361   { part , spart , sect , ssect }
⟨book|report⟩ 3362   { part , spart , chapter , schapter , sect , ssect }
3363 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3364   {
3365     \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3366     \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3367   }
```

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

```
3368 \cs_new_protected:Npn \CTEX@hyperheadinghook
3369   {
3370     \group_begin:
3371       \legacy_if:nTF { Hy@implicit }
3372         {
3373           \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3374           \seq_map_inline:Nn \c__ctex_headings_cs_seq
3375             {
3376               \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3377                 {
3378                   \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3379                   \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3380                 }
3381             }
3382         }
3383         {
3384           \seq_map_inline:Nn \c__ctex_headings_cs_seq
3385             { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3386         }
3387     \group_end:
3388   }

3389 \ctex_at_end_package:nn { hyperref }
3390   {
3391     \cs_gset_protected:Npn \CTEX@makeanchor #1
3392       {
3393         \Hy@MakeCurrentHrefAuto {#1}
3394         \Hy@raisedlink
3395           {
3396             \hyper@anchorstart { \@currentHref }
3397             \hyper@anchorend
3398           }
3399       }
3400     \CTEX@hyperheadinghook
3401   }
```

15.8.2.6　兼容 nameref 宏包

\CTEX@gettitle 在 nameref 载入后才有意义，与上述 hyperref 的处理类似。

```
3402 \cs_new_protected:Npn \CTEX@gettitle #1
3403   { }
3404 \ctex_at_end_package:nn { nameref }
3405   {
3406     \cs_gset_protected:Npn \CTEX@gettitle { \NR@gettitle }
3407     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3408       {
3409         \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
3410           { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
3411       }
3412   }
```

15.8.2.7　兼容 titlesec 宏包

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

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

\ctex_titlesec_hook: \titleformat 的设置保存在名为 \ttlf@⟨*section*⟩ 的宏中备用,它的内容是

\ttlh@⟨*shape*⟩{⟨*format*⟩}{⟨*label*⟩}{⟨*sep*⟩}{⟨*before*⟩}{⟨*after*⟩}

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

{⟨*left*⟩}{⟨*right*⟩}{⟨*before*⟩}{⟨*after*⟩}{⟨*afterindent*⟩}

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

```
\cs_new_protected:Npn \ctex_titlesec_hook:
  {
    \@ifpackagewith { titlesec } { explicit }
      {
        \cs_set_eq:NN \__ctex_titlesec_format:Nn
                      \__ctex_titlesec_format_explicit:Nn
      }
      { }
    \clist_map_inline:nn
      { indentafter , noindentafter , indentfirst , nonindentfirst }
      {
        \@ifpackagewith { titlesec } { ##1 }
          {
            \clist_map_break:n
              { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
          }
          { }
      }
    \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
  }
\cs_new_protected:Npn \__ctex_titlesec_hook:n #1
  {
    \__ctex_titlesec_format:n {#1}
    \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
  }
\cs_new_protected:Npn \__ctex_titlesec_format:n #1
  {
    \cs_if_free:cF { ttlf@#1 }
      { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
  }
\cs_new_protected:Npn \__ctex_titlesec_format:Nn #1#2
  {
    \tl_set:Ne #1
      {
        \bool_if:cTF { CTEX@#2@runin }
          { \exp_not:N \ttlh@runin }
          { \exp_not:N \ttlh@hang }
        \tl_tail:N #1
      }
  }
```

```
\cs_new_protected:Npn \__ctex_titlesec_format_explicit:Nn #1#2
  {
    \cs_set_nopar:Npx #1 ##1
      {
        \bool_if:cTF { CTEX@#2@runin }
          { \exp_not:N \ttlh@runin }
          { \exp_not:N \ttlh@hang }
        \exp_args:No \tl_tail:n { #1 { } }
      }
  }
\cs_new_protected:Npn \__ctex_titlesec_spacing:Nn #1#2
  { \tl_set:Ne #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
\cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
  {
    \exp_not:n { {#1} {#2} {#3} {#4} }
    { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
  }

\@ifpackageloaded { titlesec }
  { }
  {
    \ctex_at_end_package:nn { titlesec }
      {
        \@ifpackagewith { titlesec } { loadonly }
          { }
          { \ctex_titlesec_hook: }
      }
  }
```

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

```
\ctex_at_end_package:nn { titlesec }
  {
⟨*book|report⟩
    \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
⟨/book|report⟩
    \cs_if_free:NF \ttl@tocpart
      {
        \cs_set_protected:Npn \ttl@tocpart
          { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
      }
    \seq_map_inline:Nn \c__ctex_headings_seq
      {
        \cs_if_exist:cF { ttl@toc #1 }
          {
            \cs_new_protected:cpx { ttl@toc #1 }
              {
                \tl_set:Nn \exp_not:N \ttl@a
                  {
                    \exp_not:N \protect
                    \exp_not:N \numberline { \exp_not:c { CTEXthe #1 } }
                  }
              }
          }
      }
  }
```

在 titlesec 包定义的标题中更新 \CTEXifname。

```
\ctex_at_end_package:nn { titlesec }
  {
    \ctex_patch_cmd:Nnn \ttl@labelling
      { \let \ifttl@toclabel \ifttl@label }
      {
        \let \ifttl@toclabel \ifttl@label
        \CTEX@updatettlifname
      }
```

```
    \cs_new_protected:Npn \CTEX@updatettlifname
      { \legacy_if:nTF { ttl@label } { \CTEX@ifnametrue } { \CTEX@ifnamefalse } }
  }
```

15.8.2.8　兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题[19]。

\ctex_titleps_hook:

我们修改 titleps 包的内部命令 \ttl@settopmark 和 \ttl@setsubmark, 将 \CTEXtheXXX 等加入更新队列中。

```
\group_begin:
\char_set_catcode_other:N \#
\cs_new_protected:Npn \ctex_titleps_hook:
  {
    \ctex_patch_cmd:Nnn \ttl@settopmark
      { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
      {
        \protect \@namedef { the#1 } { \@nameuse { the#1 } }
        \CTEX@titlepslabel@set {#1}
      }
    \ctex_patch_cmd:Nnn \ttl@setsubmark
      { \protect \@namedef { the#1 } { } }
      {
        \protect \@namedef { the#1 } { }
        \CTEX@titlepslabel@clear {#1}
      }
    \ctex_patch_cmd:Nnn \ttl@setsubmark
      { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
      {
        \protect \@namedef { the#2 } { \@nameuse { the#2 } }
        \CTEX@titlepslabel@set {#2}
      }
  }
\group_end:
```

\CTEX@titlepslabel@set
\CTEX@titlepslabel@clear

这两个函数要在随后被 \xdef 展开来获得 \CTEXtheXXX 的内容, 不应该用 \protected 来定义。

```
\cs_new:Npn \CTEX@titlepslabel@set #1
  {
    \cs_if_free:cF { CTEXthe#1 }
      { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
  }
\cs_new:Npn \CTEX@titlepslabel@clear #1
  {
    \cs_if_free:cF { CTEXthe#1 }
      { \protect \@namedef { CTEXthe#1 } { } }
  }
```

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

```
\ctex_at_end_package:nn { titlesec }
  { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
\ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }
```

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

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

[19] https://github.com/CTeX-org/ctex-kit/issues/217

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

\ttl@setifthe 使 \iftheXXX 等命令在页眉设置中可用。

```
\ctex_at_end_package:nn { titleps }
  {
    \cs_set_protected:Npn \ttl@setifthe #1
      {
        \exp_args:Nco \cs_set:Npn { ifthe #1 }
          {
            \CTEXifname
              { \protect \@firstoftwo }
              { \protect \@secondoftwo }
          }
      }
    \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
  }
```

15.8.3　目录标签的宽度

\CTEX@toc@width@n

```
\cs_new_protected:Npn \CTEX@toc@width@n #1
  {
    \hbox_set:Nn \l__ctex_tmp_box {#1}
    \dim_set:Nn \@tempdima
      {
        \dim_max:nn { \@tempdima }
          { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
      }
  }
```

\numberline 为 \numberline 命令打补丁，并兼容 tocloft 和 titletoc 宏包。

__ctex_patch_toc_width:n

这里需要替换 # 本身，因此需要先切换为 other 类。表示参数的 # 用 \c_parameter_token 代替。

```
\group_begin:
\char_set_catcode_other:N \#
\use:n
  {
    \group_end:
    \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
      { \CTEX@toc@width@n {#1} }
      { }
      { \ctex_patch_failure:N \numberline }
    \cs_new_protected:Npn \__ctex_patch_toc_width:n \c_parameter_token 1
      {
        \@ifpackageloaded { \c_parameter_token 1 }
          { }
          {
            \ctex_at_end_package:nn { \c_parameter_token 1 }
              {
                \ctex_preto_cmd:NnnTF \numberline
                  { \char_set_catcode_letter:n { 64 } }
                  { \CTEX@toc@width@n {#1} }
                  { }
                  { \ctex_patch_failure:N \numberline }
              }
          }
      }
  }
\__ctex_patch_toc_width:n { tocloft  }
\__ctex_patch_toc_width:n { titletoc }
```

15.8.4　页眉信息的修改

\ps@headings
```
\langle*article\rangle
\legacy_if:nTF { @twoside }
  {
    \ctex_patch_cmd:Nnn \ps@headings
      { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
      { \CTEXifname { \CTEXthesection \quad } { } }
    \ctex_patch_cmd:Nnn \ps@headings
      { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
      { \CTEXifname { \CTEXthesubsection \quad } { } }
  }
```

不知为何，标准文档类此处对 secnumdepth 的判断为 0，与 \section 的层次 1 不符。

```
  {
    \ctex_patch_cmd:Nnn \ps@headings
      { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
      { \CTEXifname { \CTEXthesection \quad } { } }
  }
\langle/article\rangle
\langle*book|report\rangle
\ctex_patch_cmd:Nnn \ps@headings
  {
\langle book\rangle    \ifnum \c@secnumdepth > \m@ne \if@mainmatter
\langle report\rangle  \ifnum \c@secnumdepth > \m@ne
      \@chapapp \ \thechapter . ~ \ %
\langle report\rangle  \fi
\langle book\rangle    \fi \fi
  }
  { \CTEXifname { \CTEXthechapter \quad } { } }
\legacy_if:nT { @twoside }
  {
    \ctex_patch_cmd:Nnn \ps@headings
      { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
      { \CTEXifname { \CTEXthesection \quad } { } }
  }
\langle/book|report\rangle
```

\f@nch@initialise 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。

```
\ctex_at_end_package:nn { fancyhdr }
  {
    \ctex_patch_cmd:Nnn \f@nch@initialise
      { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
      { \CTEXifname { \CTEXthesection \quad } { } }
    \ctex_patch_cmd:Nnn \f@nch@initialise
      { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
      { \CTEXifname { \CTEXthesubsection \quad } { } }
    \ctex_patch_cmd:Nnn \f@nch@initialise
      { \ifnum \c@secnumdepth > \m@ne \@chapapp\ \thechapter . ~ \ \fi }
      { \CTEXifname { \CTEXthechapter \quad } { } }
    \ctex_patch_cmd:Nnn \f@nch@initialise
      { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
      { \CTEXifname { \CTEXthesection \quad } { } }
    \f@nch@initialise
```

fancyhdr 的 headings 选项会重定义 \ps@headings，这里也要打补丁。

```
    \@ifpackagewith { fancyhdr } { headings }
      {
\langle*article\rangle
        \legacy_if:nTF { @twoside }
          {
            \ctex_patch_cmd:Nnn \ps@headings
              { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
```

```
3657                { \CTEXifname { \CTEXthesection \quad } { } }
3658              \ctex_patch_cmd:Nnn \ps@headings
3659                { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3660                { \CTEXifname { \CTEXthesubsection \quad } { } }
3661            }
3662            {
3663              \ctex_patch_cmd:Nnn \ps@headings
3664                { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3665                { \CTEXifname { \CTEXthesection \quad } { } }
3666            }
3667 ⟨/article⟩
3668 ⟨*book|report⟩
3669          \ctex_patch_cmd:Nnn \ps@headings
3670            {
⟨book⟩ 3671              \ifnum \c@secnumdepth > \m@ne \if@mainmatter
⟨report⟩ 3672            \ifnum \c@secnumdepth > \m@ne
3673                \@chapapp \ \thechapter . ~ \ %
⟨report⟩ 3674            \fi
⟨book⟩ 3675              \fi \fi
3676            }
3677            { \CTEXifname { \CTEXthechapter \quad } { } }
3678          \legacy_if:nT { @twoside }
3679            {
3680              \ctex_patch_cmd:Nnn \ps@headings
3681                { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3682                { \CTEXifname { \CTEXthesection \quad } { } }
3683            }
3684 ⟨/book|report⟩
3685        }
3686        { }
3687    }

3688 ⟨/article|book|report⟩
```

15.8.5　beamer 标题页模板的修改

```
3689 ⟨*beamer⟩
3690 \ExplSyntaxOff
```

对应 \partpage。

```
3691 \defbeamertemplate*{part page}{CTEX}[1][]{%
3692   \begingroup
3693 %     \centering
3694 %     {\usebeamerfont{part name}%
3695 %      \usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}
3696 %     \vskip1em\par
3697     \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3698     \CTEX@heading@format@initial
3699     \CTEX@part@format{%
3700       \CTEX@indentbox{\CTEX@part@indent}%
3701       \ifodd \CTEX@part@numbering
3702         \CTEX@partname \CTEX@part@aftername
3703       \fi
3704       \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3705 %       \usebeamerfont{part title}\insertpart\par
3706         \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3707       \end{beamercolorbox}}%
3708     \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3709   \endgroup}
```

对应 \sectionpage。

```
3710 \defbeamertemplate*{section page}{CTEX}[1][]{%
3711   \begingroup
3712 %     \centering
3713 %     {\usebeamerfont{section name}%
```

```
%     \usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}
%    \vskip1em\par
    \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
    \CTEX@heading@format@initial
    \CTEX@section@format{%
      \CTEX@indentbox{\CTEX@section@indent}%
      \ifodd \CTEX@section@numbering
        \CTEX@sectionname \CTEX@section@aftername
      \fi
      \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
%       \usebeamerfont{section title}\insertsection\par
        \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
      \end{beamercolorbox}}%
    \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
  \endgroup}
```

对应 \subsectionpage。

```
\defbeamertemplate*{subsection page}{CTEX}[1][]{%
  \begingroup
%    \centering
%    {\usebeamerfont{subsection name}%
%     \usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
%    \vskip1em\par
    \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
    \CTEX@heading@format@initial
    \CTEX@subsection@format{%
      \CTEX@indentbox{\CTEX@subsection@indent}%
      \ifodd \CTEX@subsection@numbering
        \CTEX@subsectionname \CTEX@subsection@aftername
      \fi
      \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
%       \usebeamerfont{subsection title}\insertsubsection\par
        \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
      \end{beamercolorbox}}%
    \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
  \endgroup}
```

将 beamer 的默认模板重定向为 CTEX 模板。

```
\defbeamertemplatealias{part page}{default}{CTEX}
\defbeamertemplatealias{section page}{default}{CTEX}
\defbeamertemplatealias{subsection page}{default}{CTEX}

\ExplSyntaxOn

⟨/beamer⟩
```

15.8.6　标题编号和目录的层次设置

secnumdepth
tocdepth

secnumdepth 在 beamer 下无意义。

```
\ctex_define:n
  {
⟨*!beamer⟩
    secnumdepth           .code:n = \ctex_heading_depth:ne { secnumdepth } {#1} ,
    secnumdepth .value_required:n = true ,
⟨/!beamer⟩
    tocdepth              .code:n = \ctex_heading_depth:ne { tocdepth } {#1} ,
    tocdepth    .value_required:n = true
  }
```

\ctex_heading_depth:nn

注意此处 \setcounter 的赋值是全局的。

```
\cs_new_protected:Npn \ctex_heading_depth:nn #1#2
  {
    \prop_get:NnNTF \c__ctex_heading_level_prop {#2} \l__ctex_tmp_tl
      { \setcounter {#1} { \l__ctex_tmp_tl } }
```

```
      { \setcounter {#1} { \int_eval:n {#2} } }
  }
\cs_generate_variant:Nn \ctex_heading_depth:nn { ne }
```

\c__ctex_heading_level_prop 章节层次与名称的对应表。

```
\prop_const_from_keyval:Nn \c__ctex_heading_level_prop
  {
⟨*article|beamer⟩
    part          =  0 ,
⟨/article|beamer⟩
⟨*book|report⟩
    part          = -1 ,
    chapter       =  0 ,
⟨/book|report⟩
    section       =  1 ,
    subsection    =  2 ,
    subsubsection =  3 ,
    paragraph     =  4 ,
    subparagraph  =  5
  }
```

15.8.7　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。

```
\cs_new_protected:Npn \CTEX@setcurrentlabel@n #1
  {
    \protected@edef \@currentlabel
      {
        \cs_if_exist:cTF { CTEX@the#1 }
          { \exp_args:cc { p@#1 } { CTEX@the#1 } }
          { \exp_not:o { \@currentlabel } }
      }
  }
```

\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。较新版 LATEX 内核已经包含 \labelformat, 可以直接使用。

```
\cs_new_protected:Npn \ctex_varioref_hook:
  {
    \seq_map_inline:Nn \c__ctex_headings_seq
      { \ctex_fix_varioref_label:n { ##1 } }
    \ctex_at_end_package:nn { cleveref } { \ctex_cleveref_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。

```
\cs_new_protected:Npn \ctex_fix_varioref_label:n #1
  {
    \tl_if_empty:cT { p@#1 }
      { \exp_args:Nne \labelformat {#1} { \exp_not:c { CTEX@the#1 } } }
  }
```

patch/cleveref

cleveref 宏包长期未维护，与较新的 LaTeX 内核和 hyperref 存在兼容问题。CTEX 提供了对 cleveref 的补丁，但用户也可以选择关闭它。

当同时使用 hyperref 和 cleveref 时，\appendix 之后的交叉引用类型可能不正确（例如，appendix 退化为 chapter）。这是上游 LaTeX 内核 firstaid 对 cleveref 补丁不完整所致，与 CTEX 无关。可以使用以下方法规避：

```
\AddToHook{cmd/appendix/before}{%
  \crefalias{chapter}{appendix}%
  \crefalias{section}{subappendix}%
  \crefalias{subsection}{subsubappendix}%
  \crefalias{subsubsection}{subsubsubappendix}%
}
```

```
\ctex_define:n
  {
    patch / cleveref .bool_set:N = \l__ctex_patch_cleveref_bool ,
    patch / cleveref .initial:n = true
  }
```

\ctex_cleveref_hook:
__ctex_cleveref_hook_aux:N

需要将 cleveref 包对应命令中 \p@⟨*counter*⟩ 的参数及时展开，以兼容 \labelformat。

```
\cs_new_protected:Npn \ctex_cleveref_hook:
  {
    \bool_if:NT \l__ctex_patch_cleveref_bool
      {
        \@ifpackageloaded { hyperref }
          {
            \@ifpackagewith { hyperref } { implicit = false }
              { }
              { \__ctex_cleveref_hook_aux:N \H@refstepcounter }
          }
          {
            \__ctex_cleveref_hook_aux:N \refstepcounter@noarg
            \__ctex_cleveref_hook_aux:N \refstepcounter@optarg
          }
        \__ctex_cleveref_hook_aux:N \appendix
      }
  }
\cs_new_protected:Npn \__ctex_cleveref_hook_aux:N #1
  {
    \ctex_patch_cmd_all:NnnnTF #1
      {
        \ExplSyntaxOff
        \char_set_catcode_letter:n { 64 }
      }
      { \endcsname \csname the }
      { \expandafter \endcsname \csname the }
      { }
      { \ctex_patch_failure:N #1 }
  }
```

如果 varioref 已经被载入，则使用它来设置。

```
\cs_if_exist:NTF \labelformat
  { \ctex_varioref_hook: }
  {
    \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
    \RenewDocumentCommand \refstepcounter { m }
      {
        \CTEX@save@refstepcounter {#1}
        \CTEX@setcurrentlabel@n {#1}
      }
    \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
  }
```

15.8.8　载入 ⟨*scheme*⟩ 文件

```
\ctex_scheme_input:o { \l__ctex_scheme_tl }
⟨/class|heading⟩
```

15.8.9　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。

```
⟨*scheme&(article|book|report|beamer)⟩

\ctex_set:nn { part }
  {
    aftertitle  = \par ,
⟨*article|book|report⟩
    hang        = false ,
⟨/article|book|report⟩
⟨*plain⟩
    name        = \partname \space ,
⟨*article|book|report⟩
    number      = \thepart ,
⟨/article|book|report⟩
⟨*beamer⟩
    number      = \insertromanpartnumber ,
⟨/beamer⟩
⟨/plain⟩
⟨*chinese⟩
    number      = \chinese { part } ,
⟨/chinese⟩
⟨*article⟩
    beforeskip  = 4ex ,
    afterskip   = 3ex ,
⟨*plain⟩
    format      = \raggedright ,
    nameformat  = \Large \bfseries ,
    aftername   = \par \nobreak ,
    titleformat = \huge \bfseries ,
    afterindent = false
⟨/plain⟩
⟨*chinese⟩
    format      = \Large \bfseries \centering ,
    aftername   = \quad ,
    afterindent = true
⟨/chinese⟩
⟨/article⟩
⟨*book|report⟩
    aftername   = \par \vskip 20 \p@ ,
    beforeskip  = 0pt \@plus 1fil ,
    afterskip   = 0pt \@plus 1fil ,
    pagestyle   = plain ,
    break       = \if@openright \cleardoublepage \else \clearpage \fi ,
⟨*plain⟩
    format      = \centering ,
    nameformat  = \huge \bfseries ,
    titleformat = \Huge \bfseries
⟨/plain⟩
⟨*chinese⟩
    format      = \huge \bfseries \centering
⟨/chinese⟩
⟨/book|report⟩
⟨*beamer⟩
    format      = \centering ,
    nameformat  = \usebeamerfont { part ~ name }
                  \usebeamercolor [fg] { part ~ name } ,
    aftername   = \vskip 1em \par ,
```

```
    titleformat = \usebeamerfont { part ~ title }
⟨/beamer⟩
  }

⟨*book|report⟩
\ctex_set:nn { chapter }
  {
    pagestyle   = plain ,
    aftertitle  = \par ,
    hang        = false ,
    beforeskip  = 50 \p@ ,
    afterskip   = 40 \p@ ,
    lofskip     = 10 \p@ ,
    lotskip     = 10 \p@ ,
    break       = \if@openright \cleardoublepage \else \clearpage \fi ,
⟨*plain⟩
    name        = \chaptername \space ,
    number      = \thechapter ,
    format      = \raggedright ,
    nameformat  = \huge \bfseries ,
    aftername   = \par \nobreak \vskip 20 \p@ ,
    titleformat = \Huge \bfseries ,
    afterindent = false ,
    tocline     = \CTEXnumberline {#1} #2
⟨/plain⟩
⟨*chinese⟩
    number      = \chinese { chapter } ,
    format      = \huge \bfseries \centering ,
    aftername   = \quad ,
    afterindent = true
⟨/chinese⟩
  }
⟨/book|report⟩

\ctex_set:nn { section }
  {
⟨*article|book|report⟩
    number      = \thesection ,
    aftername   = \quad ,
    aftertitle  = \@@par ,
    beforeskip  = 3.5ex \@plus 1ex \@minus .2ex ,
    afterskip   = 2.3ex \@plus .2ex ,
    runin       = false ,
    break       = \addpenalty \@secpenalty ,
⟨*plain⟩
    format      = \Large \bfseries ,
    afterindent = false
⟨/plain⟩
⟨*chinese⟩
    format      = \Large \bfseries \centering ,
    afterindent = true
⟨/chinese⟩
⟨/article|book|report⟩
⟨*beamer⟩
⟨*plain⟩
    name        = \sectionname \space ,
⟨/plain⟩
    format      = \centering ,
    number      = \insertsectionnumber ,
    nameformat  = \usebeamerfont { section ~ name }
                  \usebeamercolor [fg] { section ~ name } ,
    aftername   = \vskip 1em \par ,
    titleformat = \usebeamerfont { section ~ title } ,
    aftertitle  = \par
⟨/beamer⟩
  }

\ctex_set:nn { subsection }
```

```
  {
⟨*article|book|report⟩
    number      = \thesubsection ,
    format      = \large \bfseries ,
    aftername   = \quad ,
    aftertitle  = \@@par ,
    beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
    afterskip   = 1.5ex  \@plus .2ex ,
    runin       = false ,
    break       = \addpenalty \@secpenalty ,
⟨*plain⟩
    afterindent = false
⟨/plain⟩
⟨*chinese⟩
    afterindent = true
⟨/chinese⟩
⟨/article|book|report⟩
⟨*beamer⟩
⟨*plain⟩
    name        = \subsectionname \space ,
    number      = \insertsubsectionnumber ,
⟨/plain⟩
⟨*chinese⟩
    number      = \arabic { section } . \arabic { subsection } ,
⟨/chinese⟩
    format      = \centering ,
    nameformat  = \usebeamerfont { subsection ~ name }
                  \usebeamercolor [fg] { subsection ~ name } ,
    aftername   = \vskip 1em \par ,
    titleformat = \usebeamerfont { subsection ~ title } ,
    aftertitle  = \par
⟨/beamer⟩
  }

⟨*article|book|report⟩

\ctex_set:nn { subsubsection }
  {
    number      = \thesubsubsection ,
    format      = \normalsize \bfseries ,
    aftername   = \quad ,
    aftertitle  = \@@par ,
    beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
    afterskip   = 1.5ex \@plus .2ex ,
    runin       = false ,
    break       = \addpenalty \@secpenalty ,
⟨*plain⟩
    afterindent = false
⟨/plain⟩
⟨*chinese⟩
    afterindent = true
⟨/chinese⟩
  }

\ctex_set:nn { paragraph }
  {
    number      = \theparagraph ,
    format      = \normalsize \bfseries ,
    aftername   = \quad ,
    beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
    break       = \addpenalty \@secpenalty ,
⟨*plain⟩
    afterindent = false
⟨/plain⟩
⟨*chinese⟩
    afterindent = true
⟨/chinese⟩
  }
```

```
\ctex_set:nn { subparagraph }
  {
    number      = \thesubparagraph ,
    format      = \normalsize \bfseries ,
    aftername   = \quad ,
    beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
    break       = \addpenalty \@secpenalty ,
⟨*plain⟩
    afterindent = false
⟨/plain⟩
⟨*chinese⟩
    afterindent = true
⟨/chinese⟩
  }
```

处理 sub3section 与 sub4section 的格式。

```
\int_compare:nNnTF \g__ctex_section_depth_int > 2
  {
    \ctex_set:nn { paragraph }
      {
        aftertitle  = \@@par ,
        afterskip   = 1ex \@plus .2ex ,
        runin       = false
      }
  }
  {
    \ctex_set:nn { paragraph }
      {
        afterskip   = 1em ,
        runin       = true
      }
  }
\int_compare:nNnTF \g__ctex_section_depth_int > 3
  {
    \ctex_set:nn { subparagraph }
      {
        aftertitle  = \@@par ,
        afterskip   = 1ex \@plus .2ex ,
        runin       = false
      }
  }
  {
    \ctex_set:nn { subparagraph }
      {
        afterskip   = 1em ,
        runin       = true
      }
  }
\int_compare:nNnTF \g__ctex_section_depth_int > 2
  { \ctex_set:nn { subparagraph } { indent = \c_zero_dim } }
  { \ctex_set:nn { subparagraph } { indent = \parindent } }
```

处理附录的格式。

```
\ctex_set:nn { appendix }
⟨*article⟩
  { number      = \@Alph \c@section }
⟨/article⟩
⟨*book|report⟩
  {
    name        = \appendixname \space ,
    number      = \@Alph \c@chapter
  }
⟨/book|report⟩

⟨/article|book|report⟩
```

```
⟨/scheme&(article|book|report|beamer)⟩
```

15.8.10　ctex.sty 的 heading 选项

```
⟨*ctex|ctexheading⟩
```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```
\clist_map_inline:nn { article , book , report , beamer }
  {
    \@ifclassloaded {#1}
      { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
      { }
  }
```

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。

```
\msg_new:nnn { ctex } { not-standard-class }
  {
    None~of~the~standard~document~classes~was~loaded.\\
    Heading~`#1'~is~selected.\\
    ctex~may~not~work~as~expected.
  }
⟨ctex⟩ \bool_if:NTF \l__ctex_heading_bool
⟨ctexheading⟩ \use:n
  {
    \tl_if_exist:NTF \c__ctex_std_class_tl
      { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
      {
        \cs_if_exist:NTF \chapter
          {
            \cs_if_exist:NF \if@mainmatter
              { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
            \tl_const:Nn \c__ctex_class_tl { book }
          }
          { \tl_const:Nn \c__ctex_class_tl { article } }
        \msg_warning:nne { ctex } { not-standard-class } { \c__ctex_class_tl }
      }
    \ctex_file_input:n { ctex-heading- \c__ctex_class_tl .def }
  }
⟨ctex⟩ { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
```

```
⟨/ctex|ctexheading⟩
```

15.8.11　标题配置文件

```
⟨*name⟩
```

```
\keys_set_known:nn { ctex }
  {
    contentsname   = 目录 ,
    listfigurename = 插图 ,
    listtablename  = 表格 ,
    figurename     = 图 ,
    tablename      = 表 ,
    abstractname   = 摘要 ,
    indexname      = 索引 ,
    bibname        = 参考文献 ,
    appendixname   = 附录 ,
    proofname      = 证明 ,
    algorithmname  = 算法 ,
    refname        = 参考文献 ,
    continuation   = （续） ,
    part    / name = { 第 , 部分 } ,
```

```
    chapter / name = { 第 , 章 }
  }
⟨/name⟩
```

15.9　chinese 方案的其他设置

```
⟨*scheme&chinese⟩
```

chinese 在标准文档类下的页面格式总采用 headings。

```
⟨article|book|report⟩ \pagestyle { headings }
```

日期格式。

```
\ctex_set:n { today = small }
```

若用户未设置宏包选项 autoindent，则自动调整首行缩进。

```
\ctex_if_autoindent_touched:F
  { \ctex_set:n { autoindent = true } }
```

使用标题定义时的设置。首先是命题名字汉化。beamer 需要汉化定理名称。

```
⟨*!generic⟩
\str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
⟨*beamer⟩
  {
    \uselanguage { ChineseGBK }
    \languagealias { chinese } { ChineseGBK }
    \ctex_file_input:n { ctex-name-gbk.cfg }
  }
  {
    \uselanguage { ChineseUTF8 }
    \languagealias { chinese } { ChineseUTF8 }
    \ctex_file_input:n { ctex-name-utf8.cfg }
  }
```

让 translator 包优先查找中文翻译。

```
\clist_put_left:Nn \trans@languagepath { chinese }
⟨/beamer⟩
⟨*!beamer⟩
  { \ctex_file_input:n { ctex-name-gbk.cfg } }
  { \ctex_file_input:n { ctex-name-utf8.cfg } }
```

对 beamer 以外的文档类，若用户未设置宏包选项 zihao，则设置 \normalsize 为五号字。beamer 不调整默认字体大小。

```
\int_compare:nNnF \g__ctex_font_size_int > { -1 }
  { \int_gset:Nn \g__ctex_font_size_int { 0 } }
```

对 beamer 以外的文档类，若用户未设置宏包选项 linespread，则设置行距初始值为 $1.3 \times 1.2 = 1.56$ 倍字体大小。beamer 不调整行距。

```
\fp_if_nan:nT { \l__ctex_line_spread_fp }
  { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
⟨/!beamer⟩
⟨/!generic⟩
```

不使用标题定义时的通用设置，注意此处 \c__ctex_std_class_tl 可能没有定义。

```
⟨*generic⟩
\tl_if_eq:NnTF \c__ctex_std_class_tl { beamer }
  {
    \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
      {
        \uselanguage { ChineseGBK }
        \languagealias { chinese } { ChineseGBK }
        \ctex_file_input:n { ctex-name-gbk.cfg }
```

```
      }
      {
        \uselanguage { ChineseUTF8 }
        \languagealias { chinese } { ChineseUTF8 }
        \ctex_file_input:n { ctex-name-utf8.cfg }
      }
    \clist_put_left:Nn \trans@languagepath { chinese }
  }
  {
    \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
      { \ctex_file_input:n { ctex-name-gbk.cfg } }
      { \ctex_file_input:n { ctex-name-utf8.cfg } }
    \int_compare:nNnF \g__ctex_font_size_int > { -1 }
      { \int_gset:Nn \g__ctex_font_size_int { 0 } }
    \fp_if_nan:nT { \l__ctex_line_spread_fp }
      { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
```

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

```
    \tl_if_exist:NT \c__ctex_std_class_tl
      { \RequirePackage { indentfirst } }
  }
```

⟨/generic⟩

⟨/scheme&chinese⟩

15.10　中文字号

⟨*class|ctex|ctexsize⟩

\zihao

```
\NewDocumentCommand \zihao { m }
  { \exp_args:Ne \ctex_zihao:n {#1} \tex_ignorespaces:D }
```

\ctex_zihao:n

```
\cs_new_protected:Npn \ctex_zihao:n #1
  {
    \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
      { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
      { \msg_error:nnn { ctex } { fontsize } {#1} }
  }
\msg_new:nnnn { ctex } { fontsize }
  { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
  {
    The~old~font~size~is~used~if~you~continue.\\
    The~available~font~sizes~are~listed~as~follow.\\
    \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
  }
```

15.10.1　定义中文字号

\c__ctex_font_size_prop
__ctex_save_font_size:nn
\ctex_save_font_size:nn

基础行距是字号的 1.2 倍, 采用 ε-TEX 的 scaling 运算得到的结果要比简单的 `1.2\dimexpr` 精确[20]。

```
\prop_new:N \l__ctex_font_size_temp_prop
\seq_new:N  \l__ctex_font_size_temp_seq
\cs_new_protected:Npn \__ctex_save_font_size:nn #1#2
  {
    \prop_put:Nne \l__ctex_font_size_temp_prop {#1}
      {
        { \dim_to_decimal:n {#2} }
        { \dim_to_decimal:n { (#2) * 6 / 5 } }
      }
    \seq_put_right:Nn \l__ctex_font_size_temp_seq {#1}
  }
\cs_new_eq:NN \ctex_save_font_size:nn \__ctex_save_font_size:nn
\group_begin:
```

```
\prop_clear:N \l__ctex_font_size_temp_prop
\seq_clear:N  \l__ctex_font_size_temp_seq
\str_case:onF { \g__ctex_font_size_system_tl }
  {
    { word }
    {
      \clist_map_inline:nn
        {
          {  8 } { 5    bp } ,
          {  7 } { 5.5  bp } ,
          { -6 } { 6.5  bp } ,
          {  6 } { 7.5  bp } ,
          { -5 } { 9    bp } ,
          {  5 } { 10.5 bp } ,
          { -4 } { 12   bp } ,
          {  4 } { 14   bp } ,
          { -3 } { 15   bp } ,
          {  3 } { 16   bp } ,
          { -2 } { 18   bp } ,
          {  2 } { 22   bp } ,
          { -1 } { 24   bp } ,
          {  1 } { 26   bp } ,
          { -0 } { 36   bp } ,
          {  0 } { 42   bp }
        }
        { \__ctex_save_font_size:nn #1 }
    }
    { letterpress }
    {
      \clist_map_inline:nn
        {
          {  8 } { 4    bp } ,
          {  7 } { 5.25 bp } ,
          { -6 } { 7.5  bp } ,
          {  6 } { 8    bp } ,
          { -5 } { 9    bp } ,
          {  5 } { 10.5 bp } ,
          { -4 } { 12   bp } ,
          {  4 } { 14   bp } ,
          { -3 } { 15   bp } ,
          {  3 } { 16   bp } ,
          { -2 } { 18   bp } ,
          {  2 } { 21   bp } ,
          { -1 } { 24   bp } ,
          {  1 } { 28   bp } ,
          { -0 } { 36   bp } ,
          {  0 } { 42   bp }
        }
        { \__ctex_save_font_size:nn #1 }
    }
  }
  {
    \file_if_exist:nTF { ctex-fontsize- \g__ctex_font_size_system_tl .def }
      {
        \ctex_file_input:n
          { ctex-fontsize- \g__ctex_font_size_system_tl .def }
      }
      {
        \msg_error:nne { ctex } { fontsize-system-not-found }
          { \g__ctex_font_size_system_tl }
        \clist_map_inline:nn
          {
            {  8 } { 5    bp } ,
            {  7 } { 5.5  bp } ,
            { -6 } { 6.5  bp } ,
            {  6 } { 7.5  bp } ,
            { -5 } { 9    bp } ,
```

```
                { 5 } { 10.5 bp } ,
                { -4 } { 12   bp } ,
                { 4 } { 14   bp } ,
                { -3 } { 15   bp } ,
                { 3 } { 16   bp } ,
                { -2 } { 18   bp } ,
                { 2 } { 22   bp } ,
                { -1 } { 24   bp } ,
                { 1 } { 26   bp } ,
                { -0 } { 36   bp } ,
                { 0 } { 42   bp }
              }
              { \__ctex_save_font_size:nn #1 }
          }
      }
    \exp_args:NNe \prop_const_from_keyval:Nn \c__ctex_font_size_prop
      { \prop_to_keyval:N \l__ctex_font_size_temp_prop }
    \exp_args:NNe \seq_const_from_clist:Nn \c__ctex_font_size_seq
      { \seq_use:Nn \l__ctex_font_size_temp_seq { , } }
\group_end:
\msg_new:nnnn { ctex } { fontsize-system-not-found }
  {
    CTeX~font~size~system~`#1'~could~not~be~found.
    \\ The~`word'~system~will~be~used~instead.
  }
  { You~may~need~to~create~`ctex-fontsize-#1.def'. }
```

\ctex_declare_math_sizes:nnnn

```
\cs_new_protected:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
  {
    \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
    \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
  }
```

__ctex_get_font_sizes:Nn

```
\cs_new_protected:Npn \__ctex_get_font_sizes:Nn #1#2
  {
    \tl_clear:N #1
    \tl_map_inline:nn {#2}
      {
        \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
          { \tl_put_right:Ne #1 { { \tl_head:N \l__ctex_tmp_tl } } }
          { \tl_put_right:Ne #1 { { \dim_to_decimal:n { ##1 } } } }
      }
  }

\clist_map_inline:nn
  {
    {  8 }{  8 }{ 5pt }{ 5pt } ,
    {  7 }{  7 }{ 5pt }{ 5pt } ,
    { -6 }{ -6 }{ 5pt }{ 5pt } ,
    {  6 }{  6 }{ 5pt }{ 5pt } ,
    { -5 }{ -5 }{ 6pt }{ 5pt } ,
    {  5 }{  5 }{ 7pt }{ 5pt } ,
    { -4 }{ -4 }{ 8pt }{ 6pt } ,
    {  4 }{  4 }{  5 }{  6 } ,
    { -3 }{ -3 }{ -4 }{ -5 } ,
    {  3 }{  3 }{  4 }{  5 } ,
    { -2 }{ -2 }{ -3 }{ -4 } ,
    {  2 }{  2 }{  3 }{  4 } ,
    { -1 }{ -1 }{ -2 }{ -3 } ,
    {  1 }{  1 }{  2 }{  3 } ,
    { -0 }{ -0 }{ -1 }{ -2 } ,
    {  0 }{  0 }{  1 }{  2 }
  }
  { \ctex_declare_math_sizes:nnnn #1 }
```

[20] http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190

15.10.2　修改默认字号大小

\ctex_set_font_size:Nnn
```
\cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
  {
    \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
      { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
      { \msg_error:nnn { ctex } { fontsize } {#2} }
  }
\cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
  { \cs_set_protected:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

\int_case:nn { \g__ctex_font_size_int }
  {
    { 0 } { \file_input:n { ctex-c5size.clo } }
    { 1 } { \file_input:n { ctex-cs4size.clo } }
  }
```

⟨/class|ctex|ctexsize⟩

⟨*c5size⟩
```
\ctex_set_font_size:Nnn \normalsize { 5 }
  {
    \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip
    \let\@listi\@listI
  }
\ctex_set_font_size:Nnn \small { -5 }
  {
    \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
    \abovedisplayshortskip \z@ \@plus2\p@
    \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
    \def\@listi{\leftmargin\leftmargini
                \topsep 4\p@ \@plus2\p@ \@minus2\p@
                \parsep 2\p@ \@plus\p@ \@minus\p@
                \itemsep \parsep}
    \belowdisplayskip \abovedisplayskip
  }
\ctex_set_font_size:Nnn \footnotesize { 6 }
  {
    \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
    \abovedisplayshortskip \z@ \@plus\p@
    \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
    \def\@listi{\leftmargin\leftmargini
                \topsep 3\p@ \@plus\p@ \@minus\p@
                \parsep 2\p@ \@plus\p@ \@minus\p@
                \itemsep \parsep}
    \belowdisplayskip \abovedisplayskip
  }
\ctex_set_font_size:Nnn \scriptsize { -6 } { }
\ctex_set_font_size:Nnn \tiny  {  7 } { }
\ctex_set_font_size:Nnn \large { -4 } { }
\ctex_set_font_size:Nnn \Large { -3 } { }
\ctex_set_font_size:Nnn \LARGE { -2 } { }
\ctex_set_font_size:Nnn \huge  {  2 } { }
\ctex_set_font_size:Nnn \Huge  {  1 } { }
```
⟨/c5size⟩
⟨*cs4size⟩
```
\ctex_set_font_size:Nnn \normalsize { -4 }
  {
    \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
    \belowdisplayskip \abovedisplayskip
    \let\@listi\@listI
  }
\ctex_set_font_size:Nnn \small { 5 }
  {
```

```
    \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
    \def\@listi{\leftmargin\leftmargini
                \topsep 9\p@ \@plus3\p@ \@minus5\p@
                \parsep 4.5\p@ \@plus2\p@ \@minus\p@
                \itemsep \parsep}
    \belowdisplayskip \abovedisplayskip
  }
\ctex_set_font_size:Nnn \footnotesize { -5 }
  {
    \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
    \def\@listi{\leftmargin\leftmargini
                \topsep 6\p@ \@plus2\p@ \@minus2\p@
                \parsep 3\p@ \@plus2\p@ \@minus\p@
                \itemsep \parsep}
    \belowdisplayskip \abovedisplayskip
  }
\ctex_set_font_size:Nnn \scriptsize { 6 } { }
\ctex_set_font_size:Nnn \tiny  { -6 } { }
\ctex_set_font_size:Nnn \large { -3 } { }
\ctex_set_font_size:Nnn \Large { -2 } { }
\ctex_set_font_size:Nnn \LARGE {  2 } { }
\ctex_set_font_size:Nnn \huge  { -1 } { }
\ctex_set_font_size:Nnn \Huge  {  1 } { }
⟨/cs4size⟩
⟨ctexsize⟩ \normalsize
⟨*class|ctex⟩
```

15.11　更新行距

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

```
\fp_if_nan:nF { \l__ctex_line_spread_fp }
  {
    \exp_args:Ne \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。

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

激活默认字体大小，更新行距、\parindent 和 \CJKglue。

```
\normalsize
```

15.12　其他功能

\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。CTEX 可以直接用在 PDF 书签中。

```
\NewDocumentCommand \CTeX { }
```

```
  { C \TeX }
\ctex_at_end_package:nn { hyperref }
  { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

captiondelimiter 过时选项。

```
\ctex_define:n
  {
    captiondelimiter .code:n =
      {
        \ctex_deprecated_option:n
          { You~can~load~the~package~`caption'~to~get~its~functionality. }
      }
  }
```

⟨/class|ctex⟩

15.12.1　列表环境的缩进

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

⟨*scheme&chinese&(article|book|report)⟩

```
\ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
\ctex_patch_cmd:Nnn \verse {  1.5em } {  2 \ccwd }
\ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
```

⟨/scheme&chinese&(article|book|report)⟩

⟨*class|ctex⟩

15.13　载入中文字库

\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。

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

\ctex_fontset_case:nnn 3 个参数依次为 pdfTEX、upTEX 和 XETEX/LuaTEX。

```
\cs_new:Npx \ctex_fontset_case:nnn #1#2#3
  {
    \sys_if_engine_pdftex:TF
      {#1}
      { \sys_if_engine_uptex:TF {#2} {#3} }
  }
```

\ctex_fontset_case:nnnn 4 个参数依次为 pdfTEX(生成 PDF)、pdfTEX(生成 DVI)、upTEX 和 XETEX/LuaTEX。

```
\cs_new:Npx \ctex_fontset_case:nnnn #1#2#3#4
  {
    \sys_if_engine_pdftex:TF
      { \sys_if_output_pdf:TF   {#1} {#2} }
      { \sys_if_engine_uptex:TF {#3} {#4} }
  }
```

\ctex_detect_platform: 根据操作系统判断默认字体配置。

```
\cs_new_protected:Npn \ctex_detect_platform:
  {
    \sys_if_platform_windows:TF
```

```
      { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
      {
        \ctex_if_platform_macos:TF
          { \tl_gset:Nn \g__ctex_fontset_tl { mac    } }
          { \tl_gset:Nn \g__ctex_fontset_tl { fandol } }
      }
  }
```

\ctex_if_platform_macos:TF 以特定字体判断 macOS 系统。

```
\cs_new_protected:Npn \ctex_if_platform_macos:TF
  { \file_if_exist:nTF { /System/Library/Fonts/Menlo.ttc } }
```

\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。

```
\cs_new_protected:Npn \ctex_load_fontset:
  {
    \tl_if_empty:NTF \g__ctex_fontset_tl
      { \ctex_detect_platform: }
      {
        \bool_lazy_or:nnTF
          { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsnew } }
          { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsold } }
          {
            \msg_warning:nnee { ctex } { deprecated-fontset }
              { \g__ctex_fontset_tl } { windows }
            \tl_gset:Nn \g__ctex_fontset_tl { windows }
          }
          {
            \file_if_exist:nF { ctex-fontset- \g__ctex_fontset_tl .def }
              {
                \use:e
                  {
                    \ctex_detect_platform:
                    \msg_error:nnee { ctex } { fontset-not-found }
                      { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
                  }
              }
          }
      }
    \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
  }
\msg_new:nnn { ctex } { deprecated-fontset }
  { CTeX~fontset~`#1'~is~deprecated.\\ Fontset~`#2'~will~be~used~instead. }
\msg_new:nnnn { ctex } { fontset-not-found }
  { CTeX~fontset~`#1'~could~not~be~found.\\ Fontset~`#2'~will~be~used~instead. }
  { You~may~run~`mktexlsr'~firstly. }
\@onlypreamble \ctex_load_fontset:
```

fontset 在导言区通过 \ctexset 载入中文字库的选项。

```
\ctex_define:n
  {
    fontset .code:n =
      {
        \ctex_if_preamble:TF
          {
            \str_if_eq:eeTF {#1} { none }
              { \msg_warning:nnn { ctex } { invalid-value } {#1} }
              {
                \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
                  {
                    \tl_gset:Ne \g__ctex_fontset_tl {#1}
                    \ctex_load_fontset:
                  }
                  {
```

```
                    \msg_error:nnee { ctex } { fontset-loaded }
                      { \g__ctex_fontset_tl } {#1}
                  }
              }
          }
          { \msg_error:nn { ctex } { fontset-only-preamble } }
      }
  }
\msg_new:nnnn { ctex } { fontset-loaded }
  {
    CTeX~fontset~`#1'~has~been~loaded.
    \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
  }
  { Only~one~fontset~can~be~loaded~in~the~preamble. }
\msg_new:nnn { ctex } { fontset-only-preamble }
  { The~`fontset'~option~can~be~used~only~in~preamble. }
```

载入中文字库。

```
\str_if_eq:onF { \g__ctex_fontset_tl } { none }
  { \ctex_load_fontset: }
```

15.14　宏包配置文件

15.14.1　ctex.cfg

```
\ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
```

⟨/class|ctex⟩

⟨*config⟩
%%
⟨/config⟩

15.14.2　ctexopts.cfg

这里仅为配置文件示例：使用 Windows 下的字体设置。

⟨*ctexopts⟩
%%
%% \ctex_set:nn { option } { fontset = windows }
⟨/ctexopts⟩

15.15　字体定义文件

15.15.1　传统定义方式

⟨*c19|c70⟩
%%
%% Chinese characters
%%
⟨c19⟩ %% character set: GBK (extension of GB 2312)
⟨c70⟩ %% character set: Unicode
%% font encoding: Unicode
%%
⟨/c19|c70⟩

CJK 宏包使用的字体族。

```
⟨rm&c19⟩ \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
⟨rm&c70⟩ \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
 ⟨sf&c19⟩ \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
 ⟨sf&c70⟩ \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
 ⟨tt&c19⟩ \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
```

```
⟨tt&c70⟩ 4601 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}
```

upLaTeX 使用的字体族。upLaTeX 在 NFSS 下使用字体编码 JY2 和 JT2 来分别表示横排与直排的日文。

```
⟨rm&jy2⟩ 4602 \DeclareKanjiFamily{JY2}{zhrm}{}
⟨rm&jt2⟩ 4603 \DeclareKanjiFamily{JT2}{zhrm}{}
⟨sf&jy2⟩ 4604 \DeclareKanjiFamily{JY2}{zhsf}{}
⟨sf&jt2⟩ 4605 \DeclareKanjiFamily{JT2}{zhsf}{}
⟨tt&jy2⟩ 4606 \DeclareKanjiFamily{JY2}{zhtt}{}
⟨tt&jt2⟩ 4607 \DeclareKanjiFamily{JT2}{zhtt}{}
```

在 upLaTeX 字体族 zhrm 下采用如下 Fallback 机制：形状 sl 优先保留加粗字重，形状 it 优先保留楷书字形。

```
4608 ⟨*rm⟩
4609 ⟨*c19⟩
4610 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
4611 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
4612 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
4613 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
4614 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4615 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4616 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
4617 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
4618 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
4619 ⟨/c19⟩
4620 ⟨*c70⟩
4621 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
4622 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
4623 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
4624 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
4625 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4626 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4627 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
4628 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
4629 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
4630 ⟨/c70⟩
4631 ⟨*jy2⟩
4632 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{}
4633 \DeclareFontShape{JY2}{zhrm}{b}{n}{<-> upzhserifb-h}{}
4634 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{}
4635 \DeclareFontShape{JY2}{zhrm}{m}{sl}{<-> upzhserifit-h}{}
4636 \DeclareFontShape{JY2}{zhrm}{b}{sl}{<-> upzhserifb-h}{}
4637 \DeclareFontShape{JY2}{zhrm}{bx}{sl}{<-> upzhserifb-h}{}
4638 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{}
4639 \DeclareFontShape{JY2}{zhrm}{b}{it}{<-> upzhserifit-h}{}
4640 \DeclareFontShape{JY2}{zhrm}{bx}{it}{<-> upzhserifit-h}{}
4641 ⟨/jy2⟩
4642 ⟨*jt2⟩
4643 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{}
4644 \DeclareFontShape{JT2}{zhrm}{b}{n}{<-> upzhserifb-v}{}
4645 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{}
4646 \DeclareFontShape{JT2}{zhrm}{m}{sl}{<-> upzhserifit-v}{}
4647 \DeclareFontShape{JT2}{zhrm}{b}{sl}{<-> upzhserifb-v}{}
4648 \DeclareFontShape{JT2}{zhrm}{bx}{sl}{<-> upzhserifb-v}{}
4649 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{}
4650 \DeclareFontShape{JT2}{zhrm}{b}{it}{<-> upzhserifit-v}{}
4651 \DeclareFontShape{JT2}{zhrm}{bx}{it}{<-> upzhserifit-v}{}
4652 ⟨/jt2⟩
4653 ⟨/rm⟩

4654 ⟨*sf⟩
4655 ⟨*c19⟩
4656 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4657 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4658 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4659 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
```

```
\DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
\DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
\DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
\DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
\DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
⟨/c19⟩
⟨*c70⟩
\DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
\DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
\DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
\DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
\DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
\DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
\DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
\DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
\DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
⟨/c70⟩
⟨*jy2⟩
\DeclareFontShape{JY2}{zhsf}{m}{n}{<-> upzhsans-h}{}
\DeclareFontShape{JY2}{zhsf}{b}{n}{<-> upzhsansb-h}{}
\DeclareFontShape{JY2}{zhsf}{bx}{n}{<-> upzhsansb-h}{}
\DeclareFontShape{JY2}{zhsf}{m}{sl}{<-> upzhsans-h}{}
\DeclareFontShape{JY2}{zhsf}{b}{sl}{<-> upzhsansb-h}{}
\DeclareFontShape{JY2}{zhsf}{bx}{sl}{<-> upzhsansb-h}{}
\DeclareFontShape{JY2}{zhsf}{m}{it}{<-> upzhsans-h}{}
\DeclareFontShape{JY2}{zhsf}{b}{it}{<-> upzhsansb-h}{}
\DeclareFontShape{JY2}{zhsf}{bx}{it}{<-> upzhsansb-h}{}
⟨/jy2⟩
⟨*jt2⟩
\DeclareFontShape{JT2}{zhsf}{m}{n}{<-> upzhsans-v}{}
\DeclareFontShape{JT2}{zhsf}{b}{n}{<-> upzhsansb-v}{}
\DeclareFontShape{JT2}{zhsf}{bx}{n}{<-> upzhsansb-v}{}
\DeclareFontShape{JT2}{zhsf}{m}{sl}{<-> upzhsans-v}{}
\DeclareFontShape{JT2}{zhsf}{b}{sl}{<-> upzhsansb-v}{}
\DeclareFontShape{JT2}{zhsf}{bx}{sl}{<-> upzhsansb-v}{}
\DeclareFontShape{JT2}{zhsf}{m}{it}{<-> upzhsans-v}{}
\DeclareFontShape{JT2}{zhsf}{b}{it}{<-> upzhsansb-v}{}
\DeclareFontShape{JT2}{zhsf}{bx}{it}{<-> upzhsansb-v}{}
⟨/jt2⟩
⟨/sf⟩

⟨*tt⟩
⟨*c19⟩
\DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
\DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
\DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
\DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
\DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
\DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
\DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
\DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
\DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
⟨/c19⟩
⟨*c70⟩
\DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
\DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
\DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
\DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
\DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
\DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
\DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
\DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
\DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
⟨/c70⟩
⟨*jy2⟩
\DeclareFontShape{JY2}{zhtt}{m}{n}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{b}{n}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{bx}{n}{<-> upzhmono-h}{}
```

```
\DeclareFontShape{JY2}{zhtt}{m}{sl}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{b}{sl}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{bx}{sl}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{m}{it}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{b}{it}{<-> upzhmono-h}{}
\DeclareFontShape{JY2}{zhtt}{bx}{it}{<-> upzhmono-h}{}
⟨/jy2⟩
⟨*jt2⟩
\DeclareFontShape{JT2}{zhtt}{m}{n}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{b}{n}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{bx}{n}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{m}{sl}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{b}{sl}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{bx}{sl}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{m}{it}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{b}{it}{<-> upzhmono-v}{}
\DeclareFontShape{JT2}{zhtt}{bx}{it}{<-> upzhmono-v}{}
⟨/jt2⟩
⟨/tt⟩
```

15.15.2　预定义字库

```
⟨*fontset⟩
```

15.15.2.1　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。

```
⟨*adobe⟩
\ctex_fontset_case:nnnn
  { \ctex_fontset_error:n { adobe } }
  {
    \ctex_zhmap_case:nnn
      {
        \setCJKmainfont  { AdobeSongStd-Light.otf }
          [
            cmap       = UniGB-UTF16-H,
            BoldFont   = AdobeHeitiStd-Regular.otf,
            ItalicFont = AdobeKaitiStd-Regular.otf
          ]
        \setCJKsansfont { AdobeHeitiStd-Regular.otf }
          [ cmap = UniGB-UTF16-H ]
        \setCJKmonofont { AdobeFangsongStd-Regular.otf }
          [ cmap = UniGB-UTF16-H ]
        \setCJKfamilyfont { zhsong } { AdobeSongStd-Light.otf       }
          [ cmap = UniGB-UTF16-H ]
        \setCJKfamilyfont { zhhei  } { AdobeHeitiStd-Regular.otf    }
          [ cmap = UniGB-UTF16-H ]
        \setCJKfamilyfont { zhkai  } { AdobeKaitiStd-Regular.otf    }
          [ cmap = UniGB-UTF16-H ]
        \setCJKfamilyfont { zhfs   } { AdobeFangsongStd-Regular.otf }
          [ cmap = UniGB-UTF16-H ]
        \ctex_punct_set:n { adobe }
        \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
        \ctex_punct_map_family:nn   { \CJKsfdefault } { zhhei  }
        \ctex_punct_map_family:nn   { \CJKttdefault } { zhfs   }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
      }
      {
        \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { adobe }
        \ctex_punct_set:n { adobe }
        \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
      }
```

```
      { \ctex_fontset_error:n { adobe } }
  }
  {
    \ctex_set_upfonts:nnnnnn
      { AdobeSongStd-Light.otf       }
      { AdobeHeitiStd-Regular.otf    }
      { AdobeKaitiStd-Regular.otf    }
      { AdobeHeitiStd-Regular.otf    }
      { AdobeHeitiStd-Regular.otf    }
      { AdobeFangsongStd-Regular.otf }
    \ctex_set_upfamily:nnn { zhsong } { upzhserif   } {}
    \ctex_set_upfamily:nnn { zhhei  } { upzhsans    } {}
    \ctex_set_upfamily:nnn { zhfs   } { upzhmono    } {}
    \ctex_set_upfamily:nnn { zhkai  } { upzhserifit } {}
  }
  {
    \setCJKmainfont { AdobeSongStd-Light       }
      [ BoldFont = AdobeHeitiStd-Regular, ItalicFont = AdobeKaitiStd-Regular ]
    \setCJKsansfont { AdobeHeitiStd-Regular    }
    \setCJKmonofont { AdobeFangsongStd-Regular }
    \setCJKfamilyfont { zhsong } { AdobeSongStd-Light       }
    \setCJKfamilyfont { zhhei  } { AdobeHeitiStd-Regular    }
    \setCJKfamilyfont { zhfs   } { AdobeFangsongStd-Regular }
    \setCJKfamilyfont { zhkai  } { AdobeKaitiStd-Regular    }
  }
⟨/adobe⟩
```

15.15.2.2 fandol

```
⟨*fandol⟩
\ctex_fontset_case:nnnn
  { \ctex_fontset_error:n { fandol } }
  {
    \ctex_zhmap_case:nnn
      {
        \setCJKmainfont { FandolSong-Regular.otf }
          [
            cmap       = UniGB-UTF16-H,
            BoldFont   = FandolSong-Bold.otf,
            ItalicFont = FandolKai-Regular.otf
          ]
        \setCJKsansfont { FandolHei-Regular.otf }
          [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
        \setCJKmonofont { FandolFang-Regular.otf }
          [ cmap = UniGB-UTF16-H ]
        \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf }
          [ cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf ]
        \setCJKfamilyfont { zhhei  } { FandolHei-Regular.otf  }
          [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf  ]
        \setCJKfamilyfont { zhfs   } { FandolFang-Regular.otf }
          [ cmap = UniGB-UTF16-H ]
        \setCJKfamilyfont { zhkai  } { FandolKai-Regular.otf  }
          [ cmap = UniGB-UTF16-H ]
        \ctex_punct_set:n { fandol }
        \ctex_punct_map_family:nn   { \CJKrmdefault         } { zhsong  }
        \ctex_punct_map_family:nn   { \CJKsfdefault         } { zhhei   }
        \ctex_punct_map_family:nn   { \CJKttdefault         } { zhfs    }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
        \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei  } { zhheib  }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault         } { zhkai   }
      }
      {
        \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { fandol }
        \ctex_punct_set:n { fandol }
        \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
```

```
        \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
      }
      { \ctex_fontset_error:n { fandol } }
  }
  {
    \ctex_set_upfonts:nnnnnn
      { FandolSong-Regular.otf }
      { FandolSong-Bold.otf    }
      { FandolKai-Regular.otf  }
      { FandolHei-Regular.otf  }
      { FandolHei-Bold.otf     }
      { FandolFang-Regular.otf }
    \ctex_set_upfamily:nnn { zhsong } { upzhserif   } { upzhserifb }
    \ctex_set_upfamily:nnn { zhhei  } { upzhsans    } { upzhsansb  }
    \ctex_set_upfamily:nnn { zhfs   } { upzhmono    } {}
    \ctex_set_upfamily:nnn { zhkai  } { upzhserifit } {}
  }
  {
    \setCJKmainfont { FandolSong-Regular }
      [
        Extension  = .otf,
        BoldFont   = FandolSong-Bold,
        ItalicFont = FandolKai-Regular
      ]
    \setCJKsansfont { FandolHei-Regular  }
      [ Extension = .otf, BoldFont = FandolHei-Bold ]
    \setCJKmonofont { FandolFang-Regular }
      [ Extension = .otf ]
    \setCJKfamilyfont { zhsong } { FandolSong-Regular }
      [ Extension = .otf, BoldFont = FandolSong-Bold ]
    \setCJKfamilyfont { zhhei  } { FandolHei-Regular  }
      [ Extension = .otf, BoldFont = FandolHei-Bold  ]
    \setCJKfamilyfont { zhfs   } { FandolFang-Regular }
      [ Extension = .otf ]
    \setCJKfamilyfont { zhkai  } { FandolKai-Regular  }
      [ Extension = .otf ]
  }
⟨/fandol⟩
```

15.15.2.3 founder

```
⟨*founder⟩
\ctex_fontset_case:nnn
  {
    \ctex_zhmap_case:nnn
      {
        \setCJKmainfont { FZSSK.TTF }
          [ BoldFont = FZXBSK.TTF, ItalicFont = FZKTK.TTF ]
        \setCJKsansfont { FZXH1K.TTF } [ BoldFont = FZHTK.TTF ]
        \setCJKmonofont { FZFSK.TTF }
        \setCJKfamilyfont { zhsong } { FZSSK.TTF } [ BoldFont = FZXBSK.TTF ]
        \setCJKfamilyfont { zhhei  } { FZHTK.TTF }
        \setCJKfamilyfont { zhkai  } { FZKTK.TTF }
        \setCJKfamilyfont { zhfs   } { FZFSK.TTF }
        \setCJKfamilyfont { zhli   } { FZLSK.TTF }
        \setCJKfamilyfont { zhyou  } { FZY1K.TTF } [ BoldFont = FZY3K.TTF ]
        \ctex_punct_set:n { founder }
        \ctex_punct_map_family:nn   { \CJKrmdefault         } { zhsong  }
        \ctex_punct_map_family:nn   { \CJKsfdefault         } { zhheil  }
        \ctex_punct_map_family:nn   { \CJKttdefault         } { zhfs    }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault         } { zhkai   }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
        \ctex_punct_map_bfseries:nn { \CJKsfdefault         } { zhhei   }
        \ctex_punct_map_bfseries:nn { zhyou                 } { zhyoub  }
      }
      {
        \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { founder }
```

```
4912        \ctex_punct_set:n { founder }
4913        \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
4914        \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
4915        \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
4916      }
4917      { \ctex_fontset_error:n { founder } }
4918  }
4919  {
4920    \ctex_set_upfonts:nnnnnn
4921      { FZSSK.TTF  }
4922      { FZXBSK.TTF }
4923      { FZKTK.TTF  }
4924      { FZXH1K.TTF }
4925      { FZHTK.TTF  }
4926      { FZFSK.TTF  }
4927    \ctex_set_upfamily:nnn { zhsong } { upzhserif   } { upzhserifb }
4928    \ctex_set_upfamily:nnn { zhhei  } { upzhsans    } { upzhsansb  }
4929    \ctex_set_upfamily:nnn { zhfs   } { upzhmono    } {}
4930    \ctex_set_upfamily:nnn { zhkai  } { upzhserifit } {}
4931    \ctex_set_upfamily:nnn { zhli   } { upschrm     } {}
4932    \ctex_set_upfamily:nnn { zhyou  } { upschgt     } {}
4933    \ctex_set_upmap:nnn    { upstsl } { FZLSK.TTF } {}
4934    \ctex_set_upmap:nnn    { upstht } { FZY1K.TTF } {}
4935  }
4936  {
4937    \setCJKmainfont { FZShuSong-Z01 }
4938      [ BoldFont = FZXiaoBiaoSong-B05, ItalicFont = FZKai-Z03 ]
4939    \setCJKsansfont { FZXiHeiI-Z08 } [ BoldFont = FZHei-B01 ]
4940    \setCJKmonofont { FZFangSong-Z02 }
4941    \setCJKfamilyfont { zhsong } { FZShuSong-Z01  }
4942      [ BoldFont = FZXiaoBiaoSong-B05 ]
4943    \setCJKfamilyfont { zhhei  } { FZHei-B01      }
4944    \setCJKfamilyfont { zhkai  } { FZKai-Z03      }
4945    \setCJKfamilyfont { zhfs   } { FZFangSong-Z02 }
4946    \setCJKfamilyfont { zhli   } { FZLiShu-S01    }
4947    \setCJKfamilyfont { zhyou  } { FZXiYuan-M01   }
4948      [ BoldFont = FZZhunYuan-M02 ]
4949  }
4950 ⟨/founder⟩
```

15.15.2.4　hanyi

```
4951 ⟨*hanyi⟩
4952 \ctex_fontset_case:nnn
4953   {
4954     \ctex_zhmap_case:nnn
4955       {
4956         \setCJKmainfont { HYShuSongErS.ttf }
4957           [ BoldFont = HYZhongSongS.ttf, ItalicFont = HYKaiTiS.ttf ]
4958         \setCJKsansfont { HYZhongHeiS.ttf  } [ BoldFont = HYDaHeiS.ttf ]
4959         \setCJKmonofont { HYFangSongS.ttf  }
4960         \setCJKfamilyfont { zhsong } { HYShuSongErS.ttf }
4961           [ BoldFont = HYZhongSongS.ttf ]
4962         \setCJKfamilyfont { zhhei  } { HYZhongHeiS.ttf  }
4963           [ BoldFont = HYDaHeiS.ttf     ]
4964         \setCJKfamilyfont { zhkai  } { HYKaiTiS.ttf     }
4965         \setCJKfamilyfont { zhfs   } { HYFangSongS.ttf  }
4966         \ctex_punct_set:n { hanyi }
4967         \ctex_punct_map_family:nn   { \CJKrmdefault         } { zhsong  }
4968         \ctex_punct_map_family:nn   { \CJKsfdefault         } { zhhei   }
4969         \ctex_punct_map_family:nn   { \CJKttdefault         } { zhfs    }
4970         \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4971         \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei  } { zhheib  }
4972         \ctex_punct_map_itshape:nn  { \CJKrmdefault         } { zhkai   }
4973       }
4974       {
4975         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { hanyi }
```

```
      \ctex_punct_set:n { hanyi }
      \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
      \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
      \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
    }
    { \ctex_fontset_error:n { hanyi } }
  }
  {
    \ctex_set_upfonts:nnnnnn
      { HYShuSongErS.ttf }
      { HYZhongSongS.ttf }
      { HYKaiTiS.ttf     }
      { HYZhongHeiS.ttf  }
      { HYDaHeiS.ttf     }
      { HYFangSongS.ttf  }
    \ctex_set_upfamily:nnn { zhsong } { upzhserif   } { upzhserifb }
    \ctex_set_upfamily:nnn { zhhei  } { upzhsans    } { upzhsansb  }
    \ctex_set_upfamily:nnn { zhfs   } { upzhmono    } {}
    \ctex_set_upfamily:nnn { zhkai  } { upzhserifit } {}
  }
  {
    \setCJKmainfont { HYShuSongEr~S }
      [ BoldFont = HYZhongSong~S, ItalicFont = HYKaiTi~S ]
    \setCJKsansfont { HYZhongHei~S  } [ BoldFont = HYDaHei~S ]
    \setCJKmonofont { HYFangSong~S  }
    \setCJKfamilyfont { zhsong } { HYShuSongEr~S }
      [ BoldFont = HYZhongSong~S ]
    \setCJKfamilyfont { zhhei  } { HYZhongHei~S  }
      [ BoldFont = HYDaHei~S     ]
    \setCJKfamilyfont { zhkai  } { HYKaiTi~S     }
    \setCJKfamilyfont { zhfs   } { HYFangSong~S  }
  }
⟨/hanyi⟩
```

15.15.2.5　mac 相关

按 Issue 351 的讨论，以 El Capitan 为分界，分别设置 `macold`（El Capitan 之前）和 `macnew`(El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。从 macOS 15 (Sequoia) 开始, 苹方字体转为 *downloadable*, 不再位于传统路径, 故增加版本号检测作为后备。

```
⟨*mac⟩
\tl_new:N \g__ctex_macos_ver_tl
\msg_new:nnn { ctex } { macos-version-detect-failed }
  { Cannot~detect~macOS~version.~Falling~back~to~macold~fontset. }
```

为方便人类阅读, 把 LuaTEX 分支挪到后面, 在不忽略空格的 catcode scheme 下编写。这样 Lua 代码就不必以 ~ 代替空格。非 LuaTEX 分支, 在末尾使用 `\use_none_delimit_by_q_stop:w` 来跳过 LuaTEX 分支。

```
\sys_if_engine_luatex:F
  {
    \ior_new:N \g__ctex_macos_ver_ior
    \bool_new:N \g__ctex_macos_ver_found_bool
    \file_if_exist:nT { /System/Library/CoreServices/SystemVersion.plist }
      {
        \ior_open:Nn \g__ctex_macos_ver_ior
          { /System/Library/CoreServices/SystemVersion.plist }
        \ior_map_inline:Nn \g__ctex_macos_ver_ior
          {
            \bool_if:NTF \g__ctex_macos_ver_found_bool
              {
                \tl_set:Nn \l_tmpa_tl {#1}
                \regex_replace_once:nnN
                  { .*? <string> (\d+) .* } { \1 } \l_tmpa_tl
                \tl_gset:NV \g__ctex_macos_ver_tl \l_tmpa_tl
```

```
                \ior_map_break:
              }
              {
                \tl_if_in:nnT {#1} { ProductVersion }
                  { \bool_gset_true:N \g__ctex_macos_ver_found_bool }
              }
          }
        \ior_close:N \g__ctex_macos_ver_ior
      }
    \use_none_delimit_by_q_stop:w
  }
\char_set_catcode_space:n { 32 }
\lua_now:n
  {
    local f = io.open("/System/Library/CoreServices/SystemVersion.plist", "r");
    if f then
      local content = f:read("*a");
      f:close();
      local ver = content:match("ProductVersion.-<string>(.-)</string>");
      if ver then
        local dot = ver:find(".", 1, true);
        local major = dot and ver:sub(1, dot - 1) or ver;
        if major then
          token.set_macro("g__ctex_macos_ver_tl", major, "global");
        end;
      end;
    end
  }
\char_set_catcode_ignore:n { 32 }
\use_none_delimit_by_q_stop:w \q_stop
\file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
  { \ctex_file_input:n { ctex-fontset-macnew.def } }
  {
    \tl_if_empty:NTF \g__ctex_macos_ver_tl
      {
        \msg_warning:nn { ctex } { macos-version-detect-failed }
        \ctex_file_input:n { ctex-fontset-macold.def }
      }
      {
        \int_compare:nNnTF { \g__ctex_macos_ver_tl } < { 15 }
          { \ctex_file_input:n { ctex-fontset-macold.def } }
          { \ctex_file_input:n { ctex-fontset-macnew.def } }
      }
  }
⟨/mac⟩
```

macold 的设置参考了 OS X Mavericks (10.9) 预装的主要简体中文字体列表。

macnew 在默认字体设置方面，引入了多字重的宋体作为罗马字族，以及引入了苹方黑体作为无衬线字族。由于 Songti SC Light 的字重与 STSong 及 Windows 上的 SimSun 更接近，故默认字重使用 Songti SC Light，而不带后缀的正常字重事实上没有使用。黑体、圆体等设置也有类似的情况。

```
⟨*macold|macnew⟩
\ctex_fontset_case:nnnn
  {
    \ctex_fontset_error:n { mac }
    \use_none_delimit_by_q_stop:w
  }
⟨*macold⟩
  { \ctex_fontset_error:n { macold } }
  { \ctex_fontset_error:n { macold } }
⟨/macold⟩
⟨*macnew⟩
  {
    \ctex_zhmap_case:nnn
```

```
      {
        \setCJKmainfont { :3:Songti.ttc }
          [
            BoldFont       = :1:Songti.ttc,
            ItalicFont     = :0:Kaiti.ttc,
            BoldItalicFont = :3:Kaiti.ttc,
          ]
        \setCJKsansfont { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
        \setCJKmonofont { STFANGSO.ttf    }
        \setCJKfamilyfont { zhsong } { :3:Songti.ttc   } [ BoldFont = :1:Songti.ttc   ]
        \setCJKfamilyfont { zhhei  } { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
        \setCJKfamilyfont { zhkai  } { :0:Kaiti.ttc    } [ BoldFont = :3:Kaiti.ttc    ]
        \setCJKfamilyfont { zhfs   } { STFANGSO.ttf    }
        \setCJKfamilyfont { zhli   } { :0:Baoli.ttc    }
        \setCJKfamilyfont { zhyou  } { :4:Yuanti.ttc   } [ BoldFont = :0:Yuanti.ttc   ]
        \ctex_punct_set:n { mac }
        \ctex_punct_map_family:nn   { \CJKrmdefault         } { zhsong  }
        \ctex_punct_map_family:nn   { \CJKsfdefault         } { zhpf    }
        \ctex_punct_map_family:nn   { \CJKttdefault         } { zhfs    }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault         } { zhkai   }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
        \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei  } { zhpfb   }
        \ctex_punct_map_bfseries:nn { zhyou                 } { zhyoub  }
      }
      {
        \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { mac }
        \ctex_punct_set:n { mac }
        \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
        \ctex_punct_map_family:nn   { \CJKsfdefault } { zhpf   }
        \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhpf   }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
      }
      { \ctex_fontset_error:n { macnew } }
    \use_none_delimit_by_q_stop:w
  }
  {
    \ctex_set_upmap:nnn { upserif   } { :3:Songti.ttc } { :1:Songti.ttc }
    \ctex_set_upmap:nnn { upserifit } { :0:Kaiti.ttc  } { }
    \ctex_set_upmap:nnn { upstsl    } { :0:Baoli.ttc  } { }
    \ctex_set_upmap:nnn { upstht    } { :4:Yuanti.ttc } { }
    \ctex_set_upmap_unicode:nnn { upsans } { :2:PingFang.ttc } { :8:PingFang.ttc }
    \ctex_set_upmap_unicode:nnn { upmono } { STFANGSO.ttf    } { }
    \ctex_set_upfamily:nnn { zhsong } { upzhserif   } { upzhserifb }
    \ctex_set_upfamily:nnn { zhhei  } { upzhsans    } { upzhsansb  }
    \ctex_set_upfamily:nnn { zhfs   } { upzhmono    } { }
    \ctex_set_upfamily:nnn { zhkai  } { upzhserifit } { }
    \ctex_set_upfamily:nnn { zhli   } { upschrm     } { }
    \ctex_set_upfamily:nnn { zhyou  } { upschgt     } { }
    \use_none_delimit_by_q_stop:w
  }
⟨/macnew⟩
  {
⟨*macold⟩
    \setCJKmainfont { STSong     }
      [ BoldFont = STHeiti, ItalicFont = STKaiti ]
    \setCJKsansfont { STXihei    } [ BoldFont = STHeiti ]
    \setCJKmonofont { STFangsong }
    \setCJKfamilyfont { zhsong } { STSong     }
    \setCJKfamilyfont { zhhei  } { STHeiti    }
    \setCJKfamilyfont { zhfs   } { STFangsong }
    \setCJKfamilyfont { zhkai  } { STKaiti    }
⟨/macold⟩
⟨*macnew⟩
    \bool_new:N \g__ctex_mac_pingfang_bool
```

为方便人类阅读, 把 LuaTEX 分支挪到 `\ctex_fontset_case:nnnn` 之后, 在不忽略空格的 cat-

code scheme 下编写。这样 Lua 代码就不必以 ~ 代替空格。macnew 的其他 \ctex_fontset_case:nnnn 和其他 \ctex_zhmap_case:nnn 分支，末尾均使用 \use_none_delimit_by_q_stop:w 来跳过 LuaTEX 分支。

```
    \sys_if_engine_luatex:F
      {
        \fontspec_font_if_exist:nTF { Kaiti~SC }
          {
            \setCJKmainfont { Songti~SC~Light }
              [
                BoldFont       = Songti~SC~Bold ,
                ItalicFont     = Kaiti~SC ,
                BoldItalicFont = Kaiti~SC~Bold ,
              ]
          }
          {
            \setCJKmainfont { Songti~SC~Light }
              [ BoldFont = Songti~SC~Bold ]
          }
        \setCJKfamilyfont { zhsong } { Songti~SC~Light }
          [ BoldFont = Songti~SC~Bold ]
        \setCJKfamilyfont { zhhei  } { Heiti~SC~Light  }
          [ BoldFont = Heiti~SC~Medium ]
        \fontspec_font_if_exist:nTF { PingFang~SC }
          {
            \setCJKsansfont { PingFang~SC }
            \setCJKfamilyfont { zhpf } { PingFang~SC }
            \bool_gset_true:N \g__ctex_mac_pingfang_bool
          }
          { \setCJKsansfont { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ] }
        \fontspec_font_if_exist:nTF { STFangsong }
          {
            \setCJKmonofont { STFangsong }
            \setCJKfamilyfont { zhfs } { STFangsong }
          }
          { \setCJKmonofont { Heiti~SC~Light } }
        \fontspec_font_if_exist:nT { Kaiti~SC }
          {
            \setCJKfamilyfont { zhkai } { Kaiti~SC }
              [ BoldFont = Kaiti~SC~Bold ]
          }
        \fontspec_font_if_exist:nT { Baoli~SC }
          { \setCJKfamilyfont { zhli } { Baoli~SC } }
        \fontspec_font_if_exist:nT { Yuanti~SC~Light }
          {
            \setCJKfamilyfont { zhyou } { Yuanti~SC~Light }
              [ BoldFont = Yuanti~SC~Regular ]
          }
        \use_none_delimit_by_q_stop:w
      }
⟨/macnew⟩
  }
⟨*macnew⟩
\char_set_catcode_space:n { 32 }
\lua_now:n
  {
    local lfs = require("lfs");
    local search_bases = {};
    local asset_roots = {
      "/System/Library/AssetsV2",
      "/System/Library/AssetsV2/PreinstalledAssetsV2/InstallWithOs",
    };
    for _, root in ipairs(asset_roots) do
      local ok, iter, obj = pcall(lfs.dir, root);
      if ok then
        for d in iter, obj do
```

```
          if d:find("^com_apple_MobileAsset_Font") then
            table.insert(search_bases, root .. "/" .. d);
          end;
        end;
      end;
    end;
    local function find_font_dir(filename)
      for _, base in ipairs(search_bases) do
        local ok, iter, obj = pcall(lfs.dir, base);
        if ok then
          for entry in iter, obj do
            if not (entry == "." or entry == "..") then
              local p = base .. "/" .. entry
                .. "/AssetData/" .. filename;
              if lfs.attributes(p, "mode") then
                return base .. "/" .. entry .. "/AssetData/";
              end;
            end;
          end;
        end;
      end;
      return nil;
    end;
    local fonts = {
      {"PingFang.ttc", "g__ctex_mac_pingfang_dir_tl"},
      {"Kaiti.ttc",    "g__ctex_mac_kaiti_dir_tl"},
      {"STFANGSO.ttf", "g__ctex_mac_stfangso_dir_tl"},
      {"Baoli.ttc",    "g__ctex_mac_baoli_dir_tl"},
      {"Yuanti.ttc",   "g__ctex_mac_yuanti_dir_tl"},
    };
    for _, f in ipairs(fonts) do
      local dir = find_font_dir(f[1]);
      token.set_macro(f[2], dir or "", "global");
    end
  }
\char_set_catcode_ignore:n { 32 }
\tl_if_empty:NTF \g__ctex_mac_kaiti_dir_tl
  {
    \setCJKmainfont { Songti~SC~Light }
      [ BoldFont = Songti~SC~Bold ]
  }
  {
    \setCJKmainfont { Songti~SC~Light }
      [
        BoldFont       = Songti~SC~Bold ,
        ItalicFont     = Kaiti.ttc ,
        ItalicFeatures = { Path = \g__ctex_mac_kaiti_dir_tl , FontIndex = 0 } ,
        BoldItalicFont = Kaiti.ttc ,
        BoldItalicFeatures = { Path = \g__ctex_mac_kaiti_dir_tl , FontIndex = 3 } ,
      ]
  }
\setCJKfamilyfont { zhsong } { Songti~SC~Light }
  [ BoldFont = Songti~SC~Bold ]
\setCJKfamilyfont { zhhei  } { Heiti~SC~Light  }
  [ BoldFont = Heiti~SC~Medium ]
\tl_if_empty:NTF \g__ctex_mac_pingfang_dir_tl
  { \setCJKsansfont { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ] }
  {
    \setCJKsansfont { PingFang.ttc }
      [
        Path          = \g__ctex_mac_pingfang_dir_tl ,
        FontIndex     = 2 ,
        BoldFont      = PingFang.ttc ,
        BoldFontIndex = 8 ,
      ]
    \setCJKfamilyfont { zhpf } { PingFang.ttc }
      [ Path = \g__ctex_mac_pingfang_dir_tl , FontIndex = 2 ]
```

```
    \bool_gset_true:N \g__ctex_mac_pingfang_bool
  }
\tl_if_empty:NTF \g__ctex_mac_stfangso_dir_tl
  { \setCJKmonofont { Heiti~SC~Light } }
  {
    \setCJKmonofont { STFANGSO.ttf }
      [ Path = \g__ctex_mac_stfangso_dir_tl ]
    \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
      [ Path = \g__ctex_mac_stfangso_dir_tl ]
  }
\tl_if_empty:NF \g__ctex_mac_kaiti_dir_tl
  {
    \setCJKfamilyfont { zhkai } { Kaiti.ttc }
      [
        Path          = \g__ctex_mac_kaiti_dir_tl ,
        FontIndex     = 0 ,
        BoldFont      = Kaiti.ttc ,
        BoldFontIndex = 3 ,
      ]
  }
\tl_if_empty:NF \g__ctex_mac_baoli_dir_tl
  {
    \setCJKfamilyfont { zhli } { Baoli.ttc }
      [ Path = \g__ctex_mac_baoli_dir_tl , FontIndex = 0 ]
  }
\tl_if_empty:NF \g__ctex_mac_yuanti_dir_tl
  {
    \setCJKfamilyfont { zhyou } { Yuanti.ttc }
      [
        Path          = \g__ctex_mac_yuanti_dir_tl ,
        FontIndex     = 4 ,
        BoldFont      = Yuanti.ttc ,
        BoldFontIndex = 0 ,
      ]
  }
\use_none_delimit_by_q_stop:w \q_stop
⟨/macnew⟩
⟨/macold|macnew⟩
```

15.15.2.6　ubuntu

```
⟨*ubuntu⟩
\ctex_fontset_case:nnnn
  { \ctex_fontset_error:n { ubuntu } }
  {
    \ctex_zhmap_case:nnn
      {
        \setCJKmainfont { :2:NotoSerifCJK-Regular.ttc }
          [ BoldFont = :2:NotoSerifCJK-Bold.ttc, ItalicFont = gkai00mp.ttf ]
        \setCJKsansfont { :2:NotoSansCJK-Regular.ttc  }
          [ BoldFont = :2:NotoSansCJK-Bold.ttc  ]
        \setCJKmonofont { :2:NotoSerifCJK-Regular.ttc }
          [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
        \setCJKfamilyfont { zhsong } { :2:NotoSerifCJK-Regular.ttc }
          [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
        \setCJKfamilyfont { zhhei  } { :2:NotoSansCJK-Regular.ttc  }
          [ BoldFont = :2:NotoSansCJK-Bold.ttc  ]
        \setCJKfamilyfont { zhkai  } { gkai00mp.ttf  }
        \ctex_punct_set:n { ubuntu }
        \ctex_punct_map_family:nn   { \CJKrmdefault        } { zhsong  }
        \ctex_punct_map_family:nn   { \CJKsfdefault        } { zhhei   }
        \ctex_punct_map_family:nn   { \CJKttdefault        } { zhsong  }
        \ctex_punct_map_itshape:nn  { \CJKrmdefault        } { zhkai   }
        \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib  }
        \ctex_punct_map_bfseries:nn
          { \CJKrmdefault, \CJKttdefault, zhsong }
```

```
5343          { zhsongb }
5344      }
5345      {
5346        \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { ubuntu }
5347        \ctex_punct_set:n { ubuntu }
5348        \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
5349        \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
5350        \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
5351      }
5352      { \ctex_fontset_error:n { ubuntu } }
5353  }
5354  {
5355    \ctex_set_upmap_unicode:nnn { upserif }
5356      { :2:NotoSerifCJK-Regular.ttc } { :2:NotoSerifCJK-Bold.ttc }
5357    \ctex_set_upmap_unicode:nnn { upsans  }
5358      { :2:NotoSansCJK-Regular.ttc  } { :2:NotoSansCJK-Bold.ttc  }
5359    \ctex_set_upmap_unicode:nnn { upmono  }
5360      { :2:NotoSerifCJK-Regular.ttc } { }
5361    \ctex_set_upmap:nnn { upserifit } { gkai00mp.ttf  } { }
5362    \ctex_set_upfamily:nnn { zhsong } { upzhserif   } { upzhserifb }
5363    \ctex_set_upfamily:nnn { zhhei  } { upzhsans    } { upzhsansb  }
5364    \ctex_set_upfamily:nnn { zhkai  } { upzhserifit } { }
5365  }
5366  {
5367    \setCJKmainfont { Noto~Serif~CJK~SC }
5368      [ BoldFont = Noto~Serif~CJK~SC~Bold, ItalicFont = AR~PL~KaitiM~GB ]
5369    \setCJKsansfont { Noto~Sans~CJK~SC  }
5370      [ BoldFont = Noto~Sans~CJK~SC~Bold ]
5371    \setCJKmonofont { Noto~Serif~CJK~SC }
5372    \setCJKfamilyfont { zhsong } { Noto~Serif~CJK~SC }
5373      [ BoldFont = Noto~Serif~CJK~SC~Bold ]
5374    \setCJKfamilyfont { zhhei  } { Noto~Sans~CJK~SC  }
5375      [ BoldFont = Noto~Sans~CJK~SC~Bold ]
5376    \setCJKfamilyfont { zhkai  } { AR~PL~KaitiM~GB   }
5377  }
5378 ⟨/ubuntu⟩
```

15.15.2.7 windows

\c__ctex_msyh_suffix_tl 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。

```
5379 ⟨*windows⟩
5380 \file_if_exist:nTF { \c_dollar_str WINDIR/Fonts/msyh.ttc }
5381   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
5382   {
5383     \file_if_exist:nTF { msyh.ttc }
5384       { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
5385       { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttf } }
5386   }

5387 \ctex_fontset_case:nnn
5388   {
5389     \ctex_zhmap_case:nnn
5390       {
5391         \ctex_punct_set:n { windows }
5392         \setCJKmainfont { simsun.ttc }
5393           [ BoldFont = simhei.ttf, ItalicFont = simkai.ttf ]
5394         \setCJKsansfont { msyh.\c__ctex_msyh_suffix_tl }
5395           [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
5396         \setCJKmonofont { simfang.ttf }
5397         \setCJKfamilyfont { zhsong  } { simsun.ttc  }
5398         \setCJKfamilyfont { zhhei   } { simhei.ttf  }
5399         \setCJKfamilyfont { zhfs    } { simfang.ttf }
5400         \setCJKfamilyfont { zhkai   } { simkai.ttf  }
5401         \setCJKfamilyfont { zhyahei } { msyh.\c__ctex_msyh_suffix_tl }
5402           [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
5403         \setCJKfamilyfont { zhli    } { simli.ttf   }
```

```
5404          \setCJKfamilyfont { zhyou   } { simyou.ttf  }
5405          \ctex_punct_map_family:nn   { \CJKrmdefault          } { zhsong   }
5406          \ctex_punct_map_bfseries:nn { \CJKrmdefault          } { zhhei    }
5407          \ctex_punct_map_itshape:nn  { \CJKrmdefault          } { zhkai    }
5408          \ctex_punct_map_family:nn   { \CJKsfdefault          } { zhyahei  }
5409          \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhyahei } { zhyaheib }
5410          \ctex_punct_map_family:nn   { \CJKttdefault          } { zhfs     }
5411        }
5412        {
5413          \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { windows }
5414          \ctex_punct_set:n { windows }
5415          \ctex_punct_map_family:nn   { \CJKrmdefault } { zhsong }
5416          \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei  }
5417          \ctex_punct_map_itshape:nn  { \CJKrmdefault } { zhkai  }
5418        }
5419        { }
5420    }
5421    {
5422      \ctex_set_upfonts:nnnnnn
5423        { simsun.ttc                   }
5424        { simhei.ttf                   }
5425        { simkai.ttf                   }
5426        { msyh.\c__ctex_msyh_suffix_tl    }
5427        { msyhbd.\c__ctex_msyh_suffix_tl  }
5428        { simfang.ttf                  }
5429      \ctex_set_upfamily:nnn { zhsong  } { upzhserif   } {}
5430      \ctex_set_upfamily:nnn { zhhei   } { upzhserifb  } {}
5431      \ctex_set_upfamily:nnn { zhfs    } { upzhmono    } {}
5432      \ctex_set_upfamily:nnn { zhkai   } { upzhserifit } {}
5433      \ctex_set_upfamily:nnn { zhyahei } { upzhsans    } { upzhsansb }
5434      \ctex_set_upfamily:nnn { zhli    } { upschrm     } {}
5435      \ctex_set_upfamily:nnn { zhyou   } { upschgt     } {}
5436      \ctex_set_upmap:nnn { upstsl } { simli.ttf  } {}
5437      \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
5438    }
5439    {
5440      \setCJKmainfont   { SimSun } [ BoldFont = SimHei , ItalicFont = KaiTi ]
5441      \setCJKsansfont   { Microsoft~YaHei } [ BoldFont = *~Bold ]
5442      \setCJKmonofont   { FangSong }
5443      \setCJKfamilyfont { zhsong  } { SimSun          }
5444      \setCJKfamilyfont { zhhei   } { SimHei          }
5445      \setCJKfamilyfont { zhfs    } { FangSong        }
5446      \setCJKfamilyfont { zhkai   } { KaiTi           }
5447      \setCJKfamilyfont { zhyahei } { Microsoft~YaHei } [ BoldFont = *~Bold ]
5448      \setCJKfamilyfont { zhli    } { LiSu            }
5449      \setCJKfamilyfont { zhyou   } { YouYuan         }
5450    }
5451 ⟨/windows⟩
```

15.15.3　中文字体命令

\songti
\heiti
\fangsong
\kaishu
\lishu
\youyuan
\yahei
\pingfang

使用 upLaTeX 编译时，macnew 字库中由于传统黑体（黑体-简）无法使用，我们用苹方来代替。同时 \yahei、\pingfang 命令被设置为与 \heiti 相同。

```
5452 ⟨*!mac⟩
5453 \NewDocumentCommand \songti   { } { \CJKfamily { zhsong  } }
5454 \NewDocumentCommand \heiti    { } { \CJKfamily { zhhei   } }
5455 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs    } }
5456 \NewDocumentCommand \kaishu   { } { \CJKfamily { zhkai   } }
5457 ⟨*windows|founder|macnew⟩
5458 \NewDocumentCommand \lishu    { } { \CJKfamily { zhli    } }
5459 \NewDocumentCommand \youyuan  { } { \CJKfamily { zhyou   } }
5460 ⟨/windows|founder|macnew⟩
5461 \NewDocumentCommand \yahei    { } { \CJKfamily { zhyahei } }
5462 ⟨*macnew⟩
```

⟨!ubuntu⟩
⟨windows⟩

```
\bool_lazy_or:nnTF
  { \sys_if_engine_pdftex_p: }
  { \sys_if_engine_uptex_p:  }
  {
    \cs_new_eq:NN \yahei    \heiti
    \cs_new_eq:NN \pingfang \heiti
  }
  {
    \bool_if:NTF \g__ctex_mac_pingfang_bool
      {
        \NewDocumentCommand \yahei    { } { \CJKfamily { zhpf } }
        \NewDocumentCommand \pingfang { } { \CJKfamily { zhpf } }
      }
      {
        \cs_new_eq:NN \yahei    \heiti
        \cs_new_eq:NN \pingfang \heiti
      }
  }
⟨/macnew⟩
⟨/!mac⟩

⟨/fontset⟩
```

15.15.4　zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```
⟨*zhmap⟩
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode 35=6  % #
  \catcode 45=12 % -
  \catcode123=1  % {
  \catcode125=2  % }
  \toks0{\endlinechar=\the\endlinechar\relax}%
  \toks2{\endlinechar=-1 }%
  \def\x#1 #2 {%
    \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
    \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
  \x  13  5 % carriage return
  \x  32 10 % space
  \x  35  6 % #
  \x  40 12 % (
  \x  41 12 % )
  \x  45 12 % -
  \x  46 12 % .
  \x  47 12 % /
  \x  58 12 % :
  \x  60 12 % <
  \x  61 12 % =
  \x  64 11 % @
  \x  91 12 % [
  \x  93 12 % ]
  \x 123  1 % {
  \x 125  2 % }
  \edef\x#1{\endgroup%
    \edef\noexpand#1{%
      \the\toks0 %
      \let\noexpand\noexpand\noexpand#1%
          \noexpand\noexpand\noexpand\undefined%
      \noexpand\noexpand\noexpand\endinput}%
    \the\toks2}%
\expandafter\x\csname ctex@zhmap@endinput\endcsname

\ifzhmappdf \begingroup\expandafter\endgroup
\expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
  \expandafter\ifx\csname ifctexpdf\endcsname\relax
```

```
    \expandafter\ifx\csname pdfoutput\endcsname\relax
      iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
  \else ifctexpdf\fi
\endcsname
```

\ProvidesFile 提供非 LaTeX 格式下的 \ProvidesFile。

```
\begingroup
\expandafter\ifx\csname ProvidesFile\endcsname\relax
  \long\def\x#1\ProvidesFile#2[#3]{%
    #1%
    \immediate\write-1{File: #2 #3}%
    \expandafter\xdef\csname ver@#2\endcsname{#3}}
  \expandafter\x%
\fi
\endgroup
```

文件标识信息。

```
⟨adobe⟩ \ProvidesFile{ctex-zhmap-adobe.tex}%
⟨adobe⟩   [2022/07/14 v2.5.10 Adobe font map loader for DVIPDFMx (CTEX)]
⟨fandol⟩ \ProvidesFile{ctex-zhmap-fandol.tex}%
⟨fandol⟩   [2022/07/14 v2.5.10 Fandol font map loader for DVIPDFMx (CTEX)]
⟨founder⟩ \ProvidesFile{ctex-zhmap-founder.tex}%
⟨founder⟩   [2022/07/14 v2.5.10 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
⟨hanyi⟩ \ProvidesFile{ctex-zhmap-hanyi.tex}%
⟨hanyi⟩   [2022/07/14 v2.5.10 Hanyi font map loader for pdfTeX and DVIPDFMx (CTEX)]
⟨mac⟩ \ProvidesFile{ctex-zhmap-mac.tex}%
⟨mac⟩   [2022/07/14 v2.5.10 Mac font map loader for DVIPDFMx (CTEX)]
⟨ubuntu⟩ \ProvidesFile{ctex-zhmap-ubuntu.tex}%
⟨ubuntu⟩   [2022/07/14 v2.5.10 Ubuntu font map loader for DVIPDFMx (CTEX)]
⟨windows⟩ \ProvidesFile{ctex-zhmap-windows.tex}%
⟨windows⟩   [2022/07/14 v2.5.10 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
```

15.15.4.1 ctex-zhmap-adobe.tex

```
⟨*adobe⟩
\ifzhmappdf
%% pdfTeX does not support OTF fonts
\else
  \special{pdf:mapline gbk@UGBK@          UniGB-UTF16-H AdobeSongStd-Light.otf}
  \special{pdf:mapline gbksong@UGBK@      UniGB-UTF16-H AdobeSongStd-Light.otf}
  \special{pdf:mapline gbkkai@UGBK@       UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
  \special{pdf:mapline gbkhei@UGBK@       UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
  \special{pdf:mapline gbkfs@UGBK@        UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
  \special{pdf:mapline cyberb@Unicode@    UniGB-UTF16-H AdobeSongStd-Light.otf}
  \special{pdf:mapline unisong@Unicode@   UniGB-UTF16-H AdobeSongStd-Light.otf}
  \special{pdf:mapline unikai@Unicode@    UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
  \special{pdf:mapline unihei@Unicode@    UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
  \special{pdf:mapline unifs@Unicode@     UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
 \special{pdf:mapline gbksongsl@UGBK@   UniGB-UTF16-H AdobeSongStd-Light.otf     -s .167}
 \special{pdf:mapline gbkkaisl@UGBK@    UniGB-UTF16-H AdobeKaitiStd-Regular.otf   -s .167}
 \special{pdf:mapline gbkheisl@UGBK@    UniGB-UTF16-H AdobeHeitiStd-Regular.otf   -s .167}
 \special{pdf:mapline gbkfssl@UGBK@    UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf     -s .167}
 \special{pdf:mapline unikaisl@Unicode@  UniGB-UTF16-H AdobeKaitiStd-Regular.otf   -s .167}
 \special{pdf:mapline uniheisl@Unicode@  UniGB-UTF16-H AdobeHeitiStd-Regular.otf   -s .167}
 \special{pdf:mapline unifssl@Unicode@  UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
\fi
⟨/adobe⟩
```

15.15.4.2 ctex-zhmap-fandol.tex

```
⟨*fandol⟩
\ifzhmappdf
%% pdfTeX does not support OTF fonts
```

```
\else
  \special{pdf:mapline gbk@UGBK@          UniGB-UTF16-H FandolSong-Regular.otf}
  \special{pdf:mapline gbksong@UGBK@      UniGB-UTF16-H FandolSong-Regular.otf}
  \special{pdf:mapline gbkkai@UGBK@       UniGB-UTF16-H FandolKai-Regular.otf}
  \special{pdf:mapline gbkhei@UGBK@       UniGB-UTF16-H FandolHei-Regular.otf}
  \special{pdf:mapline gbkfs@UGBK@        UniGB-UTF16-H FandolFang-Regular.otf}
  \special{pdf:mapline cyberb@Unicode@    UniGB-UTF16-H FandolSong-Regular.otf}
  \special{pdf:mapline unisong@Unicode@   UniGB-UTF16-H FandolSong-Regular.otf}
  \special{pdf:mapline unikai@Unicode@    UniGB-UTF16-H FandolKai-Regular.otf}
  \special{pdf:mapline unihei@Unicode@    UniGB-UTF16-H FandolHei-Regular.otf}
  \special{pdf:mapline unifs@Unicode@     UniGB-UTF16-H FandolFang-Regular.otf}
  \special{pdf:mapline gbksongsl@UGBK@    UniGB-UTF16-H FandolSong-Regular.otf -s .167}
  \special{pdf:mapline gbkkaisl@UGBK@     UniGB-UTF16-H FandolKai-Regular.otf  -s .167}
  \special{pdf:mapline gbkheisl@UGBK@     UniGB-UTF16-H FandolHei-Regular.otf  -s .167}
  \special{pdf:mapline gbkfssl@UGBK@      UniGB-UTF16-H FandolFang-Regular.otf -s .167}
  \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
  \special{pdf:mapline unikaisl@Unicode@  UniGB-UTF16-H FandolKai-Regular.otf  -s .167}
  \special{pdf:mapline uniheisl@Unicode@  UniGB-UTF16-H FandolHei-Regular.otf  -s .167}
  \special{pdf:mapline unifssl@Unicode@   UniGB-UTF16-H FandolFang-Regular.otf -s .167}
\fi
⟨/fandol⟩
```

15.15.4.3 ctex-zhmap-founder.tex

```
⟨*founder⟩
\ifzhmappdf
  \pdfmapline{=gbk@UGBK@          <FZSSK.TTF}
  \pdfmapline{=gbksong@UGBK@      <FZSSK.TTF}
  \pdfmapline{=gbkkai@UGBK@       <FZKTK.TTF}
  \pdfmapline{=gbkhei@UGBK@       <FZHTK.TTF}
  \pdfmapline{=gbkfs@UGBK@        <FZFSK.TTF}
  \pdfmapline{=gbkli@UGBK@        <FZLSK.TTF}
  \pdfmapline{=gbkyou@UGBK@       <FZY1K.TTF}
  \pdfmapline{=cyberb@Unicode@    <FZSSK.TTF}
  \pdfmapline{=unisong@Unicode@   <FZSSK.TTF}
  \pdfmapline{=unikai@Unicode@    <FZKTK.TTF}
  \pdfmapline{=unihei@Unicode@    <FZHTK.TTF}
  \pdfmapline{=unifs@Unicode@     <FZFSK.TTF}
  \pdfmapline{=unili@Unicode@     <FZLSK.TTF}
  \pdfmapline{=uniyou@Unicode@    <FZY1K.TTF}
  \pdfmapline{=gbksongsl@UGBK@    <FZSSK.TTF}
  \pdfmapline{=gbkkaisl@UGBK@     <FZKTK.TTF}
  \pdfmapline{=gbkheisl@UGBK@     <FZHTK.TTF}
  \pdfmapline{=gbkfssl@UGBK@      <FZFSK.TTF}
  \pdfmapline{=gbklisl@UGBK@      <FZLSK.TTF}
  \pdfmapline{=gbkyousl@UGBK@     <FZY1K.TTF}
  \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
  \pdfmapline{=unikaisl@Unicode@  <FZKTK.TTF}
  \pdfmapline{=uniheisl@Unicode@  <FZHTK.TTF}
  \pdfmapline{=unifssl@Unicode@   <FZFSK.TTF}
  \pdfmapline{=unilisl@Unicode@   <FZLSK.TTF}
  \pdfmapline{=uniyousl@Unicode@  <FZY1K.TTF}
\else
  \special{pdf:mapline gbk@UGBK@          unicode FZSSK.TTF}
  \special{pdf:mapline gbksong@UGBK@      unicode FZSSK.TTF}
  \special{pdf:mapline gbkkai@UGBK@       unicode FZKTK.TTF}
  \special{pdf:mapline gbkhei@UGBK@       unicode FZHTK.TTF}
  \special{pdf:mapline gbkfs@UGBK@        unicode FZFSK.TTF}
  \special{pdf:mapline gbkli@UGBK@        unicode FZLSK.TTF}
  \special{pdf:mapline gbkyou@UGBK@       unicode FZY1K.TTF}
  \special{pdf:mapline cyberb@Unicode@    unicode FZSSK.TTF}
  \special{pdf:mapline unisong@Unicode@   unicode FZSSK.TTF}
  \special{pdf:mapline unikai@Unicode@    unicode FZKTK.TTF}
  \special{pdf:mapline unihei@Unicode@    unicode FZHTK.TTF}
  \special{pdf:mapline unifs@Unicode@     unicode FZFSK.TTF}
  \special{pdf:mapline unili@Unicode@     unicode FZLSK.TTF}
  \special{pdf:mapline uniyou@Unicode@    unicode FZY1K.TTF}
```

```
  \special{pdf:mapline gbksongsl@UGBK@    unicode FZSSK.TTF -s .167}
  \special{pdf:mapline gbkkaisl@UGBK@     unicode FZKTK.TTF -s .167}
  \special{pdf:mapline gbkheisl@UGBK@     unicode FZHTK.TTF -s .167}
  \special{pdf:mapline gbkfssl@UGBK@      unicode FZFSK.TTF -s .167}
  \special{pdf:mapline gbklisl@UGBK@      unicode FZLSK.TTF -s .167}
  \special{pdf:mapline gbkyousl@UGBK@     unicode FZY1K.TTF -s .167}
  \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
  \special{pdf:mapline unikaisl@Unicode@  unicode FZKTK.TTF -s .167}
  \special{pdf:mapline uniheisl@Unicode@  unicode FZHTK.TTF -s .167}
  \special{pdf:mapline unifssl@Unicode@   unicode FZFSK.TTF -s .167}
  \special{pdf:mapline unilisl@Unicode@   unicode FZLSK.TTF -s .167}
  \special{pdf:mapline uniyousl@Unicode@  unicode FZY1K.TTF -s .167}
\fi
⟨/founder⟩
```

15.15.4.4 ctex-zhmap-hanyi.tex

```
⟨*hanyi⟩
\ifzhmappdf
  \pdfmapline{=gbk@UGBK@          <HYShuSongErS.ttf}
  \pdfmapline{=gbksong@UGBK@      <HYShuSongErS.ttf}
  \pdfmapline{=gbkkai@UGBK@       <HYKaiTiS.ttf}
  \pdfmapline{=gbkhei@UGBK@       <HYZhongHeiS.ttf}
  \pdfmapline{=gbkfs@UGBK@        <HYFangSongS.ttf}
  \pdfmapline{=cyberb@Unicode@    <HYShuSongErS.ttf}
  \pdfmapline{=unisong@Unicode@   <HYShuSongErS.ttf}
  \pdfmapline{=unikai@Unicode@    <HYKaiTiS.ttf}
  \pdfmapline{=unihei@Unicode@    <HYZhongHeiS.ttf}
  \pdfmapline{=unifs@Unicode@     <HYFangSongS.ttf}
  \pdfmapline{=gbksongsl@UGBK@    <HYShuSongErS.ttf}
  \pdfmapline{=gbkkaisl@UGBK@     <HYKaiTiS.ttf}
  \pdfmapline{=gbkheisl@UGBK@     <HYZhongHeiS.ttf}
  \pdfmapline{=gbkfssl@UGBK@      <HYFangSongS.ttf}
  \pdfmapline{=unisongsl@Unicode@ <HYShuSongErS.ttf}
  \pdfmapline{=unikaisl@Unicode@  <HYKaiTiS.ttf}
  \pdfmapline{=uniheisl@Unicode@  <HYZhongHeiS.ttf}
  \pdfmapline{=unifssl@Unicode@   <HYFangSongS.ttf}
\else
  \special{pdf:mapline gbk@UGBK@          unicode HYShuSongErS.ttf}
  \special{pdf:mapline gbksong@UGBK@      unicode HYShuSongErS.ttf}
  \special{pdf:mapline gbkkai@UGBK@       unicode HYKaiTiS.ttf}
  \special{pdf:mapline gbkhei@UGBK@       unicode HYZhongHeiS.ttf}
  \special{pdf:mapline gbkfs@UGBK@        unicode HYFangSongS.ttf}
  \special{pdf:mapline cyberb@Unicode@    unicode HYShuSongErS.ttf}
  \special{pdf:mapline unisong@Unicode@   unicode HYShuSongErS.ttf}
  \special{pdf:mapline unikai@Unicode@    unicode HYKaiTiS.ttf}
  \special{pdf:mapline unihei@Unicode@    unicode HYZhongHeiS.ttf}
  \special{pdf:mapline unifs@Unicode@     unicode HYFangSongS.ttf}
  \special{pdf:mapline gbksongsl@UGBK@    unicode HYShuSongErS.ttf -s .167}
  \special{pdf:mapline gbkkaisl@UGBK@     unicode HYKaiTiS.ttf     -s .167}
  \special{pdf:mapline gbkheisl@UGBK@     unicode HYZhongHeiS.ttf  -s .167}
  \special{pdf:mapline gbkfssl@UGBK@      unicode HYFangSongS.ttf  -s .167}
  \special{pdf:mapline unisongsl@Unicode@ unicode HYShuSongErS.ttf -s .167}
  \special{pdf:mapline unikaisl@Unicode@  unicode HYKaiTiS.ttf     -s .167}
  \special{pdf:mapline uniheisl@Unicode@  unicode HYZhongHeiS.ttf  -s .167}
  \special{pdf:mapline unifssl@Unicode@   unicode HYFangSongS.ttf  -s .167}
\fi
⟨/hanyi⟩
```

15.15.4.5 ctex-zhmap-mac.tex

```
⟨*mac⟩
\ifzhmappdf
%% pdfTeX does not support OTF fonts
\else
  \special{pdf:mapline gbk@UGBK@          UniGB-UTF16-H :3:Songti.ttc}
  \special{pdf:mapline gbksong@UGBK@      UniGB-UTF16-H :3:Songti.ttc}
```

```
  \special{pdf:mapline gbkkai@UGBK@       UniGB-UTF16-H :0:Kaiti.ttc}
  \special{pdf:mapline gbkhei@UGBK@       unicode       :2:PingFang.ttc}
  \special{pdf:mapline gbkfs@UGBK@        unicode       STFANGSO.ttf}
  \special{pdf:mapline gbkli@UGBK@        UniGB-UTF16-H :0:Baoli.ttc}
  \special{pdf:mapline gbkyou@UGBK@       UniGB-UTF16-H :4:Yuanti.ttc}
  \special{pdf:mapline cyberb@Unicode@    UniGB-UTF16-H :3:Songti.ttc}
  \special{pdf:mapline unisong@Unicode@   UniGB-UTF16-H :3:Songti.ttc}
  \special{pdf:mapline unikai@Unicode@    UniGB-UTF16-H :0:Kaiti.ttc}
  \special{pdf:mapline unihei@Unicode@    unicode       :2:PingFang.ttc}
  \special{pdf:mapline unifs@Unicode@     unicode       STFANGSO.ttf}
  \special{pdf:mapline unili@Unicode@     UniGB-UTF16-H :0:Baoli.ttc}
  \special{pdf:mapline uniyou@Unicode@    UniGB-UTF16-H :4:Yuanti.ttc}
  \special{pdf:mapline gbksongsl@UGBK@    UniGB-UTF16-H :3:Songti.ttc   -s .167}
  \special{pdf:mapline gbkkaisl@UGBK@     UniGB-UTF16-H :0:Kaiti.ttc    -s .167}
  \special{pdf:mapline gbkheisl@UGBK@     unicode       :2:PingFang.ttc -s .167}
  \special{pdf:mapline gbkfssl@UGBK@      unicode       STFANGSO.ttf    -s .167}
  \special{pdf:mapline gbklisl@UGBK@      UniGB-UTF16-H :0:Baoli.ttc    -s .167}
  \special{pdf:mapline gbkyousl@UGBK@     UniGB-UTF16-H :4:Yuanti.ttc   -s .167}
  \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H :3:Songti.ttc   -s .167}
  \special{pdf:mapline unikaisl@Unicode@  UniGB-UTF16-H :0:Kaiti.ttc    -s .167}
  \special{pdf:mapline uniheisl@Unicode@  unicode       :2:PingFang.ttc -s .167}
  \special{pdf:mapline unifssl@Unicode@   unicode       STFANGSO.ttf    -s .167}
  \special{pdf:mapline unilisl@Unicode@   UniGB-UTF16-H :0:Baoli.ttc    -s .167}
  \special{pdf:mapline uniyousl@Unicode@  UniGB-UTF16-H :4:Yuanti.ttc   -s .167}
\fi
</mac>
```

15.15.4.6 ctex-zhmap-ubuntu.tex

```
<*ubuntu>
\ifzhmappdf
%% pdfTeX does not support OTF fonts
\else
  \special{pdf:mapline gbk@UGBK@          unicode :2:NotoSerifCJK-Regular.ttc}
  \special{pdf:mapline gbksong@UGBK@      unicode :2:NotoSerifCJK-Regular.ttc}
  \special{pdf:mapline gbkkai@UGBK@       unicode gkai00mp.ttf}
  \special{pdf:mapline gbkhei@UGBK@       unicode :2:NotoSansCJK-Regular.ttc}
  \special{pdf:mapline gbkfs@UGBK@        unicode :2:NotoSerifCJK-Regular.ttc}
  \special{pdf:mapline cyberb@Unicode@    unicode :2:NotoSerifCJK-Regular.ttc}
  \special{pdf:mapline unisong@Unicode@   unicode :2:NotoSerifCJK-Regular.ttc}
  \special{pdf:mapline unikai@Unicode@    unicode gkai00mp.ttf}
  \special{pdf:mapline unihei@Unicode@    unicode :2:NotoSansCJK-Regular.ttc}
  \special{pdf:mapline unifs@Unicode@     unicode :2:NotoSerifCJK-Regular.ttc}
  \special{pdf:mapline gbksongsl@UGBK@    unicode :2:NotoSerifCJK-Regular.ttc -s .167}
  \special{pdf:mapline gbkkaisl@UGBK@     unicode gkai00mp.ttf               -s .167}
  \special{pdf:mapline gbkheisl@UGBK@     unicode :2:NotoSansCJK-Regular.ttc  -s .167}
  \special{pdf:mapline gbkfssl@UGBK@      unicode :2:NotoSerifCJK-Regular.ttc -s .167}
  \special{pdf:mapline unisongsl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
  \special{pdf:mapline unikaisl@Unicode@  unicode gkai00mp.ttf               -s .167}
  \special{pdf:mapline uniheisl@Unicode@  unicode :2:NotoSansCJK-Regular.ttc  -s .167}
  \special{pdf:mapline unifssl@Unicode@   unicode :2:NotoSerifCJK-Regular.ttc -s .167}
\fi
</ubuntu>
```

15.15.4.7 ctex-zhmap-windows.tex

```
<*windows>
\ifzhmappdf
  \pdfmapline{=gbk@UGBK@          <simsun.ttc}
  \pdfmapline{=gbksong@UGBK@      <simsun.ttc}
  \pdfmapline{=gbkkai@UGBK@       <simkai.ttf}
  \pdfmapline{=gbkhei@UGBK@       <simhei.ttf}
  \pdfmapline{=gbkfs@UGBK@        <simfang.ttf}
  \pdfmapline{=gbkli@UGBK@        <simli.ttf}
  \pdfmapline{=gbkyou@UGBK@       <simyou.ttf}
  \pdfmapline{=cyberb@Unicode@    <simsun.ttc}
  \pdfmapline{=unisong@Unicode@   <simsun.ttc}
  \pdfmapline{=unikai@Unicode@    <simkai.ttf}
```

```
  \pdfmapline{=unihei@Unicode@    <simhei.ttf}
  \pdfmapline{=unifs@Unicode@     <simfang.ttf}
  \pdfmapline{=unili@Unicode@     <simli.ttf}
  \pdfmapline{=uniyou@Unicode@    <simyou.ttf}
  \pdfmapline{=gbksongsl@UGBK@    <simsun.ttc}
  \pdfmapline{=gbkkaisl@UGBK@     <simkai.ttf}
  \pdfmapline{=gbkheisl@UGBK@     <simhei.ttf}
  \pdfmapline{=gbkfssl@UGBK@      <simfang.ttf}
  \pdfmapline{=gbklisl@UGBK@      <simli.ttf}
  \pdfmapline{=gbkyousl@UGBK@     <simyou.ttf}
  \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
  \pdfmapline{=unikaisl@Unicode@  <simkai.ttf}
  \pdfmapline{=uniheisl@Unicode@  <simhei.ttf}
  \pdfmapline{=unifssl@Unicode@   <simfang.ttf}
  \pdfmapline{=unilisl@Unicode@   <simli.ttf}
  \pdfmapline{=uniyousl@Unicode@  <simyou.ttf}
\else
  \special{pdf:mapline gbk@UGBK@          unicode :0:simsun.ttc -v 50}
  \special{pdf:mapline gbksong@UGBK@      unicode :0:simsun.ttc -v 50}
  \special{pdf:mapline gbkkai@UGBK@       unicode simkai.ttf    -v 70}
  \special{pdf:mapline gbkhei@UGBK@       unicode simhei.ttf    -v 150}
  \special{pdf:mapline gbkfs@UGBK@        unicode simfang.ttf   -v 50}
  \special{pdf:mapline gbkli@UGBK@        unicode simli.ttf     -v 150}
  \special{pdf:mapline gbkyou@UGBK@       unicode simyou.ttf    -v 60}
  \special{pdf:mapline cyberb@Unicode@    unicode :0:simsun.ttc -v 50}
  \special{pdf:mapline unisong@Unicode@   unicode :0:simsun.ttc -v 50}
  \special{pdf:mapline unikai@Unicode@    unicode simkai.ttf    -v 70}
  \special{pdf:mapline unihei@Unicode@    unicode simhei.ttf    -v 150}
  \special{pdf:mapline unifs@Unicode@     unicode simfang.ttf   -v 50}
  \special{pdf:mapline unili@Unicode@     unicode simli.ttf     -v 150}
  \special{pdf:mapline uniyou@Unicode@    unicode simyou.ttf    -v 60}
  \special{pdf:mapline gbksongsl@UGBK@    unicode :0:simsun.ttc -v 50  -s .167}
  \special{pdf:mapline gbkkaisl@UGBK@     unicode simkai.ttf    -v 70  -s .167}
  \special{pdf:mapline gbkheisl@UGBK@     unicode simhei.ttf    -v 150 -s .167}
  \special{pdf:mapline gbkfssl@UGBK@      unicode simfang.ttf   -v 50  -s .167}
  \special{pdf:mapline gbklisl@UGBK@      unicode simli.ttf     -v 150 -s .167}
  \special{pdf:mapline gbkyousl@UGBK@     unicode simyou.ttf    -v 60  -s .167}
  \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -v 50  -s .167}
  \special{pdf:mapline unikaisl@Unicode@  unicode simkai.ttf    -v 70  -s .167}
  \special{pdf:mapline uniheisl@Unicode@  unicode simhei.ttf    -v 150 -s .167}
  \special{pdf:mapline unifssl@Unicode@   unicode simfang.ttf   -v 50  -s .167}
  \special{pdf:mapline unilisl@Unicode@   unicode simli.ttf     -v 150 -s .167}
  \special{pdf:mapline uniyousl@Unicode@  unicode simyou.ttf    -v 60  -s .167}
\fi
⟨/windows⟩

\ctex@zhmap@endinput
⟨/zhmap⟩
```

15.15.5　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。

```
⟨*spa⟩
⟨*macro⟩
\input expl3-generic %
\ExplSyntaxOn
\sys_if_engine_xetex:F
  {
    \msg_new:nnn { ctex } { xetex }
      { XeTeX~is~required~to~compile~this~document! }
    \msg_fatal:nn { ctex } { xetex }
  }
```

CJKpunct 定义的标点符号是:

‘ “ 「 『 〔 （ ［ ｛ 〈 《 〖 【
—…、。，．：；！？%）〕 ］ ｝ 〉 》 〗 】 ’ ” 」 』

注意顺序不能改变。

```
\seq_const_from_clist:Nn \c__ctex_punct_seq
  {
    "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
    "3008 , "300A , "3016 , "3010 ,
    "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
    "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
    "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
  }
```

\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。

```
\cs_new_protected:Npn \ctex_make_spa:nn #1#2
  {
    \iow_open:Nn \g__ctex_spa_iow {#1}
    \clist_map_inline:nn {#2}
      { \__ctex_write_family:nn ##1 }
    \iow_close:N \g__ctex_spa_iow
  }
\iow_new:N \g__ctex_spa_iow
\cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

\cs_new_protected:Npn \__ctex_write_family:nn #1#2
  {
    \group_begin:
      \__ctex_if_font_exists:nTF {#2}
        { \tex_font:D \l__ctex_punct_font = "[#2]" ~ at ~ 100 pt \scan_stop: }
        { \tex_font:D \l__ctex_punct_font =  "#2"  ~ at ~ 100 pt \scan_stop: }
      \l__ctex_punct_font
      \clist_clear:N \l__ctex_punct_bounds_clist
      \seq_map_inline:Nn \c__ctex_punct_seq
        {
          \exp_args:No \__ctex_save_bounds:n
            { \int_use:N \tex_XeTeXcharglyph:D ##1 }
        }
      \iow_now:Ne \g__ctex_spa_iow
        {
          \token_to_str:N \ctexspadef {#1}
```

最后这三个逗号对 CJKpunct 来说是必要的。

```
            { \l__ctex_punct_bounds_clist , , , }
        }
    \group_end:
  }
```

__ctex_if_font_exists:nTF　优先搜索字体是否为 TEX Tree 字体或是否存在于当前路径下[21]，如果不存在，__ctex_write_family:nn 会根据结果在 shell 中调用 kpathsea 查文件或 fontconfig 查名称[22]。

```
\prg_new_protected_conditional:Nnn \__ctex_if_font_exists:n { TF }
  {
    \tl_clear:N \l__ctex_if_otf_exists_tl
    \tl_clear:N \l__ctex_if_ttf_exists_tl
    \bool_case:nF
      {
        { \file_if_exist_p:n { #1.otf }          }
        { \tl_set:Nn \l__ctex_if_otf_exists_tl {#1} }
        { \file_if_exist_p:n { #1.ttf }          }
```

[21] 事实上 TEXINPUTS 也会被一并优先搜索。

[22] 如果用户把字体目录加入到了 TEXINPUTS 中, \file_if_exist_p:n 可能找到一个 XETEX [basename] 加载不了的文件, 因为 XETEX 搜索字体专属路径 OPENTYPEFONTS/TTFONTS, 不搜索 TEXINPUTS)。

```
        { \tl_set:Nn \l__ctex_if_ttf_exists_tl {#1} }
      }
      {
        \sys_get_shell:nnN { kpsewhich ~ #1.otf } { \endlinechar=-1 }
          \l__ctex_if_otf_exists_tl
        \sys_get_shell:nnN { kpsewhich ~ #1.ttf } { \endlinechar=-1 }
          \l__ctex_if_ttf_exists_tl
      }
    \bool_lazy_and:nnTF
      { \tl_if_blank_p:V \l__ctex_if_otf_exists_tl }
      { \tl_if_blank_p:V \l__ctex_if_ttf_exists_tl }
      { \prg_return_false: } { \prg_return_true: }
  }
```

\l__ctex_if_otf_exists_tl
\l__ctex_if_ttf_exists_tl
用于存储 shell 输出的 kpsewhich 结果。

```
\tl_clear_new:N \l__ctex_if_otf_exists_tl
\tl_clear_new:N \l__ctex_if_ttf_exists_tl

\cs_new_protected:Npn \__ctex_save_bounds:n #1
  {
    \clist_put_right:Ne \l__ctex_punct_bounds_clist
      {
        \__ctex_calc_bounds:nn { 1 } {#1} ,
        \__ctex_calc_bounds:nn { 3 } {#1}
      }
  }
\clist_new:N \l__ctex_punct_bounds_clist
```

CJKpunct 要求的格式是边界空白宽度与 1 em 的比值的一百倍。

```
\cs_new:Npn \__ctex_calc_bounds:nn #1#2
  {
    \fp_eval:n
      {
        round
          (
            \dim_to_decimal_in_unit:nn
              { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
              { 1 em }
          )
      }
  }
\ExplSyntaxOff
```

〈/macro〉

下面是 CTEX 定义的一些字体。

```
〈*make〉
\input ctex-spa-macro %
\MAKESPA {ctexpunct.spa}
  {
    {adobezhsong}     {AdobeSongStd-Light} ,
    {adobezhhei}      {AdobeHeitiStd-Regular} ,
    {adobezhkai}      {AdobeKaitiStd-Regular} ,
    {adobezhfs}       {AdobeFangsongStd-Regular} ,
%
    {fandolzhsong}    {FandolSong-Regular} ,
    {fandolzhsongb}   {FandolSong-Bold} ,
    {fandolzhhei}     {FandolHei-Regular} ,
    {fandolzhheib}    {FandolHei-Bold} ,
    {fandolzhkai}     {FandolKai-Regular} ,
    {fandolzhfs}      {FandolFang-Regular} ,
%
    {founderzhsong}   {FZShuSong-Z01} ,
    {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
    {founderzhhei}    {FZHei-B01} ,
```

```
    {founderzhheil}   {FZXiHeiI-Z08} ,
    {founderzhkai}    {FZKai-Z03} ,
    {founderzhfs}     {FZFangSong-Z02} ,
    {founderzhli}     {FZLiShu-S01} ,
    {founderzhyou}    {FZXiYuan-M01} ,
    {founderzhyoub}   {FZZhunYuan-M02} ,
%
    {hanyizhsong}     {HYShuSongEr S} ,
    {hanyizhsongb}    {HYZhongSong S} ,
    {hanyizhhei}      {HYZhongHei S} ,
    {hanyizhheib}     {HYDaHei S} ,
    {hanyizhkai}      {HYKaiTi S} ,
    {hanyizhfs}       {HYFangSong S} ,
%
    {maczhsong}       {Songti SC Light} ,
    {maczhsongb}      {Songti SC Bold} ,
    {maczhhei}        {Heiti SC Medium} ,
    {maczhheil}       {Heiti SC Light} ,
    {maczhkai}        {Kaiti SC} ,
    {maczhkaib}       {Kaiti SC Bold} ,
    {maczhfs}         {STFangsong} ,
    {maczhli}         {Baoli SC} ,
    {maczhyou}        {Yuanti SC Light} ,
    {maczhyoub}       {Yuanti SC Regular} ,
    {maczhpf}         {PingFang SC} ,
    {maczhpfb}        {PingFang SC Semibold} ,
%
    {ubuntuzhsong}    {Noto Serif CJK SC} ,
    {ubuntuzhsongb}   {Noto Serif CJK SC Bold} ,
    {ubuntuzhhei}     {Noto Sans CJK SC} ,
    {ubuntuzhheib}    {Noto Sans CJK SC Bold} ,
    {ubuntuzhkai}     {AR PL KaitiM GB} ,
%
    {windowszhsong}   {SimSun} ,
    {windowszhhei}    {SimHei} ,
    {windowszhkai}    {KaiTi} ,
    {windowszhfs}     {FangSong} ,
    {windowszhli}     {LiSu} ,
    {windowszhyou}    {YouYuan} ,
    {windowszhyahei}  {Microsoft YaHei} ,
    {windowszhyaheib} {Microsoft YaHei Bold}
  }
\primitive\end
⟨/make⟩
⟨/spa⟩
```

15.16　translator 宏包的中文字典

⟨*dict⟩

包括 ChineseGBK 和 ChineseUTF8 两种形式, 目前只翻译 beamer 宏包需要的定理环境名称。

```
⟨*theorem⟩
\providetranslation{Comments}{评论}
\providetranslation{comments}{评论}
\providetranslation{Comment}{评论}
\providetranslation{comment}{评论}
\providetranslation{Corollaries}{推论}
\providetranslation{corollaries}{推论}
\providetranslation{Corollary}{推论}
\providetranslation{corollary}{推论}
\providetranslation{Definitions}{定义}
\providetranslation{definitions}{定义}
\providetranslation{Definition}{定义}
\providetranslation{definition}{定义}
```

```
\providetranslation{Examples}{例}
\providetranslation{examples}{例}
\providetranslation{Example}{例}
\providetranslation{example}{例}
\providetranslation{Exercises}{练习}
\providetranslation{exercises}{练习}
\providetranslation{Exercise}{练习}
\providetranslation{exercise}{练习}
\providetranslation{Facts}{事实}
\providetranslation{facts}{事实}
\providetranslation{Fact}{事实}
\providetranslation{fact}{事实}
\providetranslation{Key Lemmas}{关键引理}
\providetranslation{key lemmas}{关键引理}
\providetranslation{Key Lemma}{关键引理}
\providetranslation{key lemma}{关键引理}
\providetranslation{Key Observations}{关键观察}
\providetranslation{key observations}{关键观察}
\providetranslation{Key Observation}{关键观察}
\providetranslation{key observation}{关键观察}
\providetranslation{Lemmas}{引理}
\providetranslation{lemmas}{引理}
\providetranslation{Lemma}{引理}
\providetranslation{lemma}{引理}
\providetranslation{Main Theorems}{主要定理}
\providetranslation{main theorems}{主要定理}
\providetranslation{Main Theorem}{主要定理}
\providetranslation{main theorem}{主要定理}
\providetranslation{Observations}{观察}
\providetranslation{observations}{观察}
\providetranslation{Observation}{观察}
\providetranslation{observation}{观察}
\providetranslation{Problems}{问题}
\providetranslation{problems}{问题}
\providetranslation{Problem}{问题}
\providetranslation{problem}{问题}
\providetranslation{Proofs}{证明}
\providetranslation{proofs}{证明}
\providetranslation{Proof}{证明}
\providetranslation{proof}{证明}
\providetranslation{Proof Sketch}{证明提要}
\providetranslation{Proof sketch}{证明提要}
\providetranslation{proof sketch}{证明提要}
\providetranslation{Proof Sketches}{证明提要}
\providetranslation{Proof sketches}{证明提要}
\providetranslation{proof sketches}{证明提要}
\providetranslation{Sketch of Proof}{证明提要}
\providetranslation{Sketch of Proofs}{证明提要}
\providetranslation{Sketch of proof}{证明提要}
\providetranslation{Sketch of proofs}{证明提要}
\providetranslation{sketch of proof}{证明提要}
\providetranslation{sketch of proofs}{证明提要}
\providetranslation{Propositions}{命题}
\providetranslation{propositions}{命题}
\providetranslation{Proposition}{命题}
\providetranslation{proposition}{命题}
\providetranslation{Remarks}{注}
\providetranslation{remarks}{注}
\providetranslation{Remark}{注}
\providetranslation{remark}{注}
\providetranslation{Solutions}{解}
\providetranslation{solutions}{解}
\providetranslation{Solution}{解}
\providetranslation{solution}{解}
\providetranslation{Theorems}{定理}
\providetranslation{theorems}{定理}
\providetranslation{Theorem}{定理}
```

```
\providetranslation{theorem}{定理}
</theorem>
</dict>
```

15.17　ctexcap 宏包

```
<*ctexcap>
```

ctexcap 是过时宏包。

```
\clist_new:N \l__ctex_ctexcap_options_clist
\clist_set:Ne \l__ctex_ctexcap_options_clist
  { \exp_not:v { opt@ \@currname . \@currext } , heading }
\msg_new:nnn { ctexcap } { deprecated }
  {
    Package~`ctexcap'~is~deprecated.\\
    Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
    \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
  }
\msg_warning:nne { ctexcap } { deprecated }
  { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }
```

ctexcap 是默认打开 heading 选项的 ctex。

```
\PassOptionsToPackage { heading = true } { ctexcap }
\RequirePackageWithOptions { ctex }
```

```
</ctexcap>
```

15.18　ctexhook 宏包

```
<*ctexhook>
```

\ctex_if_format_at_least:nTF　与 \IfFormatAtLeastTF 同义。

```
\cs_new:Npn \ctex_if_format_at_least:nTF
  { \@ifl@t@r \fmtversion }
```

\ctex_file_input:n
\ctex_push_file:
\ctex_pop_file:

输入文件, 关闭 LaTeX3 语法环境, 并设置 @ 为字母类, 利用 l3cctab 实现。我们使用 \file_input:n 而不是 LaTeX 2_ε 的 \input 或者 \InputIfFileExists 载入文件, 因此 LaTeX 2_ε 的文件钩子都无效。

```
\cs_new_protected:Npn \ctex_file_input:n #1
  {
    \ctex_push_file:
      \file_input:n {#1}
    \ctex_pop_file:
  }
\bool_if_exist:NTF \l__kernel_expl_bool
  {
    \cs_new_protected:Npn \ctex_push_file:
      {
        \seq_gpush:Nx \g__ctex_expl_status_seq
          { \bool_if:NTF \l__kernel_expl_bool { 1 } { 0 } }
        \bool_set_false:N \l__kernel_expl_bool
        \cctab_begin:N \c__ctex_package_cctab
      }
    \cs_new_protected:Npn \ctex_pop_file:
      {
        \cctab_end:
        \seq_gpop:NN \g__ctex_expl_status_seq \l__ctex_expl_status_tl
        \int_if_odd:nTF { \l__ctex_expl_status_tl }
          { \bool_set_true:N  \l__kernel_expl_bool }
          { \bool_set_false:N \l__kernel_expl_bool }
      }
```

```
    \tl_new:N \l__ctex_expl_status_tl
    \seq_new:N \g__ctex_expl_status_seq
  }
  {
    \cs_new_protected:Npn \ctex_push_file:
      { \cctab_begin:N \c__ctex_package_cctab }
    \cs_new_protected:Npn \ctex_pop_file:
      { \cctab_end: }
  }
\cctab_const:Nn \c__ctex_package_cctab
  {
    \cctab_select:N \c_document_cctab
    \char_set_catcode_letter:n { 64 }
  }
```

\ctex_disable_package:n 禁止宏包载入。采用 LaTeX 2020-10-01 提供的 \disable@package@load 实现，否则采用传统方式：预定义 \ver@⟨*package*⟩.sty 标识符。

```
\cs_new_protected:Npn \ctex_disable_package:n #1
  {
    \@ifpackageloaded {#1}
      { \msg_error:nnee }
      { \__ctex_disable_package_aux:nnnn }
      { ctexhook } { disable-package } {#1} { \@currname }
  }
\cs_new_protected:Npn \__ctex_disable_package_aux:nnnn #1#2#3#4
  {
    \cs_if_exist:NTF \disable@package@load
      {
        \exp_args:Nne \disable@package@load {#3}
          { \msg_warning:nnnn {#1} {#2} {#3} {#4} }
      }
      { \tl_const:cn { ver@ #3 . \@pkgextension } { 9999/99/99 } }
  }
\msg_new:nnn { ctexhook } { disable-package }
  { Package~`#1'~can~not~be~loaded~with~`#2'. }
```

\ctex_replace_package:nn 替换宏包。采用 LaTeX 2020-10-01 提供的 \declare@file@substitution 实现，否则给出无效警告。

```
\ctex_if_format_at_least:nTF { 2020/10/01 }
  {
    \cs_new_protected:Npn \ctex_replace_package:nn #1#2
      {
        \declare@file@substitution
          { #1 . \@pkgextension }
          { #2 . \@pkgextension }
      }
  }
  {
    \cs_new_protected:Npn \ctex_replace_package:nn
      { \msg_warning:nnnn { ctexhook } { replace-package-invalid } }
    \msg_new:nnn { ctexhook } { replace-package-invalid }
      {
        \token_to_str:N \ctex_replace_package:nn \{#1\}\{#2\}~is~invalid~
        before~LaTeX~2020-10-01.
      }
  }
```

\ctex_at_begin_package:nn 如果宏包已经被载入，则钩子无效，给出警告。

```
\cs_new_protected:Npn \ctex_at_begin_package:nn #1
  {
    \@ifpackageloaded {#1}
      { \__ctex_package_loaded_warning:nn {#1} }
```

```
      { \ctex_gadd_package_hook:nnn { before } {#1} }
  }
\cs_new_protected:Npn \__ctex_package_loaded_warning:nn #1#2
  { \msg_warning:nne { ctexhook } { invalid-hook } {#1} }
\msg_new:nnn { ctexhook } { invalid-hook }
  {
    Package~`#1'~is~loaded. \\
    \token_to_str:N \ctex_at_begin_package:nn \{#1\}\{...\}~is~invalid.
  }
```

\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。

```
\cs_new_protected:Npn \ctex_at_end_package:nn #1
  {
    \@ifpackageloaded {#1}
      { \use:n }
      { \ctex_gadd_package_hook:nnn { after } {#1} }
  }
```

LaTeX 2020/10/01 开始提供常用钩子管理机制。在新机制下, 我们只需要做简单的包装。

```
\ctex_if_format_at_least:nTF { 2020/10/01 }
  {
    \cs_new_protected:Npx \ctex_gadd_ltxhook:nn #1
      { \hook_gput_code:nnn {#1} { \c_novalue_tl } }
    \cs_new_protected:Npn \ctex_at_end_preamble:n
      { \ctex_gadd_ltxhook:nn { begindocument/before } }
    \cs_new_protected:Npn \ctex_after_end_preamble:n
      { \ctex_gadd_ltxhook:nn { begindocument/end } }
    \cs_new_protected:Npx \ctex_gadd_package_hook:nnn #1#2
      {
        \ctex_if_format_at_least:nTF { 2021/11/15 }
          { \ctex_gadd_ltxhook:nn { package/#2/#1 } }
          { \ctex_gadd_ltxhook:nn { package/#1/#2 } }
      }
    \file_input_stop:
  }
  { }
```

对于 LaTeX 2020/10/01 之前的版本, 需要自行补丁。

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

```
\cs_new_protected:Npn \ctex_at_end_preamble:n
  { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl }
\cs_new_protected:Npn \ctex_after_end_preamble:n
  { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl }
\cs_new_protected:Npn \CTEX@document@left@hook
  { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
\cs_new_protected:Npn \CTEX@document@right@hook
  { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
\cs_set_nopar:Npx \document
  {
    \CTEX@document@left@hook
    \exp_not:o { \document }
    \CTEX@document@right@hook
  }
\tl_new:N \g__ctex_end_preamble_hook_tl
\tl_new:N \g__ctex_after_end_preamble_hook_tl
```

\ctex_gadd_package_hook:nnn
\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn
给钩子附加内容。

```
\cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
  { \ctex_gadd_hook:cn { g__ctex_at_ #1 _ #2 _hook_tl } }
```

```
\cs_new_protected:Npn \ctex_gadd_hook:Nn #1
  {
    \tl_if_exist:NF #1 { \tl_new:N #1 }
    \tl_gput_right:Nn #1
  }
\cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }
```

`\ctex_use_package_hook:nn` 宏包钩子, 只执行一次, 用后清除。

```
\cs_new_protected:Npn \ctex_use_package_hook:nn #1#2
  {
    \group_begin: \exp_args:NNc \group_end:
    \__ctex_use_package_hook_aux:N { g__ctex_at_ #1 _ #2 _hook_tl }
  }
\cs_new_protected:Npn \__ctex_use_package_hook_aux:N #1
  { \cs_if_exist_use:NT #1 { \cs_undefine:N #1 } }
```

`\@reset@ptions` `\CTEX@reset@ptions@hook` `\@pushfilename` 内部的 `\@currname` 和 `\@currext` 保存的是前一个宏包的状态, 不能使用。需要对其后的 `\@reset@ptions` 做补丁来实现 `\ctex_at_begin_package:nn` 的功能。

```
\tl_put_right:Nn \@reset@ptions { \CTEX@reset@ptions@hook }
\cs_new_protected:Npn \CTEX@reset@ptions@hook
  {
    \cs_if_eq:NNT \@currext \@pkgextension
      { \ctex_use_package_hook:nn { before } { \@currname } }
  }
```

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

```
\tl_put_left:Nn \@popfilename { \CTEX@popfilename@hook }
\cs_new_protected:Npn \CTEX@popfilename@hook
  {
    \cs_if_eq:NNT \@currext \@pkgextension
      { \ctex_use_package_hook:nn { after } { \@currname } }
  }
```

```
⟨/ctexhook⟩
```

15.19　ctexpatch 宏包

```
⟨*ctexpatch⟩
```

`\ctex_patch_cmd_once:NnnnTF` 只进行第一次匹配进行替换。参数 #2 是宏重建时的 `\catcode` 设置。

```
\cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
  {
    \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnnw #1
      { once } {#2} { \use_i:nn } { \use_ii:nn }
  }
```

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

```
\cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
  {
    \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnnw #1
      { all } {#2} { \use_i:nn } { \use_ii:nn }
  }
```

`\ctex_patch_cmd:Nnn` 快捷方式, 在补丁的时候关闭 LaTeX3 语法和设置 `@` 为字母类, 补丁失败时给出警告。

```
\cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
  {
    \ctex_patch_boot:NNnnTF \__ctex_patch_cmd:Nnnnnw #1
```

```
      { once }
      {
        \ExplSyntaxOff
        \char_set_catcode_letter:n { 64 }
      }
      { }
      { \ctex_patch_failure:N #1 }
  }
\cs_new_protected:Npn \ctex_patch_failure:N #1
  { \msg_warning:nne { ctexpatch } { patch-failure } { \token_to_str:N #1 } }
\msg_new:nnn { ctexpatch } { patch-failure }
  { Oops!~Command~`#1'~is~NOT~patchable.\\ }
```

\ctex_preto_cmd:NnnTF 在宏的原本定义前面增加钩子。

```
\cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
  {
    \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
      { left } {#2} { \use_i:nn } { \use_ii:nn }
  }
```

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```
\cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
  {
    \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
      { right } {#2} { \use_i:nn } { \use_ii:nn }
  }
```

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

```
\cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
  {
    \cs_set_protected:Npx \__ctex_patch_true:w  { \exp_not:n {#5} }
    \cs_set_protected:Npx \__ctex_patch_false:w { \exp_not:n {#6} }
    \group_begin:
      \char_set_catcode_other:n { 35 }
      \ctex_parse_name:NN #1 #2 {#3} {#4}
  }
\cs_new_eq:NN \__ctex_patch_true:w  \use_i:nn
\cs_new_eq:NN \__ctex_patch_false:w \use_ii:nn
```

\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):

```
\protect␣\xaa␣␣                    % \DeclareRobustCommand\xaa[1]{...}
\protect␣\xab␣␣                    % \DeclareRobustCommand\xab[1][]{...}
\@protected@testopt␣\xac␣\\xac␣{}  % \newcommand\xac[1][]{...}
\@testopt␣\\xad␣{}                 % \newrobustcmd\xad[1][]{...}
\x@protect␣\1\protect␣\1␣␣         % \DeclareRobustCommand\1[1]{...}
\x@protect␣\2\protect␣\2␣␣         % \DeclareRobustCommand\2[1][]{...}
\@protected@testopt␣\3\\3␣{}       % \newcommand\3[1][]{...}
\@testopt␣\\4␣{}                   % \newrobustcmd\4[1][]{...}
```

ctexpatch 的主要原理是先对宏的 \meaning 作字符串替换, 然后再用 \scantokens 来重建它。我们希望对宏的实际定义打补丁, 为此需要先得到对应的名字。letltxmacro、show2e 和 xpatch 宏包中都有类似的工作。

```
\cs_new_protected:Npn \ctex_parse_name:NN #1#2
  { \ctex_parse_name:NNe #1#2 { \cs_to_str:N #2 } }
\group_begin:
\cs_set_protected:Npn \__ctex_tmp:w #1#2#3
```

```
  {
    \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
      {
        \bool_lazy_or:nnTF
          { \cs_if_exist_p:c { ##3 ~ } }
          { \cs_if_exist_p:c { #1##3 } }
          {
            \group_begin:
            \use:e
              {
                \group_end:
                \__ctex_parse_name:nNNNnN
                  { \cs_replacement_spec:N ##2 }
                  \exp_not:N ##2
                  \exp_not:c { ##3 ~ }
                  \exp_not:c { #1##3 }
              } { ##3 } ##1
          }
          { ##1##2 }
      }
    \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
      {
        \exp_args:Nc ##6
          {
            \str_case:nnTF {##1}
              {
                { \protect ##3 } { }
                { \x@protect ##2 \protect ##3 } { }
              }
              {
                \str_if_eq:eeTF
                  { \exp_not:n { #1@protected@ ##3 #1##3 } }
                  {
                    \exp_last_unbraced:Ne \__ctex_parse_name:w
                      { \cs_replacement_spec:N ##3 } #3 ~ #2 \q_stop
                  }
                  { #1##5 ~ } { ##5 ~ }
              }
              {
                \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
                  {
                    { #1@protected@ ##2 ##4 } { }
                    { #1@ ##4 } { }
                  }
                  { #1##5 } {##5}
              }
          }
      }
    \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
  }
\use:e
  {
    \__ctex_tmp:w
      { \c_backslash_str }
      { \c_left_brace_str }
      { \tl_to_str:n { testopt } }
  }
\group_end:
\cs_generate_variant:Nn \ctex_parse_name:NNn { NNe }
```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

分别保存宏的 \meaning 中的前缀、参数文本和替换文本。

```
\str_new:N \l__ctex_prefix_str
\str_new:N \l__ctex_parameter_str
\str_new:N \l__ctex_replacement_str
```

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

```
\group_begin:
  \cs_set_protected:Npn \__ctex_tmp:w #1
    {
      \prg_new_protected_conditional:Npnn
        \ctex_get_macro_meaning:N ##1 { TF }
        {
          \exp_after:wN \__ctex_get_macro_meaning:w
            \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
        }
      \cs_new_protected:Npn \__ctex_get_macro_meaning:w
          ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
        {
          \tl_if_empty:nTF { ##4 }
            { \prg_return_false: }
            {
              \str_set:Nn \l__ctex_prefix_str      { ##1 }
              \str_set:Nn \l__ctex_parameter_str   { ##2 }
              \str_set:Nn \l__ctex_replacement_str { ##3 }
              \prg_return_true:
            }
        }
    }
  \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
\group_end:
```

\ctex_if_rescanable:NnTF
检查宏是否可以重建。

```
\cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
  {
    \ctex_get_macro_meaning:NTF #1
      {
        \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
        \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
      }
      {#4}
  }
\cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:
```

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

```
\cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
  {
    \__ctex_patch_rescan:NNn \l__ctex_prefix_tl      \l__ctex_prefix_str      {#2}
    \__ctex_patch_rescan:NNn \l__ctex_parameter_tl   \l__ctex_parameter_str   {#2}
    \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
    \use:e
      {
        \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
          \exp_not:o { \l__ctex_parameter_tl }
            { \exp_not:o { \l__ctex_replacement_tl } }
      }
  }
\cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
  {
    \str_if_empty:NTF #2
      { \tl_clear:N #1 }
      { \tl_set_rescan:Nno #1 {#3} {#2} }
  }
\tl_new:N \l__ctex_prefix_tl
\tl_new:N \l__ctex_parameter_tl
\tl_new:N \l__ctex_replacement_tl
```

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

```
\cs_new_protected:Npn \__ctex_patch_cmd:Nnnnnw #1#2#3#4#5
```

```
  {
    \group_end:
    \ctex_if_rescanable:NnTF #1 {#3}
      {
        \use:e
          {
            \__ctex_patch_replace:nnnTF {#2}
              { \tl_to_str:n {#4} }
              { \tl_to_str:n {#5} }
          }
          {
            \__ctex_patch_rebuild:Nn #1 {#3}
            \__ctex_patch_true:w
          }
          { \__ctex_patch_false:w }
      }
      { \__ctex_patch_false:w }
  }
```

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

```
\cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
  {
    \tl_if_in:NnTF \l__ctex_replacement_str {#2}
      { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
  }
```

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```
\cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
  {
    \group_end:
    \ctex_get_macro_meaning:NTF #1
      {
        \str_if_empty:NTF \l__ctex_parameter_str
          { \__ctex_hookto_cmd_parameterless:Nnnnw }
          { \__ctex_hookto_cmd_parameter:Nnnnw }
          #1 {#2} {#3} {#4}
      }
      { \__ctex_patch_false:w }
  }
```

__ctex_hookto_cmd_parameterless:Nnnnw 如果宏没有参数, 可以直接进行附加操作。注意保持宏的前缀。

```
\cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
  {
    \str_if_empty:NF \l__ctex_prefix_str
      { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
    \tex_edef:D #1
      {
        \use:c { __ctex_ #2 _hook_aux:nn }
          { \exp_not:o {#1} }
          { \exp_not:n {#4} }
      }
    \__ctex_patch_true:w
  }
\cs_generate_variant:Nn \tl_rescan:nn { no }
\cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
\cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn
```

__ctex_hookto_cmd_parameter:Nnnnw 如果宏有参数, 需要在字符串中进行附加, 然后再重建。

```
\cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
  {
    \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
    \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
```

```
      {
        \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
        \__ctex_patch_rebuild:Nn #1 {#3}
        \__ctex_patch_true:w
      }
      { \__ctex_patch_false:w }
  }
```

⟨/ctexpatch⟩

版本历史

v2.0 (2014/03/06 – 2015/05/06)

General: c5size, cs4size 是过时选项。 44
captiondelimiter 是过时选项。 130
fancyhdr 成为过时选项, 原选项功能总是打开。 47
fntef 成为过时选项, 原选项功能总是打开。 47
hyperref 成为过时选项, 原选项功能总是打开。 47
indent, noindent 是过时选项。 45
nofonts, adobefonts, winfonts 是过时选项。 46
nopunct 是过时选项。 46
nospace 是过时选项。 46
nozhmap 是过时选项。 46
punct 选项可以设置标点格式。 46
ctex 宏包新增 heading 选项。 46
\CTEXindent, \CTEXnoindent 是过时命令。 94
\CTEXsetup, \CTEXoptions 是过时命令。 16, 91
\CTEXunderdot、\CTEXunderline、\CTEXunderdblline、\CTEXunderwave、\CTEXsout、\CTEXxout 是过时命令; CTEXfilltwosides 是过时环境。 85
标题设置新增 pagestyle 选项。 98
调整 \footnotesep 的大小, 以适合行距的变化。 129
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。 48
将标题汉化功能加入 ctex.sty。 95
将中文字号功能提取到可以独立使用的 ctexsize。 125
解决 etoolbox 与 breqn 关于 \end 的冲突。 130
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。 85
删除 c19gbsn.fd 和 c19gkai.fd。 1
通过 LuaTeX-ja 宏包支持 LuaLaTeX。 58
新增 autoindent 选项。 45
新增 fontset 选项。 46
新增 linespread 选项。 45
新增 linestretch 选项。 92
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。 . . 47
新增 zhmCJK 支持选项。 46
新增 zihao 选项。 44
新增统一设置接口 \ctexset。 91
应用 LaTeX3 重新整理代码。 1
中文字号不再采用近似值。 125
自动检测操作系统, 载入对应的字体配置。 84
\CJK@surr: 解决与 \nouppercase 的冲突。 52

v2.0.1 (2015/05/15)

General: 修复 10pt、11pt 等选项无效的问题。 48

v2.0.2 (2015/05/16)

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。 125

v2.1 (2015/05/18 – 2015/06/19)

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。 97
nameformat 可以接受章节名字为参数。 96
不依赖 ifpdf 宏包。 43
不再设置 hyperref 宏包的 colorlinks 选项。 84
给 pdfLaTeX 下的非 UTF-8 编码 CJK 字体族加上 CMap。 50
将章节标题设置功能提取到可以独立使用的宏包 ctexheading 中。 1
新的标题格式选项 aftertitle。 97
修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。 119
\ctex_if_platform_macos:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。 131
hyperref: 补充定义 \hypersetup。 47

v2.2 (2015/06/21 – 2015/06/30)

General: beforeskip 和 afterskip 选项的符号不再有特殊意义。 97
beforeskip、afterskip 和 indent 选项支持表达式。 . 100
不再依赖 etoolbox 宏包。 1
非 ctexart 类的 part/beforeskip 和 part/afterskip 选项有意义。 100
给 enumitem 宏包注册 \chinese 和 \zhnum。 94
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。 . 157
将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。 119
删去 etoolbox 与 breqn 的兼容补丁。 130
新的标题格式选项 afterindent。 97
新的标题格式选项 numbering。 97
新的标题格式选项 runin。 97
新增子宏包 ctexpatch 实现给宏打补丁功能。 160

v2.3 (2015/09/17 – 2016/01/05)

General: .value_required: 和 .value_forbidden: 已过时。 44
代码实现避免使用 \lowercase 技巧(Joseph Wright)。 . 43
更新 LuaTeX-ja 支持(20150922.0)。 58
更新 unicode-math 宏包补丁。 61
兼容 titleps 宏包。 112
修复 nameformat 作用域问题。 96
与 LaTeX3 (2015/12/20) 同步。 41

v2.4 (2015/02/19 – 2016/04/25)

General: 初步支持 upLaTeX。 81
加强 beamer 宏包支持。 1
提供 upLaTeX 的 NFSS 字体定义。 133
提供 translator 宏包的中文定理名称翻译。 155
正确更新 CJK 包的 \CJKfamilydefault。 83
正确设置 upTeX 下字体命令。 81
\ctex_parse_name:NN: 修复宏名解析错误。 161

v2.4.1 (2016/04/26 – 2016/05/14)

General: beamer 不调整默认行距。 124
beamer 不调整默认字体大小。 124
使用 bootfont.bin 判断 Windows XP 以避免权限问题。 145
随字体更新 upTeX 的 \xkanjiskip。 88
新的标题格式选项 part/fixbeforeskip 和 chapter/fixbeforeskip。 98
正确更新 upLaTeX 的 \CJKfamilydefault。 83
\ccwd: 正确设置 upTeX 下的 \ccwd。 86
\chinese: 支持 \pagenumbering。 94
zihao: 不允许无参 zihao 选项。 45

v2.4.2 (2016/05/15)
General: 恢复 luatexja 对 \emshape 和 \eminnershape 的重定义。 61
\em: 兼容 upLaTeX 2016/05/07u00 的定义。 81
v2.4.3 (2016/06/03 – 2016/08/26)
General: 更新 unicode-math 补丁。 67
简化 fontspec 补丁。 61
确保 \proofname 非空。 95
删除选项 part/fixbeforeskip 和 chapter/fixbeforeskip。 98
新的标题格式选项 fixskip。 97
v2.4.4 (2016/09/09 – 2016/09/19)
General: 不再默认设置 xeCJK 的伪粗体。 58
改进 hyperref 宏包的标题锚点设置。 108
解决 zhmap 文件的 \catcode 问题。 52
使用 titlesec 时, 章节目录也使用 CTeX 的编号。 111
提供 \CTEXifname。 99
提供 \partmark。 99
新的标题格式选项 break。 97
v2.4.5 (2016/10/01 – 2016/10/25)
General: 新的标题格式选项 chapter/lofskip 和 chapter/lotskip。 98
新的标题格式选项 hang。 97
新的标题格式选项 tocline。 97
\ps@headings: 修复补丁失败。 114
v2.4.6 (2016/10/31 – 2016/11/20)
General: \CTEXifname 初始为假。 99
支持字体属性可选项在后的新语法。 72
重新初始化 \ifthechapter 等。 113
v2.4.7 (2016/12/23 – 2016/12/27)
General: 依赖 pxeverysel 宏包。 82
\ps@headings: 修复 ctexrep 类的 \chaptermark 汉化错误。 114
v2.4.8 (2017/02/23)
General: 解决与 fontspec 2017/01/24 v2.5d 的字体族匹配兼容问题。 74
v2.4.9 (2017/02/27)
General: 调整 unicode-math 补丁的代码顺序。 67
v2.4.10 (2017/07/19 – 2017/07/23)
General: 常数 \c_minus_one 已过时。 41
定义 \cht, \cdp 和 \cwd。 89
使用 lazy 函数对 Boolean 表达式进行最小化运算 (LaTeX3 2017/07/19)。 41
v2.4.11 (2017/08/17 – 2017/11/21)
General: 不把 Enclosed Alphanumerics 设置为 JAchar。 59
不把希腊和西里尔字母设置为 JAchar。 59
因上游 l3keys 变化, 重新定义 format␣+ 等带空格加号的选项。 97
\ps@headings: 补充页眉空格。 114
v2.4.12 (2017/12/05 – 2018/01/27)
General: 同步 LaTeX3 2017/12/16。 1
修正 \ctexset 在 ctexheading 包中无定义的错误(曾祥东)。 91
正确使用 \ltjsetkanjiskip 和 \ltjsetxkanjiskip。 80
\CTeX: 不依赖 \ifincsname。 129
v2.4.13 (2018/03/23)
\CTEX@selectfont@hook: 修正导言区 \selectfont 钩子位置。 87
v2.4.14 (2018/05/01)
General: 配置 macnew 的默认字体设置。 140
区分 macold 及 macnew。 139
为 macnew 配置字体命令。 146
v2.4.15 (2019/01/29 – 2019/04/05)
General: 定义 part/hang 和 chapter/hang。 100
将 upLaTeX 的默认字体由 mc 改为 zhrm, 并启用 \jfam。 81
将 JY2 和 JT2 编码的字体定义提取到单独的文件中。 132
局部指定 autoindent 为 false, 并交换 \CTEX@XXX@indent 与 \CTEX@XXX@format 的顺序。 115
同步 LaTeX3 2019/03/05。 1
统一"方正细黑一_GBK"的名称为 FZXiHeiI-Z08。 137
显式补丁 upLaTeX 的 \rmfamily 等字体命令。 81
修正 part/indent 和 chapter/indent 的实现方法。 100
v2.4.16 (2019/05/11 – 2019/05/29)
General: 更好地兼容 nameref 宏包。 109
修正 part/indent 和 chapter/indent 的实现方法, 在其标题内部禁用 autoindent。 100
允许设置 autoindent 为 0。 43
v2.5 (2019/10/25 – 2020/04/30)
General: ubuntu 改用思源(Noto CJK)和文鼎字库, 不再支持使用 pdfLaTeX 编译。 144
标题选项 format 也可以接受参数。 100
不再支持 Windows XP 系统, windowsold 和 windowsnew 成为过时字库选项。 145
不再自动载入 CJKfntef 或 xeCJKfntef 宏包。 85
操作系统检测移动至载入中文字库处, 且不再需要依赖特定引擎。 84
处理 \ctex_file_input:n 在 ctexsize 中未定义的错误。 41
给 LaTeX 和 upLaTeX 下的文档类指定驱动为 DVIPDFMx。 49
更新 LuaTeX-ja 支持(20200412.0)。 58
兼容 titletoc 宏包。 113
删除 fontspec 补丁。 61
使用环境变量代替绝对路径查找字体。 145
所有引擎下默认编码均设为 UTF-8。 45
为 macnew 增加粗楷体、隶书和圆体的定义。 140
移除 \CTEXunderdot、\CTEXunderline、\CTEXunderdblline、\CTEXunderwave、\CTEXsout、\CTEXxout、CTEXfilltwosides 等命令和环境。 85
应用新内核中的 \labelformat。 118
允许 macnew 在 LaTeX 和 upLaTeX 下使用。 140
在 ctexsize 也载入 fix-cm。 41
增加宏包开头钩子。 157
增加字体映射文件 zhmacfonts.tex。 150
重构标题选项 indent 和 hang。 98
重构字体选项 AlternateFont。 74
\CTEX@selectfont@hook: 兼容 KOMA-Script 的 \selectfont 补丁。 87
\ctex_if_platform_macos:TF: 改用 /System/Library/Fonts/Menlo.ttc 为特征文件。 131
fntef: 仅在该选项启用时会载入 CJKfntef 或 xeCJKfntef 宏包。 47

v2.5.1 (2020/05/02)
General: zhconv 更名为 ctex-zhconv。 1
v2.5.2 (2020/05/05 – 2020/05/06)
General: ctexmakespa.tex 更名为 ctex-spa-make.tex。 152
ctexspamacro.tex 更名为 ctex-spa-macro.tex。 . . 152
zhadobefonts.tex 等字体映射文件更名为 ctex-zhmap-*.tex。 147
兼容 LaTeX 2020-02-02 之前的版本。 82
新增标题选项 secnumdepth 和 tocdepth。 116
修正 macnew 和 ubuntu 字库的 CJKpunct 标点信息。 . 139
v2.5.3 (2020/05/31 – 2020/06/06)
General: 不再依赖 xunicode, 单独补丁 tuenc.def。 60
兼容 cleveref。 117
正确关闭和恢复 LaTeX3 语法环境。 42
v2.5.4 (2020/06/07 – 2020/08/16)
General: 更新 LuaTeX-ja 支持(20200808.0)。 58
兼容 LaTeX 2020/10/01 的钩子机制。 157
同时兼容 cleveref 和 hyperref。 117
修正主要字体命令补丁。 82
应用 l3cctab。 42
v2.5.5 (2020/10/06 – 2020/10/17)
General: 放弃应用 l3cctab。 42
进一步应用 LaTeX 2020/10/01 的新钩子。 83
\CJKhook: 不再通过旧的钩子命令来定义。 57
v2.5.6 (2021/01/11 – 2021/02/16)
General: 使用正确的导言区末尾钩子。 159
\f@nch@initialise: 更新 fancyhdr 宏包的补丁。 114
v2.5.7 (2021/06/04 – 2021/06/12)
General: 更好地兼容 cmap 包。 51
兼容 LaTeX 2021/06/01 的字体钩子。 41
禁用 DVIPDFMx 驱动的 unicode 书签设置。 84
使用 \disable@package@load 禁止宏包载入。 158
同时兼容 cleveref 和 beamer。 117
重新应用 l3cctab。 157
\CJK@addcmap: 应用 \pdfnobuiltintounicode。 53
\ctex_add_cmap:n: 确保 cmap 文件存在。 51
v2.5.8 (2021/11/18 – 2021/12/04)
General: 兼容 LaTeX 2021/11/15。 159
兼容 microtype。 63
兼容 titlesec 包和 \CTEXifname。 111
简化部分 Lua 函数。 61
v2.5.9 (2022/05/26 – 2022/05/27)
General: 设置消息模块的名字和类型。 41
依赖 chinese-jfm 宏包。 68
v2.5.10 (2022/06/10 – 2022/07/11)
General: 不直接依赖 xparse 和 l3keys2e。 41
更新一些内部函数。 41
解决 CJK 包与 \text_uppercase:n 等转化函数的冲突。 54
取消 LaTeX 2022-06-01 对书名号的定义。 52
展开传递 pagestyle 的值。 99
\ctex_add_cmap:n: 使用封装好的函数。 51
v2.6.0 (2024/04/20 – 2026/06/23)
General: macnew 增加 macOS 15+ 兼容, 字体运行时检测。 139
Breaking: \newCJKfontfamily 定义的字体切换命令改为局部定义, 与 \newcommand 行为一致。在分组内调用时, 命令不再泄漏到分组外(#751)。 72
Breaking: 移除对 LaTeX 2020/10/01 之前版本的字体钩子兼容代码。现在需要 LaTeX 2020/10/01 或更新版本(#746)。 82
4 字节 UTF-8 字符也优先使用 \DeclareUnicodeCharacter 的定义(#815)。 54
补全 upLaTeX 字体编码 JY2 和 JT2 的 Fallback 机制。 . . 133
将 experiment/font-size-system 的 traditional 选项更名为 letterpress(#813)。 44
使用 \l_keys_key_str 和 \l_keys_choice_str 替代已废弃的 _tl 版本(#806)。 44
提升 LaTeX3 版本至 2022/10/09。 41
提升 LaTeX3 最低版本要求至 2025/10/09。 41
文档: 在标准字体命令、中文字号表附近提示 experiment/font-size-system 的影响; 说明 letterpress 只是金属活字排印字号体系之一(#871)。 44
文档说明 runin 与 aftertitle 的交互(#574)。 23
文档字体统一为 Noto CJK 系列(#686)。 1
新增实验性 experiment/CJKecglue 选项(#717)。 90
新增实验性 experiment/font-size-system 选项(#543)。 44
修复 hyperref 在 ctex 之前被加载时 driverfallback 选项重复设置的警告(#715)。 84
修复参数签名, __ctex_char_auxi:NNNN 原误为 NNN。 . 54
修复了 TeX tree 字体无法制作 .spa 文件的问题。 152
修复内联 \verb 前 xkanjiskip 丢失的问题(移植 lltjcore 对 \verb 和 \do@noligs 的修复)。 80
\ctex_undeclare_unicode_character:n: 替换废弃命令 \char_to_utfviii_bytes:n。 52

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

\# 3517, 3576
.../afterindent *26*
.../aftername *22*
.../aftername+ *22*
.../afterskip *25*
.../aftertitle *22*
.../aftertitle+ *22*
.../beforeskip *25*
.../break *26*
.../break+ *26*
.../fixskip *25*
.../format *20*
.../format+ *20*
.../hang *24*
.../indent *25*
.../name *18*
.../nameformat *20*
.../nameformat+ *20*
.../number *19*
.../numberformat *21*
.../numberformat+ *21*
.../titleformat *22*
.../titleformat+ *22*
.../tocline *27*
\/ 1911
\\ 17, 18, 91, 147, 149, 201, 210, 830, 1396, 1478, 2617, 4106, 4107, 4219, 4220, 4326, 4540, 4542, 4571, 6058, 6059, 6060, 6153, 6248
\{ 1407, 1408, 6060, 6139, 6154
\} 1407, 1408, 6060, 6139, 6154
\␣ 3623, 3631, 3644, 3647, 3673, 3681
10pt *10*, 329
11pt *10*, 329
12pt *10*, 329

Numbers

\1 5027

A

\abovedisplayshortskip . 4382, 4390, 4401, 4421, 4429, 4440
\abovedisplayskip 4381, 4384, 4389, 4396, 4400, 4407, 4420, 4423, 4428, 4435, 4439, 4446
\abstractname 2601
abstractname *14*
\addCJKfontfeature 1478, 1529
\addCJKfontfeatures 1495
\addcontentsline 2831, 2876, 2880, 2885, 2917, 2921, 2926, 3026, 3031, 3037, 3043, 3172, 3186
\AddEnumerateCounter 2556, 2558, 2559, 2560
\addpenalty 3129, 3947, 3980, 4014, 4028, 4042
\addtocontents 2819, 2825, 3048, 3049
\addvspace 2820, 2826, 2837, 2840, 3048, 3049, 3129, 3133, 3697, 3708, 3716, 3727, 3735, 3746
adobefonts *36*
\algorithmname 2608, 2624
algorithmname *14*
AlternateFont *32*, *38*, 1614
\appendix 3341, 3823
appendix/name *28*, 3330
appendix/number *28*, 3330
appendix/numbering *27*, 3330
\appendixname 2603, 4091
appendixname *14*
\arabic 3994
\AtBeginDocument 570, 1888
\AtBeginDvi 434
\AtBeginShipoutFirst 439
\AtBeginUTFCommand 960
\AtEndOfClass 115
\AtEndOfPackage 116
\AtEndUTFCommand 961
autoindent *12*, 180, 2402

B

\baselinestretch 2461
\begin 3704, 3723, 3742
\begingroup 400, 2890, 2932, 2952, 2973, 3072, 3097, 3162, 3200, 3233, 3692, 3711, 3730, 5485, 5519, 5526
\belowdisplayshortskip . 4383, 4391, 4402, 4422, 4430, 4441
\belowdisplayskip 4384, 4396, 4407, 4423, 4435, 4446
\bfdefault 1263, 1269, 1604, 1611
\bfseries 2895, 2899, 2937, 2941, 2957, 2978, 3078, 3083, 3102, 3875, 3877, 3881, 3894, 3895, 3898, 3924, 3926, 3932, 3949, 3953, 3974, 4008, 4025, 4039
\bgroup 1903
\bibname 2606, 2609, 2617, 2625, 2631, 2632
bibname *14*
bool commands:
\bool_case:nTF 5861
\bool_const:Nn 49, 52
\bool_gset_false:N 243, 248
\bool_gset_true:N 238, 5033, 5174, 5280
\bool_if:NTF 504, 579, 1436, 1989, 2208, 3255, 3447, 3457, 3468, 3811, 4110, 5023, 5471, 6081
\bool_if_exist:NTF 6076
\bool_lazy_and:nnTF 5874
\bool_lazy_or:nnTF 4517, 5463, 6276
\bool_new:N 111, 1473, 5016, 5150
\bool_set_false:N 1448, 6082, 6091
\bool_set_true:N 1466, 6090

\c_false_bool 49, 3289, 3291
\c_true_bool 52, 3288, 3290
box commands:
\box_ht:N 4462
\box_new:N 57
\box_use_drop:N 3265
\box_wd:N 2145, 3264, 3572

C

c5size 35
cap 36, 288
captiondelimiter 37, 4469
\catcode 5485, 5486, 5487, 5488, 5489, 5493, 5494
CCT 35
cctab commands:
\cctab_begin:N 6083, 6098
\cctab_const:Nn 6102
\cctab_end: 6087, 6100
\cctab_select:N 6104
\c_document_cctab 6104
CCTfont 35
\ccwd 31, 109, 184, 2141, 2199, 2200, 2201, 2408, 2446, 2472, 2473, 2480, 2506, 2507, 2508, 2511, 2513, 2514, 2525, 2536, 2538, 4479, 4480, 4481
\cdp 2296
\centering 2933, 2974, 3693, 3712, 3731, 3881, 3893, 3898, 3902, 3932, 3953, 3961, 3996
\chapter 3003, 4116
chapter/beforeskip 37
chapter/fixbeforeskip 37
chapter/lofskip 27, 2742
chapter/lotskip 27, 2742
chapter/numbering 17
chapter/pagestyle 23, 2742
\chaptermark 3047
\chaptername 3921
char commands:
\char_generate:nn 727, 730, 731, 735, 736, 737, 741, 742, 743, 744
\char_set_catcode:nn 679
\char_set_catcode_active:n 672
\char_set_catcode_comment:n 848
\char_set_catcode_ignore:n 5057, 5248
\char_set_catcode_letter:n 1891, 3592, 3831, 6105, 6240
\char_set_catcode_other:N 1163, 3517, 3576
\char_set_catcode_other:n 6264
\char_set_catcode_space:n 966, 2301, 5040, 5200
\char_value_catcode:n 681
CharRange 32, 1614
\Chinese 37, 2552
\chinese 31, 2550, 2560, 3868, 3931
\cht 2296
\CJKaddEncHook 527
\CJKbold 4617, 4618, 4628, 4629, 4657, 4658, 4660, 4661, 4663, 4664, 4668, 4669, 4671, 4672, 4674, 4675, 4703, 4704, 4706, 4707, 4709, 4710, 4714, 4715, 4717, 4718, 4720, 4721
\CJKfamily 1335, 1355, 1709, 1919, 2014, 2015, 2016, 2017, 5453, 5454, 5455, 5456, 5458, 5459, 5461, 5473, 5474
\CJKfamilydefault 587, 1568, 1570, 1574, 1581, 1594, 2008, 2009, 2017, 2026, 2033, 2037, 2045, 2056, 2058
\CJKfontspec 1495
\CJKglue 2144, 2158, 2161, 2174, 2186
\CJKhook 761
\CJKnormal 4610, 4611, 4612, 4613, 4614, 4615, 4616, 4621, 4622, 4623, 4624, 4625, 4626, 4627, 4656, 4659, 4662, 4667, 4670, 4673, 4702, 4705, 4708, 4713, 4716, 4719
\CJKpunctmapfamily 796, 797, 798, 799, 800, 801, 805, 806, 807, 808, 809, 810
\CJKrmdefault 518, 1403, 1414, 1533, 1570, 1573, 1574, 2009, 2011, 2014, 2020, 2023, 2041, 4772, 4775, 4776, 4781, 4782, 4783, 4836, 4839, 4841, 4846, 4847, 4848, 4902, 4905, 4906, 4913, 4914, 4915, 4967, 4970, 4972, 4977, 4978, 4979, 5103, 5106, 5107, 5114, 5116, 5117, 5336, 5339, 5342, 5348, 5349, 5350, 5405, 5406, 5407, 5415, 5416, 5417
\CJKsfdefault 519, 1404, 1415, 1540, 2012, 2015, 2021, 2024, 2042, 4773, 4837, 4840, 4903, 4907, 4968, 4971, 5104, 5108, 5115, 5337, 5340, 5408, 5409
\CJKtilde 588
\CJKttdefault ... 520, 1405, 1416, 1546, 2013, 2016, 2022, 2025, 2043, 4774, 4838, 4904, 4969, 5105, 5338, 5342, 5410
clearalternatefont 33, 1793
\cleardoublepage 2851, 3004, 3891, 3919
\clearpage 2853, 3004, 3891, 3919
clist commands:
\clist_clear:N 1284, 5844
\clist_concat:NNN 1304
\clist_const:Nn 775
\clist_gput_right:Nn 345, 357, 383, 384
\clist_gset:Nn 1556
\clist_if_empty:NTF 1639
\clist_map_break:n 3426, 4101
\clist_map_function:nN .. 1622, 1796, 1798, 1831, 1845
\clist_map_inline:Nn 765, 1739
\clist_map_inline:nn 330, 788, 1838, 2136, 3421, 4098, 4242, 4265, 4296, 4344, 5831
\clist_new:N 126, 1301, 1557, 1635, 1682, 5889, 6053
\clist_put_left:Nn 1305, 4167, 4192
\clist_put_right:Nn 1465, 1728, 5883
\clist_set:Nn 1288, 1625, 1665, 1666, 1885, 6054
\clist_use:Nn 6063
\clubpenalty 3232, 3239
codepoint commands:
\codepoint_str_generate:n 526
\contentsname 2596
contentsname 13
continuation 14
cs commands:
\cs_generate_variant:Nn 80, 458, 1236, 1367, 1454, 1474, 1475, 1683,

1699, 1771, 1792, 1831, 1839, 3267, 3768, 6202, 6331, 6438
\cs_gset:Npn 650, 657, 664
\cs_gset_eq:NN 239, 244, 249, 441, 500, 501, 643, 644, 645, 646, 675, 1441, 1444, 3378, 3379, 3385, 3410
\cs_gset_protected:Npn 493, 3342, 3391, 3406
\cs_gset_protected:Npx 1332, 1703, 1789, 3318
\cs_gset_protected_nopar:Npx 1106
\cs_if_eq:NNTF . 1197, 3376, 3409, 6213, 6219, 6364, 6444
\cs_if_exist:NTF 3, 24, 321, 363, 426, 469, 1056, 1072, 1104, 1138, 1216, 1507, 1702, 1787, 2160, 2260, 3492, 3788, 3838, 4116, 4118, 6116
\cs_if_exist_p:N 6277, 6278
\cs_if_exist_use:N 1351, 1352
\cs_if_exist_use:NTF 652, 659, 666, 916, 1038, 1373, 1807, 6209
\cs_if_free:NTF 452, 462, 540, 561, 767, 783, 1438, 1603, 2210, 2219, 2222, 2421, 2435, 2556, 3440, 3485, 3542, 3547, 3551
\cs_new:Npn 101, 108, 592, 594, 600, 608, 614, 616, 625, 631, 633, 673, 710, 712, 714, 726, 728, 733, 739, 746, 747, 748, 1123, 1153, 1158, 1160, 1164, 1280, 1346, 1354, 1399, 1410, 1860, 2550, 2724, 3292, 3300, 3308, 3327, 3465, 3540, 3545, 5890, 6068, 6321, 6439
\cs_new:Npx 4488, 4494
\cs_new_eq:NN 95, 96, 97, 113, 114, 322, 459, 487, 591, 814, 1081, 1108, 1174, 1204, 1529, 1536, 2052, 2186, 2207, 2216, 2551, 2552, 2553, 2782, 2801, 2808, 3276, 3286, 3287, 3288, 3289, 3290, 3291, 3341, 3365, 3366, 3841, 4114, 4119, 4234, 5467, 5468, 5477, 5478, 5836, 6267, 6268, 6368, 6440
\cs_new_protected:Npn 60, 63, 65, 67, 69, 99, 115, 116, 118, 134, 136, 141, 422, 424, 428, 433, 438, 450, 460, 465, 467, 472, 516, 525, 532, 542, 683, 689, 695, 701, 707, 708, 709, 753, 759, 763, 781, 786, 794, 803, 812, 888, 890, 896, 898, 906, 908, 913, 926, 931, 946, 1035, 1045, 1059, 1064, 1070, 1125, 1136, 1148, 1167, 1176, 1192, 1206, 1231, 1281, 1302, 1311, 1330, 1347, 1357, 1382, 1420, 1432, 1434, 1456, 1458, 1481, 1564, 1589, 1598, 1621, 1623, 1630, 1637, 1643, 1650, 1652, 1657, 1663, 1684, 1693, 1700, 1706, 1717, 1725, 1731, 1737, 1755, 1764, 1772, 1784, 1802, 1819, 1837, 1840, 1842, 1855, 1874, 1882, 1896, 1912, 1914, 1938, 1952, 1965, 1978, 1994, 1999, 2031, 2063, 2067, 2119, 2127, 2131, 2141, 2155, 2198, 2202, 2217, 2251, 2255, 2275, 2420, 2422, 2447, 2455, 2467, 2469, 2488, 2498, 2503, 2658, 2688, 2735, 2765, 2772, 2777, 2785, 2792, 2799, 2803, 2809, 2811, 2814, 2830, 3244, 3250, 3252, 3260, 3268, 3277, 3358, 3368, 3402, 3413, 3433, 3438, 3443, 3453, 3463, 3513, 3518, 3566, 3584, 3762, 3784, 3793, 3799, 3809, 3826, 4210, 4225, 4329, 4334, 4364, 4370, 4484, 4500, 4510, 4512, 5828, 5837, 5881, 6070, 6078, 6085, 6097, 6099, 6107, 6114, 6127, 6135, 6143, 6149, 6156, 6166, 6168, 6179, 6181, 6183, 6185, 6195, 6197, 6203, 6208, 6211, 6217, 6224, 6229, 6234, 6245, 6249, 6254, 6259, 6269, 6274, 6293, 6344, 6359, 6369, 6381, 6390, 6409, 6414, 6426, 6441
\cs_new_protected:Npx 480, 919, 3494, 6164, 6170
\cs_new_protected_nopar:Npn 1082, 1096, 1098
\cs_replacement_spec:N 6285, 6307
\cs_set:Npn 2213, 3557
\cs_set_eq:NN 98, 185, 190, 195, 526, 539, 559, 756, 760, 769, 784, 900, 933, 941, 948, 950, 951, 1050, 1075, 1076, 1175, 1205, 1223, 1278, 1335, 1702, 1709, 1719, 1720, 1721, 1787, 1788, 1911, 2036, 2161, 2231, 2234, 2235, 2238, 2567, 2572, 2575, 2580, 2810, 2812, 3279, 3280, 3281, 3282, 3283, 3284, 3373, 3417, 3427
\cs_set_nopar:Npn 1937
\cs_set_nopar:Npx 3455, 6187
\cs_set_protected:Npn 755, 1209, 1224, 2158, 2190, 3487, 3555, 4371, 6272, 6336
\cs_set_protected:Npx 911, 1511, 2204, 6261, 6262
\cs_to_str:N *67*, 1506, 1997, 2002, 6270
\cs_undefine:N *52*, 1315, 1316, 1320, 1321, 1722, 2239, 6209
cs4size *35*
\csname 401, 404, 405, 3130, 3139, 3140, 3141, 3148, 3168, 3169, 3171, 3174, 3183, 3184, 3185, 3188, 3833, 3834, 5518, 5520, 5521, 5522, 5527, 5531
\CTeX *31*, 4465

ctex commands:

\ctex_add_cmap:n 455, 460, 460, 500
\ctex_add_dvi_zhmap: 433, 435, 441
\ctex_add_to_selectfont:n 2202, 2202, 2266, 2294, 2318, 2434
\ctex_after_end_preamble:n *43*, 98, 435, 6168, 6179, 6181
\ctex_appto_cmd:NnnTF 938, 1890, 6254, 6254
\ctex_assign_heading_name:nn .. 2694, 2732, 2732, 3334
\ctex_at_begin_package:nn . *160*, 836, 6143, 6143, 6154
\ctex_at_end:n 115, 115, 116, 261, 297, 303, 314, 319, 677, 848, 2186, 2188, 4578
\ctex_at_end_package:nn *160*, 436, 498, 838, 957, 963, 1214, 2079, 2082, 2108, 2116, 2123, 2554, 3389, 3404, 3473, 3480, 3505, 3550, 3552, 3553, 3589, 3635, 3797, 3847, 4467, 6156, 6156
\ctex_at_end_preamble:n 324, 578, 584, 2030, 6166, 6179, 6179
\ctex_at_shipout_first:n 428, 430, 438, 440, 2086, 2110
\ctex_auto_ignorespaces: 591, 591, 760, 2328
\ctex_CJK_input:n 532, 532, 539
\ctex_cleveref_hook: 3797, 3809, 3809
\ctex_declare_math_sizes:nnnn 4329, 4329, 4363
\ctex_define:n 59, 63, 1793, 1832, 2320, 2341, 2353, 2402, 2436, 2568, 2594, 2622, 2632, 2635, 2638, 2764, 3330, 3753, 3804, 4469, 4545
\ctex_define_option:n 59, 60, 156, 351, 354
\ctex_deprecated_command:Nn 134, 141, 2385, 2387, 2395, 2397, 2535, 2542
\ctex_deprecated_option:n 134, 138, 199, 208, 296, 302, 306, 313, 318, 4473
\ctex_deprecated_option:nn 134, 263
\ctex_detect_platform: 4500, 4500, 4515, 4530
\ctex_disable_package:n . 40, 41, 43, 44, 840, 6107, 6107
\ctex_family_cmap:nn 450, 450, 458, 459

\ctex_file_input:n 59, 360, 521, 815, 949, 2377, 4125, 4160, 4165, 4170, 4171, 4185, 4190, 4196, 4197, 4290, 4537, 4578, 5060, 5065, 5069, 5070, 6070, 6070
\ctex_fix_varioref_label:n 3796, 3799, 3799
\ctex_fontset_case:nnn .. 4488, 4488, 4887, 4952, 5387
\ctex_fontset_case:nnnn 141, 142, 4494, 4494, 4748, 4812, 5075, 5319
\ctex_fontset_error:n 4484, 4484, 4749, 4785, 4813, 4850, 4917, 4981, 5077, 5081, 5082, 5119, 5320, 5352
\ctex_gadd_hook:Nn 6195, 6196, 6197, 6202
\ctex_gadd_ltxhook:nn 426, 429, 2002, 2256, 6164, 6167, 6169, 6173, 6174
\ctex_gadd_package_hook:nnn 6147, 6160, 6170, 6195, 6195
\ctex_gadd_selectfont_hook:n .. 2251, 2255, 2259, 2263
\ctex_get_macro_meaning:N 6339
\ctex_get_macro_meaning:NTF 6335, 6361, 6417
\ctex_hang_from:n 3252, 3256, 3260
\ctex_heading_depth:nn .. 3756, 3759, 3762, 3762, 3768
\ctex_heading_glue:n 3268, 3268, 3276
\ctex_heading_hang:Nnn .. 3246, 3251, 3252, 3252, 3267
\ctex_hypersetup:n 84, 322, 2060, 2063, 2067, 2077, 2081, 2093, 2096, 2100, 2102
\ctex_if_autoindent_touched:TF 113, 113, 185, 190, 195, 4152
\ctex_if_ccglue_touched: 2171
\ctex_if_ccglue_touched:TF 2171, 2192, 2457
\ctex_if_ccglue_touched_p: 2171
\ctex_if_format_at_least:nTF 48, 6068, 6068, 6125, 6162, 6172
\ctex_if_platform_macos:TF 4505, 4510, 4510
\ctex_if_preamble:TF 97, 97, 98, 4549
\ctex_if_rescanable:NnTF 6359, 6359, 6393
\ctex_ignorespaces_case:N . 753, 753, 2325, 2328, 2331
\ctex_indent_box:n 2772, 2772, 2782
\ctex_load_fontset: 4512, 4512, 4544, 4557, 4577
\ctex_load_std_class:n ... 117, 118, 387, 390, 393, 396
\ctex_load_zhmap:nnnn 516, 516, 523, 4779, 4844, 4911, 4975, 5112, 5346, 5413
\ctex_ltj_add_font_features:n . 1456, 1456, 1474, 1526
\ctex_ltj_add_font_features:nn 1456, 1457, 1458, 1475
\ctex_ltj_add_kyenc:n 990, 1259
\ctex_ltj_char_range_key:nn 1879, 1882, 1882
\ctex_ltj_clear_alt_font:n 1014, 1750
\ctex_ltj_clear_alternate_font:n 1796, 1802
\ctex_ltj_declare_alternate_shape:nnnnnn 1711, 1731, 1731, 1733
\ctex_ltj_declare_char_range:n 1834, 1837, 1839
\ctex_ltj_declare_char_range:nn ... 1841, 1842, 1842
\ctex_ltj_def_char_range_key:n 1847, 1874, 1874
\ctex_ltj_ensure_default_family: .. 1564, 1564, 2049
\ctex_ltj_extract_font: 1075, 1082, 1082
\ctex_ltj_family_if_exist:nN 1368, 1381
\ctex_ltj_family_if_exist:nNTF 1359, 1368, 1568, 1573, 1591, 1594, 1805
\l_ctex_ltj_family_tl 1361, 1366, 1423, 1424, 1439, 1443, 1446, 1457, 1799, 1800
\ctex_ltj_fontspec:nn 1420, 1420, 1432, 1433, 1467, 1521
\ctex_ltj_get_and_define_fonts:nN . 1167, 1167, 1175
\ctex_ltj_get_and_define_fonts_al:nN ... 1171, 1174
\ctex_ltj_get_and_define_fonts_ja:nN ... 1170, 1176
\ctex_ltj_if_alternate_shape_exist:n 1118
\ctex_ltj_if_alternate_shape_exist:nTF 1085, 1100, 1118, 1127
\ctex_ltj_if_jfont:nTF 68, 1153, 1153, 1165, 1169
\ctex_ltj_if_jfont_math:NTF 1160, 1160, 1199
\ctex_ltj_is_kenc:n 991, 1155
\ctex_ltj_math_group_hook: 67, 1198, 1204, 1224
\ctex_ltj_patch_external_font:n 992, 1124
\ctex_ltj_pickup_alt_font:nn 1018, 1129
\ctex_ltj_pickup_font: 1050, 1052, 1070, 1070, 1081, 1143, 1179, 1181, 1183
\ctex_ltj_reset_alternate_font:n 1798, 1819
\ctex_ltj_save_alternate_family:Nnnn 1676, 1697, 1700, 1700
\ctex_ltj_save_alternate_seq:Nn 1647, 1652, 1652, 1683
\ctex_ltj_save_char_range:n 1845, 1852, 1852
\ctex_ltj_save_char_range:nn 1854, 1855
\ctex_ltj_select_alternate_font: .. 1125, 1125, 2269
\ctex_ltj_select_font: 1035, 1035, 2268
\ctex_ltj_set_alt_font:nnnn 1007, 1766, 1769
\ctex_ltj_set_alternate_family:nn . 1651, 1684, 1684
\ctex_ltj_set_alternate_family:nnn 1696, 1704, 1706, 1706
\ctex_ltj_set_alternate_family:Nnnnn 1688, 1693, 1699
\ctex_ltj_set_alternate_family:nnTF 1289, 1637, 1637
\ctex_ltj_set_alternate_prop:n 1616, 1621, 1621
\ctex_ltj_set_alternate_shape:n 1757, 1764, 1771, 1776
\ctex_ltj_set_alternate_shape:nnN . 1743, 1772, 1772
\ctex_ltj_set_alternate_shape:nnn . 1746, 1752, 1752
\ctex_ltj_set_alternate_shape:nnnn 1754, 1755
\ctex_ltj_set_alternate_shape:Nnnnnnn 1734, 1737, 1737
\ctex_ltj_set_family:nnn 1281, 1281, 1451, 1498, 1515, 1533, 1540, 1546, 1552
\ctex_ltj_set_kanjiskip:N 1912, 1912, 2165
\ctex_ltj_set_math_letter:NN 1206, 1218, 1226
\ctex_ltj_set_xkanjiskip:N 1912, 1914, 2282, 2366
\ctex_ltj_subst_font: 1076, 1098, 1098
\ctex_ltj_swap_cs:NN 1710, 1717, 1717
\ctex_ltj_switch_family:n 1356, 1357, 1367, 1424, 1452, 1512
\ctex_ltj_update_mathfont: 1586, 1589, 1589
\ctex_ltj_update_mathfont:n 1592, 1595, 1598
\ctex_ltj_use_jfont: 1092, 1096, 1096
\ctex_ltj_use_math_group:Nn ... 1192, 1192, 1205, 1223
\ctex_ltj_zero_globaldefs: 893, 1030
\ctex_make_spa:nn 5828, 5828, 5836
\ctex_parse_name:NN 2242, 6265, 6269, 6269
\ctex_parse_name:NNn 6270, 6274, 6331

\ctex_patch_boot:NNnnTF 6226, 6231, 6236, 6251, 6256, 6259, 6259
\ctex_patch_cmd:Nnn 563, 1933, 3507, 3520, 3526, 3532, 3605, 3608, 3613, 3619, 3630, 3637, 3640, 3643, 3646, 3655, 3658, 3663, 3669, 3680, 4479, 4480, 4481, 6234, 6234
\ctex_patch_cmd_all:NnnnTF 3828, 6229, 6229
\ctex_patch_cmd_once:NnnnTF ... 1927, 2244, 6224, 6224
\ctex_patch_failure:N 942, 944, 1894, 2249, 3583, 3595, 3836, 6243, 6245
\ctex_plane_to_utfxvibe:Nn 540, 542, 559
\ctex_pop_file: 78, 537, 6070, 6074, 6085, 6099
\ctex_preto_cmd:NnnTF 934, 3580, 3591, 6249, 6249
\ctex_provide_font_hook:NNN 1994, 1994, 2004, 2005, 2006, 2007
\ctex_punct_map_bfseries:nn 786, 786, 4775, 4782, 4839, 4840, 4847, 4906, 4907, 4908, 4914, 4970, 4971, 4978, 5107, 5108, 5109, 5116, 5340, 5341, 5349, 5406, 5409, 5416
\ctex_punct_map_family:nn 781, 781, 4772, 4773, 4774, 4781, 4836, 4837, 4838, 4846, 4902, 4903, 4904, 4913, 4967, 4968, 4969, 4977, 5103, 5104, 5105, 5114, 5115, 5336, 5337, 5338, 5348, 5405, 5408, 5410, 5415
\ctex_punct_map_itshape:nn 803, 803, 4776, 4783, 4841, 4848, 4905, 4915, 4972, 4979, 5106, 5117, 5339, 5350, 5407, 5417
\ctex_punct_map_series:nnn 790, 791, 794
\ctex_punct_set:n . 763, 763, 4771, 4780, 4835, 4845, 4901, 4912, 4966, 4976, 5102, 5113, 5335, 5347, 5391, 5414
\ctex_punct_space:nn 812, 812, 814
\ctex_push_file: 71, 534, 6070, 6072, 6078, 6097
\ctex_replace_package:nn 6125, 6127, 6135, 6139
\ctex_save_font_size:nn 9, 4223, 4234
\ctex_scheme_input:n 69, 69, 80, 3849, 4127
\ctex_select_size: .. 2409, 2416, 2419, 2420, 2441, 2501
\ctex_set:n 59, 65, 261, 2379, 2399, 4151, 4153
\ctex_set:nn 34, 59, 67, 139, 2389, 3852, 3910, 3938, 3970, 4005, 4022, 4036, 4052, 4060, 4068, 4076, 4083, 4084, 4085, 4585
\ctex_set_default_ccwd:Nn 99, 99, 194, 2415, 2440
\ctex_set_deprecated_option:n 134, 136, 173, 174, 228, 229, 230, 252, 257, 288, 289
\ctex_set_font_size:Nnn 4364, 4364, 4379, 4387, 4398, 4409, 4410, 4411, 4412, 4413, 4414, 4415, 4418, 4426, 4437, 4448, 4449, 4450, 4451, 4452, 4453, 4454
\ctex_set_ignorespaces: 753, 755, 757, 759, 762
\ctex_set_jfm:n 1231, 1231, 1236, 1247, 1251, 2348
\ctex_set_upfamily:nnn 1938, 1938, 1985, 4795, 4796, 4797, 4798, 4860, 4861, 4862, 4863, 4927, 4928, 4929, 4930, 4931, 4932, 4991, 4992, 4993, 4994, 5129, 5130, 5131, 5132, 5133, 5134, 5362, 5363, 5364, 5429, 5430, 5431, 5432, 5433, 5434, 5435
\ctex_set_upfonts:nnnnnn 1978, 1978, 1988, 4788, 4853, 4920, 4984, 5422
\ctex_set_upmap:nnn 1952, 1952, 1980, 1981, 1982, 1983, 1986, 4933, 4934, 5123, 5124, 5125, 5126, 5361, 5436, 5437
\ctex_set_upmap_unicode:nnn 1965, 1965, 1987, 5127, 5128, 5355, 5357, 5359
\ctex_set_zhmap:n 421, 422, 445, 521, 1954, 1967
\ctex_titleps_hook: 3516, 3518, 3551, 3552
\ctex_titlesec_hook: 3413, 3413, 3477
\ctex_undeclare_unicode_character:n 525, 525, 529, 530
\ctex_update_ccglue: 2155, 2155, 2464, 2486, 2529
\ctex_update_ccwd: 2141, 2141, 2193, 2458, 2537
\ctex_update_default_family: 53, 578, 2030, 2031, 2031
\ctex_update_em_unit: 2198, 2198, 2449, 2505
\ctex_update_kanjisize: 2296, 2318
\ctex_update_parindent: 2427, 2488, 2488, 2530
\ctex_update_size: 2421, 2422, 2422, 2435
\ctex_update_stretch: 2426, 2447, 2447
\ctex_update_xkanjiskip: 2274, 2275, 2294
\ctex_update_ziju: 2429, 2503, 2503
\ctex_use_package_hook:nn 6203, 6203, 6214, 6220
\ctex_use_zhmap: 424, 430, 434, 440
\ctex_varioref_hook: 3793, 3793, 3839, 3847
\ctex_zhmap_case:nnn 142, 114, 114, 239, 244, 249, 4751, 4815, 4889, 4954, 5086, 5322, 5389
\ctex_zihao:n 4209, 4210, 4210
\ctex_ziju:n 2497, 2498

ctex internal commands:

__ctex_add_cmap_auxi:n 463, 465
__ctex_add_cmap_auxii:Nn 466, 467
\g__ctex_after_end_preamble_hook_tl 6182, 6186, 6194
__ctex_assign_heading_name:nnn ... 2732, 2734, 2735
\l__ctex_autoindent_tl 112, 184, 189, 194, 2408, 2412, 2415, 2490, 2493, 2768
__ctex_calc_bounds:nn 5885, 5886, 5890
__ctex_ccglue: 2160, 2161, 2174, 2186
\l__ctex_ccglue_skip 2159, 2165, 2168, 2170, 2178, 2182, 2462, 2476, 2485, 2520, 2528
__ctex_char:N 592, 643
__ctex_char:NN 594, 644
__ctex_char:NNN 608, 645
__ctex_char:NNNN 625, 646
__ctex_char_aux:NN 597, 598, 600, 650, 654
__ctex_char_auxi:NNNN 611, 614
__ctex_char_auxi:NNNNN 628, 631
__ctex_char_auxii:NNN 612, 615, 616, 657, 661
__ctex_char_auxii:NNNN 629, 632, 633, 664, 668
__ctex_char_preproc:w 673, 675
__ctex_char_raw:n 726, 749
__ctex_char_raw:nn 728, 746
__ctex_char_raw:nnn 733, 747
__ctex_char_raw:nnnn 739, 748
__ctex_char_raw_nn:n 746, 750
__ctex_char_raw_nnn:n 747, 751
__ctex_char_raw_nnnn:n 748, 752
__ctex_char_wrap:nn 710, 723
__ctex_char_wrap:nnn 712, 724
__ctex_char_wrap:nnnn 714, 725
\c__ctex_class_tl 72, 74, 120, 4114, 4120, 4122, 4123, 4125
__ctex_clear_fntef_color:n ... 2119, 2127, 2131, 2138

__ctex_cleveref_hook_aux:N 3809, 3817, 3820, 3821, 3823, 3826
\c__ctex_cmap_encoding_seq 448, 454
\l__ctex_cmap_file_tl 475, 479, 489
\l__ctex_ctexcap_options_clist 6053, 6054, 6063
__ctex_def_heading_keys:n 2688, 2688, 2762
__ctex_default_ccwd_aux:n 100, 101
__ctex_default_ccwd_aux:w 104, 108
__ctex_disable_package_aux:nnnn 6111, 6114
\g__ctex_encoding_tl 110, 218, 221, 224, 373, 377, 507, 512, 581, 587, 647, 2075, 2547, 4155, 4181, 4195
\g__ctex_end_preamble_hook_tl 6180, 6184, 6193
\c__ctex_engine_file_str 23, 2377
\c__ctex_engine_str 23
\c__ctex_everysel_loaded_bool 46, 1989, 2208
\g__ctex_expl_status_seq 6080, 6088, 6094
\l__ctex_expl_status_tl 6088, 6089, 6093
\l__ctex_family_default_init_tl 2033, 2051
__ctex_family_default_wrap:n 2036, 2052, 2055
\g__ctex_font_size_int 151, 167, 168, 169, 340, 344, 381, 4172, 4173, 4198, 4199, 4372
\c__ctex_font_size_prop 4212, 4223, 4339, 4366
\c__ctex_font_size_seq 4221, 4320
\g__ctex_font_size_system_tl 160, 4238, 4288, 4291, 4295
\l__ctex_font_size_temp_prop . . 4223, 4227, 4236, 4319
\l__ctex_font_size_temp_seq . . . 4224, 4232, 4237, 4321
\l__ctex_font_size_tl 4212, 4213, 4331, 4332, 4366, 4367
\g__ctex_fontset_tl 227, 4503, 4506, 4507, 4514, 4518, 4519, 4522, 4523, 4526, 4532, 4537, 4554, 4556, 4561, 4576
__ctex_get_font_sizes:Nn 4331, 4334, 4334
__ctex_get_macro_meaning:w 6335, 6341, 6344
\l__ctex_heading_bool 265, 4110
\c__ctex_heading_level_prop 3764, 3769
\l__ctex_heading_skip 2789, 2796, 2798, 2800, 2801, 2816, 2817, 2820, 2822, 2823, 2826, 3271, 3272, 3273
\c__ctex_headings_cs_seq 3360, 3374, 3384, 3407
\c__ctex_headings_seq . . . 2649, 2759, 3490, 3564, 3795
__ctex_hookto_cmd:Nnnnw 6251, 6256, 6414, 6414
__ctex_hookto_cmd_parameter:Nnnnw . 6421, 6441, 6441
__ctex_hookto_cmd_parameterless:Nnnnw 6420, 6426, 6426
__ctex_if_font_exists:n 5857
__ctex_if_font_exists:nTF 5840, 5857
\l__ctex_if_otf_exists_tl 5859, 5864, 5870, 5875, 5879
\l__ctex_if_ttf_exists_tl 5860, 5866, 5872, 5876, 5879
__ctex_initial_heading:n 2658, 2658, 2761
__ctex_insert_indent: 2775, 2777, 3258
__ctex_left_hook_aux:nn 6439
\l__ctex_line_spread_fp 129, 177, 4174, 4175, 4200, 4201, 4458, 4460
\l__ctex_line_stretch_tl 2440, 2445, 2446, 2450
\l__ctex_ltj_add_alternate_bool 1436, 1448, 1466, 1473
__ctex_ltj_alternate_cs:n 1316, 1319, 1320, 1321, 1351, 1352, 1354, 1439, 1442, 1443, 1445, 1446, 1647, 1689, 1749, 1768, 1808, 1825
\l__ctex_ltj_alternate_family_tl 1713, 1716
\l__ctex_ltj_alternate_options_clist 1666, 1667, 1678, 1682
\l__ctex_ltj_alternate_prop 1285, 1632, 1633, 1636, 1646, 1654
\l__ctex_ltj_base_CJKfamily_tl 1286, 1300, 1349, 1749, 1768
\l__ctex_ltj_base_family_tl 78, 1317, 1344, 1350, 1695, 1735, 1788, 1805, 1811, 1812, 1824
\l__ctex_ltj_char_range_clist 1284, 1618, 1639, 1665, 1677, 1690, 1712, 1734, 1885
\g__ctex_ltj_char_range_prop 1741, 1842
\l__ctex_ltj_char_range_tl 1741, 1744, 1844, 1846, 1848, 1850, 1857
__ctex_ltj_chardef_text_cmd: 902, 906
__ctex_ltj_chardef_text_cmd_aux:N 907, 908
__ctex_ltj_check_family:n 1311, 1311, 1645
\l__ctex_ltj_cmd_tl 901, 905, 907
\l__ctex_ltj_current_font_tl 64, 1038, 1042, 1043, 1056, 1104, 1109
\l__ctex_ltj_current_shape_tl . 1127, 1130, 1133, 1134
__ctex_ltj_declare_char_range:nn 1838, 1840
\g__ctex_ltj_default_features_clist 1304, 1556, 1557
__ctex_ltj_family_csname:n . . . 1315, 1332, 1346, 1373
\g__ctex_ltj_family_font_name_prop 1291, 1308, 1313, 1384, 1460, 1566, 1578
\g__ctex_ltj_family_font_options_prop 1292, 1308, 1463
\g__ctex_ltj_family_int . 1426, 1428, 1727, 1728, 1730
\g__ctex_ltj_family_name_prop 1308, 1317, 1338, 1370, 1695
__ctex_ltj_family_unknown_warning:n 1364, 1382, 1382, 1816
\l__ctex_ltj_font_options_clist 1288, 1293, 1294, 1295, 1301, 1337, 1464, 1465, 1468, 1686, 1687, 1691
\g__ctex_ltj_fontname_seq 1061, 1066, 1069
__ctex_ltj_fontspec:nnn 1427, 1434, 1454
\g__ctex_ltj_fontspec_family_tl 1336, 1339, 1345, 1350, 1375
\g__ctex_ltj_fontspec_prop 1422, 1450, 1455
__ctex_ltj_gset_family_cs:nn 1296, 1330, 1330
__ctex_ltj_if_alt_set:nTF 1029, 1120
__ctex_ltj_if_jfont_math:w 1161, 1164
\l__ctex_ltj_jfm_tl 1231, 1262, 1264, 1306
__ctex_ltj_ltj_if_jfont:w 1155, 1158
\c__ctex_ltj_math_fam_int 1270, 1271
\c__ctex_ltj_math_family_tl 1600, 1602, 1604, 1607, 1611
\c__ctex_ltj_math_tl 1265, 1266, 1268, 1270, 1552, 1591, 1601, 1606, 1610
__ctex_ltj_msg_def_family_map:n 1397, 1399
__ctex_ltj_msg_family_map:n 1329, 1396, 1410
__ctex_ltj_noembed_wrap:n 1278, 1280
__ctex_ltj_pass_args:nnnn 1481, 1481, 1497, 1514, 1520, 1532, 1539, 1545, 1551
__ctex_ltj_patch_external_font:w . 1088, 1123, 1123

__ctex_ltj_patch_text_cmd:NN 928, 931
__ctex_ltj_pickup_alt_font:Nn 1024, 1151
__ctex_ltj_pop_fontname: 1055, 1064, 1145
__ctex_ltj_post_arg:w 1484, 1490
__ctex_ltj_push_alternate_prop:n 1622, 1623
__ctex_ltj_push_alternate_prop:nn . 1627, 1630, 1886
__ctex_ltj_push_fontname:n 1051, 1059, 1142
\c__ctex_ltj_range_max_int 1869, 1873
\c__ctex_ltj_range_min_int 1868, 1872
__ctex_ltj_range_normalization:nn . 1759, 1858, 1860
\l__ctex_ltj_redirect_jfm_prop 1231
\g__ctex_ltj_reset_alternate_prop .....
..... 1322, 1810, 1822, 1830
__ctex_ltj_save_alternate_auxi:w 1655, 1657
__ctex_ltj_save_alternate_auxii:w . 1660, 1661, 1663
__ctex_ltj_save_alternate_shape:Nn .....
..... 1748, 1767, 1784, 1784, 1792
__ctex_ltj_select_font_aux: .. 1039, 1045, 1045, 1057
__ctex_ltj_set_alternate_family:n 1340, 1347
__ctex_ltj_set_alternate_family_aux:nnn 1641, 1650
__ctex_ltj_set_family_aux:nnn 1640, 1643
__ctex_ltj_tmp:w 1719, 1721, 1722
\l__ctex_ltj_tmp_clist 1625, 1626, 1628, 1635
\l__ctex_ltj_tmp_tl 1066, 1067, 1287,
1288, 1313, 1324, 1327, 1461, 1469, 1505, 1512, 1515,
1568, 1573, 1591, 1592, 1594, 1595, 1669, 1671, 1672, 1679
\g__ctex_ltj_unknown_family_seq ... 1386, 1388, 1393
__ctex_ltj_update_family_uid:N 1294, 1686, 1724, 1725
__ctex_ltj_use_global_options:N .....
..... 1295, 1302, 1302, 1667, 1687
\g__ctex_mac_baoli_dir_tl 5300, 5303
\g__ctex_mac_kaiti_dir_tl 5249, 5259, 5261, 5290, 5294
\g__ctex_mac_pingfang_bool 5150, 5174, 5280, 5471
\g__ctex_mac_pingfang_dir_tl 5268, 5273, 5279
\g__ctex_mac_stfangso_dir_tl 5282, 5286, 5288
\g__ctex_mac_yuanti_dir_tl 5305, 5309
\g__ctex_macos_ver_found_bool 5016, 5023, 5033
\g__ctex_macos_ver_ior 5015, 5019, 5021, 5036
\g__ctex_macos_ver_tl 5010, 5028, 5062, 5068
\c__ctex_msyh_suffix_tl .....
..... 5379, 5394, 5395, 5401, 5402, 5426, 5427
\c__ctex_package_cctab 6083, 6098, 6102
__ctex_package_loaded_warning:nn 6146, 6149
\l__ctex_parameter_str 6332, 6351, 6372, 6419
\l__ctex_parameter_tl 6372, 6377, 6388
__ctex_parse_name:nNNNnN 6284, 6293
__ctex_parse_name:w 6306, 6312, 6321
\l__ctex_patch_cleveref_bool 3806, 3811
__ctex_patch_cmd:Nnnnnw 6226, 6231, 6236, 6390, 6390
__ctex_patch_false:w 6262, 6268, 6405, 6407, 6424, 6450
__ctex_patch_rebuild:Nn .....
..... 6363, 6369, 6369, 6402, 6443, 6447
__ctex_patch_replace:nnnTF 6397, 6409, 6409
__ctex_patch_rescan:NNn 6371, 6372, 6373, 6381
__ctex_patch_toc_width:n 3575, 3584, 3600, 3601
__ctex_patch_true:w 6261, 6267, 6403, 6436, 6448
__ctex_plus_key_aux:nn .....
..... 2714, 2716, 2717, 2718, 2719, 2720, 2721, 2724
\l__ctex_prefix_str 6332, 6350, 6371, 6428, 6429
\l__ctex_prefix_tl 6371, 6376, 6387
__ctex_provide_font_hook_aux:NNNN 1996, 1999
\l__ctex_punct_bounds_clist ... 5844, 5853, 5883, 5889
\c__ctex_punct_family_clist 765, 775
\l__ctex_punct_font 5841, 5842, 5843
\c__ctex_punct_seq 5820, 5845
\l__ctex_punct_tl .....
..... 254, 574, 575, 822, 1251, 2345, 2346, 2347, 2348
__ctex_rebuild_cmd:w ... 6363, 6364, 6368, 6443, 6444
\l__ctex_replacement_str .....
..... 6332, 6352, 6373, 6411, 6412, 6446
\l__ctex_replacement_tl 6373, 6378, 6389
__ctex_restore_selectfont: 2217, 2239, 2248
__ctex_right_hook_aux:nn 6440
__ctex_save_bounds:n 5847, 5881
__ctex_save_cmap:Nn 469, 472
__ctex_save_font_size:nn .....
..... 4223, 4225, 4234, 4261, 4284, 4315
\l__ctex_scheme_tl 276, 3849, 4127
\g__ctex_section_depth_int .....
..... 81, 270, 272, 4050, 4066, 4082
\c__ctex_section_headings_seq 2645, 3315, 3431
__ctex_set_font_size:nnNn 4367, 4370
\g__ctex_spa_iow 5830, 5833, 5835, 5850
\c__ctex_std_class_tl 124, 4098, 4113, 4114, 4179, 4202
\g__ctex_std_options_clist 117, 345, 357, 383, 384
__ctex_titlesec_format:n 3427, 3435, 3438
__ctex_titlesec_format:Nn 3417, 3441, 3443
__ctex_titlesec_format_explicit:Nn 3418, 3453
__ctex_titlesec_hook:n 3427, 3431, 3433
__ctex_titlesec_spacing:Nn 3436, 3463
__ctex_titlesec_spacing:nnnnnn 3464, 3465
__ctex_tmp:N 2213, 2242
__ctex_tmp:w 6272, 6325, 6336, 6357
\l__ctex_tmp_box .....
..... 55, 2144, 2145, 3263, 3264, 3265, 3568, 3572
\l__ctex_tmp_dim 55,
2450, 2451, 2460, 2463, 2472, 2510, 2512, 2514, 2523, 2525
\l__ctex_tmp_int 55, 544,
546, 547, 549, 553, 555, 2471, 2473, 2474, 2480, 2481,
2512, 2513, 2514, 2517, 2518, 2523, 2525, 3314, 3317, 3321
\l__ctex_tmp_tl 55, 329, 336, 351, 352,
2215, 2244, 2690, 2744, 2757, 2764, 3764, 3765, 4339, 4340
__ctex_update_stretch_auxi: 2452, 2455
__ctex_update_stretch_auxii: 2190, 2453, 2467
__ctex_update_stretch_auxiii: 2194, 2468, 2469
__ctex_use_package_hook_aux:N 6206, 6208
__ctex_write_family:nn 153, 5832, 5837
\l__ctex_xkanjiskip_skip 2274, 2365, 2366, 2367
\l__ctex_xkanjiskip_tl 2281, 2286, 2287, 2364
\c__ctex_zero_tl 2424, 2431, 2433
\g__ctex_zhmap_tl 423, 425, 444
\g__ctex_zhmCJK_bool . 111, 238, 243, 248, 504, 579, 1993

\l__ctex_ziju_dim 2506, 2507, 2511, 2522, 2524, 2528, 2532
\l__ctex_ziju_tl 2424, 2432, 2433, 2500, 2506
ctex.newluacmd 969
\ctexbibname 2618, 2638
\CTEXcounter 2553
\CTEXdigits 31, 2563
CTEXfilltwosides 36, 38
\CTEXifname 29, 2808, 2898, 2940, 3082, 3294, 3302, 3310, 3559, 3607, 3610, 3615, 3627, 3632, 3639, 3642, 3645, 3648, 3657, 3660, 3665, 3677, 3682
\CTEXindent 36, 2533
\CTEXnoindent 36, 2533
\CTEXnumber 31, 2563
\CTEXnumberline 3308, 3328, 3928
\CTEXoptions 37, 2381
ctexpar internal commands:
__ctexpar 3204
\ctexset 5, 2378, 2388, 2398
\CTEXsetfont 36, 2419
\CTEXsetup 37, 2381
\CTEXsout 36, 38
\ctexspadef 812, 5852
\CTEXthechapter 28, 3025, 3303, 3483, 3627, 3645, 3677
\CTEXtheparagraph 28
\CTEXthepart 28, 3295, 3488
\CTEXthesection 28, 3607, 3615, 3632, 3639, 3648, 3657, 3665, 3682
\CTEXthesubparagraph 28
\CTEXthesubsection 28, 3610, 3642, 3660
\CTEXthesubsubsection 28
\CTEXunderdblline 36, 38
\CTEXunderdot 36, 38
\CTEXunderline 36, 38
\CTEXunderwave 36, 38
\CTEXxout 36, 38
\CurrentOption 357
\cwd 2296

D

\d 5027
declarecharrange 32, 1832
\DeclareErrorKanjiFont 1921
\DeclareFontEncoding 1253
\DeclareFontFamily 490, 1260, 4596, 4597, 4598, 4599, 4600, 4601
\DeclareFontShape 1261, 1263, 1942, 1943, 1946, 1947, 1948, 1949, 4610, 4611, 4612, 4613, 4614, 4615, 4616, 4617, 4618, 4621, 4622, 4623, 4624, 4625, 4626, 4627, 4628, 4629, 4632, 4633, 4634, 4635, 4636, 4637, 4638, 4639, 4640, 4643, 4644, 4645, 4646, 4647, 4648, 4649, 4650, 4651, 4656, 4657, 4658, 4659, 4660, 4661, 4662, 4663, 4664, 4667, 4668, 4669, 4670, 4671, 4672, 4673, 4674, 4675, 4678, 4679, 4680, 4681, 4682, 4683, 4684, 4685, 4686, 4689, 4690, 4691, 4692, 4693, 4694, 4695, 4696, 4697, 4702, 4703, 4704, 4705, 4706, 4707, 4708, 4709, 4710, 4713, 4714, 4715, 4716, 4717, 4718, 4719, 4720, 4721, 4724, 4725, 4726, 4727, 4728, 4729, 4730, 4731, 4732, 4735, 4736, 4737, 4738, 4739, 4740, 4741, 4742, 4743
\DeclareFontSubstitution 1256
\DeclareKanjiFamily 1940, 1941, 4602, 4603, 4604, 4605, 4606, 4607
\DeclareKanjiSubstitution 1922, 1923
\DeclareMathSizes 4332
\DeclareSymbolFont 1266, 1601, 1924
\def 407, 1900, 2871, 2912, 2948, 2969, 2985, 3016, 3057, 3067, 3092, 3112, 3138, 3142, 3179, 3194, 3210, 3211, 3215, 4392, 4403, 4431, 4442, 5492, 5528
\defaultCJKfontfeatures 1530
\defbeamertemplate 3691, 3710, 3729
\defbeamertemplatealias 3748, 3749, 3750
\detokenize 408
dim commands:
\dim_abs:n 2524
\dim_add:Nn 2507
\dim_compare:nNnTF 2451, 2492, 2508, 2515, 2779, 2788, 3272
\dim_eval:n 2478, 2523
\dim_max:nn 3571
\dim_min:nn 2524
\dim_new:N 58, 2154, 2297, 2298, 2299, 2532
\dim_set:Nn 2145, 2199, 2200, 2201, 2450, 2460, 2493, 2506, 2510, 2538, 2774, 2795, 3254, 3569, 4462
\dim_sub:Nn 2514
\dim_to_decimal:n 4229, 4230, 4341
\dim_to_decimal_in_unit:nn 5896
\dim_use:N 105, 4462
\dim_zero:N 2544
\c_max_dim 2451, 2788
\c_zero_dim 2463, 2478, 2492, 2508, 2515, 2703, 2779, 3272, 4083
\directlua 871, 875
\do 412, 1904
\document 6187, 6190
\dospecials 1904

E

\edef 5511, 5512
\else 402, 405, 406, 410, 1899, 1902, 1929, 1930, 1934, 1935, 2843, 2852, 2861, 2877, 2882, 2918, 2923, 3004, 3012, 3028, 3034, 3040, 3053, 3063, 3123, 3128, 3147, 3152, 3173, 3178, 3187, 3209, 3225, 3238, 3891, 3919, 5523, 5524, 5552, 5576, 5625, 5674, 5698, 5730, 5779
else commands:
\else: 2175
\em 1927
\eminnershape 1929, 1930, 1935, 1937
\end 3707, 3726, 3745, 5965
\endcsname 401, 404, 405, 3130, 3139, 3140, 3141, 3148, 3168, 3169, 3171, 3174, 3183, 3184, 3185, 3188, 3833, 3834, 5518, 5520, 5521, 5522, 5525, 5527, 5531
\endgroup 402, 419, 2903, 2944, 2960, 2981, 3086, 3105, 3170, 3208, 3233, 3709, 3728, 3747, 5511, 5519, 5534

\endinput 402, 5516
\endlinechar 5490, 5491, 5869, 5871
\everypar 3127, 3228, 3240
\EverySelectfont 2252
exp commands:
\exp_after:wN 104, 907, 1072, 1088, 1090, 1161, 3464, 4213, 4332, 4367, 6341
\exp_args:cc 3789
\exp_args:Nc 466, 1996, 3436, 3441, 6295
\exp_args:Nco 3557
\exp_args:Ne 463, 474, 506, 545, 584, 593, 677, 819, 2002, 2497, 2546, 2804, 4209, 4460
\exp_args:NNc 928, 1037, 6205
\exp_args:Nnc *117*
\exp_args:NNe 2690, 4318, 4320
\exp_args:Nne 649, 656, 663, 3802, 6118
\exp_args:NNNo 4461
\exp_args:NNo 2764
\exp_args:Nno 492
\exp_args:No . 121, 351, 674, 1169, 1627, 3460, 5847, 6357
\exp_last_unbraced:Ne 602, 618, 635, 6306
\exp_last_unbraced:NNNNo 107
\exp_last_unbraced:NNo 2243
\exp_not:N . 575, 586, 587, 588, 652, 653, 659, 660, 666, 667, 922, 923, 1062, 1109, 1141, 1256, 1338, 1339, 1676, 2041, 2042, 2043, 2055, 2664, 2665, 2666, 2671, 2673, 2674, 2675, 2676, 2679, 2680, 2683, 2695, 2696, 2697, 2698, 2699, 2708, 2709, 2710, 2712, 2713, 2726, 2728, 2730, 3320, 3322, 3323, 3324, 3325, 3448, 3449, 3458, 3459, 3496, 3498, 3499, 3802, 4532, 6286, 6287, 6288, 6376
\exp_not:n 103, 144, 654, 661, 668, 1335, 1336, 1337, 1468, 1469, 1677, 1678, 1679, 1704, 1760, 1761, 1789, 2036, 2056, 2205, 3467, 3790, 6055, 6190, 6261, 6262, 6304, 6376, 6377, 6378, 6433, 6434
\expandafter 401, 402, 404, 405, 413, 417, 3834, 5493, 5494, 5518, 5519, 5520, 5521, 5522, 5527, 5531, 5532
experiment 2353
experiment/CJKecglue *13*, 2353
experiment/font-size-system *9*, 159
\ExplSyntaxOff 849, 935, 939, 1928, 2245, 3580, 3690, 3830, 5902, 6239
\ExplSyntaxOn 886, 3751, 5813

F

\familydefault 2039
fancyhdr *36*, 311
\fangsong 5452
\fi 402, 405, 406, 414, 415, 416, 418, 1902, 1909, 2834, 2839, 2845, 2854, 2863, 2866, 2881, 2886, 2896, 2907, 2922, 2927, 2938, 2964, 2988, 2996, 2997, 3000, 3004, 3014, 3032, 3038, 3044, 3056, 3066, 3070, 3079, 3080, 3090, 3095, 3109, 3113, 3119, 3125, 3132, 3134, 3156, 3157, 3175, 3189, 3192, 3213, 3222, 3241, 3242, 3606, 3609, 3614, 3624, 3625, 3631, 3638, 3641, 3644, 3647, 3656, 3659, 3664, 3674, 3675, 3681, 3703, 3722, 3741, 3891, 3919, 5523, 5524, 5533, 5571, 5595, 5652, 5693, 5725, 5749, 5806
fi commands:
\fi: 2028, 2175
\figurename 2599
figurename *14*
file commands:
\g_file_curr_name_str 834
\file_get_full_name:nNTF 474
\file_if_exist:nTF 28, 4288, 4511, 4526, 5017, 5059, 5380, 5383
\file_if_exist_input:nTF 74
\file_if_exist_p:n *153*, 5863, 5865
\file_input:n *157*, 75, 77, 536, 4374, 4375, 6073
\file_input_stop: 6176
\fmtversion 6069
fntef *36*, *38*, 292
\font 4596, 4597, 4598, 4599, 4600, 4601
fontset *7*, 227, 4545
\fontsize 4213
fontspec commands:
\fontspec_font_if_exist:nTF 5153, 5170, 5177, 5183, 5188, 5190
\fontspec_gset_family:Nnn 1336
\fontspec_set_family:Nnn 1713
fontspec internal commands:
__fontspec_fontname_wrap:n *69*, 1278
\footnotesep 4461
\footnotesize 4398, 4437, 4461
fp commands:
\fp_eval:n 2500, 5892
\fp_if_nan:nTF 4174, 4200, 4458
\fp_set:Nn 4175, 4201
\fp_use:N 2431, 4460
\c_nan_fp 178
\c_zero_fp 2431

G

GBK *6*, *38*, 215
\gdef . 409, 415, 3346, 3347, 3348, 3349, 3352, 3353, 3354, 3355
\global 3008, 3227, 3230
\globaljfont 1090
\glueexpr 3697, 3708, 3716, 3727, 3735, 3746
group commands:
\group_begin: . 671, 892, 965, 1037, 1047, 1074, 1102, 1140, 1162, 1208, 1283, 1334, 1708, 1786, 1804, 1821, 2035, 2212, 2300, 2670, 2678, 2742, 3270, 3370, 3516, 3575, 4235, 4461, 5839, 6184, 6205, 6263, 6271, 6280, 6335
\group_end: 676, 897, 1034, 1037, 1053, 1078, 1115, 1144, 1166, 1212, 1298, 1341, 1714, 1790, 1817, 1828, 2047, 2243, 2317, 2681, 2685, 2764, 3274, 3387, 3539, 3579, 4322, 4461, 5855, 6184, 6205, 6283, 6330, 6358, 6392, 6416
\group_insert_after:N 1112, 1113
\gtfamily 1929, 1934

H

\hbox 1902

hbox commands:
\hbox_set:Nn 2144, 3263, 3568
heading 10, 265
\heiti 5452
hook commands:
\hook_gput_code:nnn 6165
\hskip 3165, 3181, 3203, 3210, 3236, 3638, 3641
\hspace 2876, 2917, 3295, 3303, 3488
\Huge 2941, 2978, 3083, 3102, 3895, 3926, 4415, 4454
\huge 2899, 2937, 2957, 3078, 3877, 3894, 3898, 3924, 3932, 4414, 4453
hyperref 36, 316
\hypersetup 321, 322, 2064
\hyphenchar 4596, 4597, 4598, 4599, 4600, 4601

I

\if 408
if commands:
\if_bool:N 1993
\if_false: 96
\if_meaning:w 2174
\if_true: 95
\ifctexpdf 94
\ifdim 3117, 3160, 3198, 3217
\ifmmode 1902
\ifnum 406, 2872, 2894, 2913, 2936, 3017, 3076, 3143, 3173, 3187, 3606, 3609, 3614, 3621, 3622, 3631, 3638, 3641, 3644, 3647, 3656, 3659, 3664, 3671, 3672, 3681, 5523
\ifodd 403, 415, 2839, 2841, 2866, 2873, 2907, 2914, 2964, 2988, 3010, 3021, 3070, 3090, 3095, 3109, 3121, 3132, 3148, 3161, 3199, 3218, 3222, 3701, 3720, 3739
\ifx 401, 404, 405, 5521, 5522, 5527
\ifzhmappdf . . 5519, 5550, 5574, 5598, 5655, 5696, 5728, 5752
\ignorespaces 3243
\immediate 5530
indent 35, 197
\indexname 2602
indexname 14
\input 5812, 5905
\insertcontinuationtext 2611, 2620, 2627
\insertpart 3705, 3706
\insertromanpartnumber 3695, 3864
\insertsection 3724, 3725
\insertsectionnumber 3714, 3962
\insertsubsection 3743, 3744
\insertsubsectionnumber 3733, 3991
int commands:
\int_case:nn 381, 4372
\int_compare:nNnTF 546, 910, 2474, 2517, 4050, 4066, 4082, 4172, 4198, 5068
\int_const:Nn 1270, 1872, 1873
\int_div_truncate:nn 92, 553
\int_eval:n 1864, 1865, 1868, 1869, 3766
\int_from_hex:n 545
\int_gincr:N 1426, 1727
\int_gset:Nn 83, 153, 167, 168, 169, 270, 272, 340, 344, 4173, 4199
\int_if_odd:nTF 6089
\int_incr:N 3317
\int_mod:nn 555
\int_new:N 56, 82, 152, 1730
\int_set:Nn 535, 544, 2471, 2512, 2769
\int_sub:Nn 549, 2518
\int_to_Hex:n 547, 552, 554
\int_use:N 1428, 1728, 3321, 5848
\int_value:w 593, 604, 605, 620, 621, 622, 637, 638, 639, 640, 674, 680
\int_zero:N 3314
\c_max_char_int 1873
\c_zero_int 2474
\interlinepenalty . 2891, 2933, 2953, 2974, 3098, 3166, 3204
ior commands:
\ior_close:N 5036
\ior_map_break: 5029
\ior_map_inline:Nn 5021
\ior_new:N 5015
\ior_open:Nn 5019
iow commands:
\iow_close:N 5833
\iow_indent:n 6060
\iow_new:N 5835
\iow_now:Nn 5850
\iow_open:Nn 5830
\itemsep 4395, 4406, 4434, 4445
\itshape 1929, 1930, 1934, 1935

J

\jfam 1200, 1271, 1926

K

\kaishu 5452
\kanjifamily 1920, 2023, 2024, 2025
\kanjifamilydefault 2026
\kern 1910
kernel internal commands:
\l__kernel_expl_bool 6076, 6081, 6082, 6090, 6091
keys commands:
\keys_define:nn 61, 64, 1245, 1275, 1614, 1724, 1878, 2355
\keys_if_exist:nnTF 1876
\l_keys_key_str 129, 131, 147
\keys_set:nn 66, 68
\keys_set_known:nn 4130
\keys_set_known:nnN 1287

L

\labelformat 3802, 3838
\language 1906
\languagealias 4159, 4164, 4184, 4189
\LARGE 4413, 4452
\Large 2895, 3875, 3881, 3949, 3953, 4412, 4451
\large 3974, 4411, 4450
\lastbox 3231

\leavevmode 1902, 2834, 3113
\leftmargin 4392, 4403, 4431, 4442
\leftmargini 4392, 4403, 4431, 4442
legacy commands:
\legacy_if:nTF . 1156, 3371, 3514, 3603, 3628, 3653, 3678
\let 411, 414, 1904, 3146, 3155, 3508, 3510, 4385, 4424, 5514, 5520
\linespread 4460
linespread *12*, 177
linestretch *12*, 2436
\linewidth 2472, 2480, 2511
\lishu 5452
\listfigurename 2597
listfigurename *13*
\listtablename 2598
listtablename *14*
\LoadClass 124
\long 5528
\ltjdefcharrange . 850, 851, 852, 854, 859, 860, 864, 867, 868
LTJFONTUID 1724
\ltjgetparameter 2149, 2178, 2278, 2292
\ltjlineendcomment 848
\ltjsetkanjiskip 1913
\ltjsetmathletter 1210
\ltjsetparameter 870, 873, 879, 1897
\ltjsetxkanjiskip 1915
lua commands:
\lua_now:n 967, 2302, 5041, 5201

M

\makeatletter 935, 939
\MAKESPA 5836, 5906
\markboth 2807, 2901, 2930
\mathgroup 1200
\mcfamily 1934
\mddefault 1257, 1261, 1267, 1602, 1607
mode commands:
\mode_if_math:TF 1194
\mode_leave_vertical: *55*, 685, 691, 697, 703
msg commands:
\msg_critical:nnn 33, 834, 4485
\msg_error:nn 4565
\msg_error:nnn 22, 825, 844, 847, 1508, 2584, 4214, 4294, 4368
\msg_error:nnnn 4531, 4560, 6110
\msg_fatal:nn 5818
\g_msg_module_name_prop 6, 7, 8, 9, 10, 11
\g_msg_module_type_prop 5
\msg_new:nnn 128, 130, 132, 146, 148, 371, 828, 1328, 1394, 1476, 1501, 2615, 4104, 4486, 4539, 4574, 5011, 5816, 6056, 6123, 6137, 6151, 6247
\msg_new:nnnn ... 14, 25, 88, 2586, 4216, 4323, 4541, 4568
\msg_redirect_name:nnn 837, 839
\msg_warning:nn 220, 376, 1471, 2338, 2349, 2372, 2637, 5064
\msg_warning:nnn 135, 1389, 4123, 4552, 6062, 6150, 6246
\msg_warning:nnnn 143, 1324, 4521, 6119, 6136

N

\newCJKfontfamily 1495
\NewDocumentCommand 3, 1355, 1490, 1495, 1503, 1518, 1524, 1530, 1537, 1543, 1549, 1555, 1752, 1852, 1919, 2379, 2382, 2392, 2419, 2496, 2533, 2540, 2563, 2565, 2732, 4208, 4465, 5453, 5454, 5455, 5456, 5458, 5459, 5461, 5473, 5474
\newfontfeature 1272, 1273, 1274
\newpage 2990, 2995
\ngostype 24
\nobreak 2895, 2904, 2961, 3078, 3083, 3087, 3102, 3106, 3219, 3876, 3925
\nobreakspace 2895, 2937
nocap *36*, 288
\noexpand 5512, 5514, 5515, 5516
nofonts *36*
noindent *35*
nopunct *36*
\normalem 2118
\normalfont 1534, 1541, 1547, 1585, 2007, 2029, 2767, 2891, 2933, 2953, 2974, 3073, 3098, 3325
\normalsize 4008, 4025, 4039, 4379, 4418, 4456, 4464
nospace *36*
nozhmap *36*
\null 2864, 2993
\numberline 3027, 3174, 3188, 3303, 3311, 3499, 3575

O

\onecolumn 2859

P

\pagestyle 4150
\par 2794, 2835, 2895, 2901, 2902, 2937, 2941, 2943, 2957, 2959, 2978, 2980, 3078, 3083, 3085, 3102, 3104, 3114, 3169, 3207, 3219, 3696, 3697, 3705, 3708, 3715, 3716, 3724, 3727, 3734, 3735, 3743, 3746, 3854, 3876, 3887, 3905, 3913, 3925, 3965, 3967, 3999, 4001
paragraph/afterskip *37*
paragraph/beforeskip *37*
paragraph/numbering *17*
paragraph/runin *23*
\parindent 2492, 2493, 2515, 2536, 2538, 2543, 2544, 2891, 2953, 3073, 3098, 4084
\parsep 4394, 4395, 4405, 4406, 4433, 4434, 4444, 4445
\part 2832
part/beforeskip *37*
part/fixbeforeskip *37*
part/numbering *17*
part/pagestyle *23*, 2742
\partmark 2806, 2889, 2931
\partname 2895, 2937, 3695, 3859
\PassOptionsToClass 121
\PassOptionsToPackage 506, 2068, 2074, 2546, 6064
patch/cleveref *35*, 3804
\patchcmd 1910

pdf commands:
\pdf_object_ref_last: 484
\pdf_object_unnamed_write:nn 477
\pdffontattr 564, 567
\pdfmapline 5599, 5600, 5601, 5602, 5603, 5604, 5605, 5606, 5607, 5608, 5609, 5610, 5611, 5612, 5613, 5614, 5615, 5616, 5617, 5618, 5619, 5620, 5621, 5622, 5623, 5624, 5656, 5657, 5658, 5659, 5660, 5661, 5662, 5663, 5664, 5665, 5666, 5667, 5668, 5669, 5670, 5671, 5672, 5673, 5753, 5754, 5755, 5756, 5757, 5758, 5759, 5760, 5761, 5762, 5763, 5764, 5765, 5766, 5767, 5768, 5769, 5770, 5771, 5772, 5773, 5774, 5775, 5776, 5777, 5778
\pdfnobuiltintounicode 482, 566
\pdfoutput 406, 5523
\pdfstringdefDisableCommands 4468
\pingfang 5452
prg commands:
\prg_do_nothing: 441, 487, 1204, 1702, 1787, 2207, 2325, 3287, 6368
\prg_generate_conditional_variant:Nnn 1381
\prg_new_conditional:Npnn 1118, 2171
\prg_new_protected_conditional:Nnn 5857
\prg_new_protected_conditional:Npnn 1368, 6338
\prg_return_false: 1121, 1378, 2175, 2179, 2183, 5877, 6348
\prg_return_true: 1120, 1371, 1376, 2175, 2179, 2183, 5877, 6353
\primitive 5965
\ProcessKeyOptions 363, 364
\ProcessKeysOptions 367
\proofname 2589, 2604
proofname 14
prop commands:
\prop_clear:N 1285, 4236
\prop_const_from_keyval:Nn 3769, 4318
\prop_get:NnN 1463
\prop_get:NnNTF 1233, 1370, 1422, 1460, 1695, 1741, 3764, 4212, 4339, 4366
\prop_gpop:NnNTF 1313, 1317, 1822
\prop_gput:Nnn 5, 6, 7, 8, 9, 10, 11, 1291, 1292, 1338, 1450, 1810, 1846
\prop_gremove:Nn 1322
\prop_if_empty:NTF 1384, 1566, 1646
\prop_map_break:n 1580
\prop_map_inline:Nn 1578, 1654
\prop_new:N 1237, 1308, 1309, 1310, 1455, 1636, 1830, 1851, 4223
\prop_put:Nnn 1633, 4227
\prop_remove:Nn 1632
\prop_set_from_keyval:Nn 1238
\prop_to_keyval:N 4319
\protect 596, 610, 627, 2820, 2826, 3027, 3048, 3049, 3174, 3188, 3303, 3311, 3498, 3521, 3523, 3527, 3529, 3533, 3535, 3543, 3548, 3560, 3561, 6299, 6300
\ProvideDocumentCommand 2806
\ProvidesFile . 5526, 5535, 5537, 5539, 5541, 5543, 5545, 5547
\providetranslation 5970, 5971, 5972, 5973, 5974, 5975, 5976, 5977, 5978, 5979, 5980, 5981, 5982, 5983, 5984, 5985, 5986, 5987, 5988, 5989, 5990, 5991, 5992, 5993, 5994, 5995, 5996, 5997, 5998, 5999, 6000, 6001, 6002, 6003, 6004, 6005, 6006, 6007, 6008, 6009, 6010, 6011, 6012, 6013, 6014, 6015, 6016, 6017, 6018, 6019, 6020, 6021, 6022, 6023, 6024, 6025, 6026, 6027, 6028, 6029, 6030, 6031, 6032, 6033, 6034, 6035, 6036, 6037, 6038, 6039, 6040, 6041, 6042, 6043, 6044, 6045, 6046, 6047, 6048, 6049
punct 11, 36, 254, 2341
\punctstyle 575, 2346

Q

\quad . 3139, 3606, 3607, 3609, 3610, 3614, 3615, 3627, 3632, 3639, 3642, 3645, 3648, 3656, 3657, 3659, 3660, 3664, 3665, 3677, 3682, 3882, 3933, 3942, 3975, 4009, 4026, 4040
quark commands:
\q_mark 1655, 1663, 6342, 6345
\q_stop 105, 108, 1155, 1158, 1161, 1164, 5058, 5315, 6307, 6312, 6321, 6342, 6345
\quotation 4478

R

\raggedright 2891, 2953, 3073, 3098, 3874, 3923
\refname 2605, 2610, 2617, 2626, 2634, 2635
refname 14
\refstepcounter . . . 2875, 2916, 3023, 3150, 3784, 3841, 3842
regex commands:
\regex_replace_once:nnN 5026
\relax 401, 404, 405, 408, 1902, 2913, 2936, 3115, 3151, 3159, 3165, 3181, 3197, 3210, 3216, 3235, 3638, 3641, 3697, 3708, 3716, 3727, 3735, 3746, 5485, 5490, 5493, 5521, 5522, 5527
\renewcommand 2833, 2849, 3003
\RenewDocumentCommand 3842
\RequirePackage 4, 36, 37, 51, 297, 303, 314, 324, 366, 509, 513, 514, 515, 818, 842, 845, 959, 964, 1990, 2079, 2089, 2549, 4203
\RequirePackageWithOptions 6065
resetalternatefont 33, 1793
reverse commands:
\reverse_if:N 1993
\rmdefault 2041
\rmfamily 2004

S

scan commands:
\scan_stop: 78, 105, 923, 1090, 1200, 1788, 2461, 2472, 5841, 5842, 6186
scheme 11, 276
\scriptsize 4409, 4448
\secdef 2846, 2868, 3015
secnumdepth 17, 3753
section/afterskip 37
section/beforeskip 37
section/numbering 17

section/runin 23
\sectionname 3714, 3959
\selectfont 1362, 1813, 1826, 1920, 2004, 2005, 2006, 2242, 2249, 2271, 2421, 4213
seq commands:
\seq_clear:N 4237
\seq_const_from_clist:Nn 448, 2646, 2649, 3360, 4320, 5820
\seq_gpop:NN 6088
\seq_gpop:NNTF 1066
\seq_gpush:Nn 1061, 6080
\seq_gput_right:Nn 1388
\seq_if_in:NnTF 454, 1386
\seq_map_function:NN 3431, 3564
\seq_map_inline:Nn 2759, 3315, 3363, 3374, 3384, 3407, 3490, 3795, 5845
\seq_new:N 1069, 1393, 4224, 6094
\seq_put_right:Nn 4232
\seq_use:Nn 4321
\seq_use:Nnnn 4221
\setbox 3231
\setCJKfamilyfont 1407, 1495, 4763, 4765, 4767, 4769, 4805, 4806, 4807, 4808, 4827, 4829, 4831, 4833, 4876, 4878, 4880, 4882, 4895, 4896, 4897, 4898, 4899, 4900, 4941, 4943, 4944, 4945, 4946, 4947, 4960, 4962, 4964, 4965, 5001, 5003, 5005, 5006, 5096, 5097, 5098, 5099, 5100, 5101, 5144, 5145, 5146, 5147, 5166, 5168, 5173, 5180, 5185, 5189, 5192, 5264, 5266, 5278, 5287, 5292, 5302, 5307, 5330, 5332, 5334, 5372, 5374, 5376, 5397, 5398, 5399, 5400, 5401, 5403, 5404, 5443, 5444, 5445, 5446, 5447, 5448, 5449
\setCJKmainfont 1403, 1530, 4753, 4801, 4817, 4866, 4891, 4937, 4956, 4997, 5088, 5140, 5155, 5163, 5251, 5255, 5324, 5367, 5392, 5440
\setCJKmathfont 1530
\setCJKmonofont 1405, 1530, 4761, 4804, 4825, 4874, 4894, 4940, 4959, 5000, 5095, 5143, 5179, 5182, 5283, 5285, 5328, 5371, 5396, 5442
\setCJKromanfont 1536, 1562
\setCJKsansfont 1404, 1530, 4759, 4803, 4823, 4872, 4893, 4939, 4958, 4999, 5094, 5142, 5172, 5176, 5269, 5271, 5326, 5369, 5394, 5441
\setcounter 3765, 3766
\SetSymbolFont 1268, 1606, 1610, 1925
\sfdefault 2042
\sffamily 2005
\shapedefault 1257, 1261, 1263, 1267, 1269, 1602, 1604, 1607, 1611
skip commands:
\skip_horizontal:N 2159, 3273
\skip_horizontal:n 2359
\skip_if_eq:nnTF 2178, 2182, 2277, 2817, 2823
\skip_new:N 2170, 2290, 2798
\skip_set:Nn 2149, 2152, 2281, 2291, 2365, 2462, 2476, 2520, 2528, 2800, 2816, 2822, 3271
\skip_set_eq:NN 2168, 2283, 2367
\skip_sub:Nn 2789, 2796
\skip_use:N 2820, 2826
\skip_zero:N 2485
\c_zero_skip 2701, 2702, 2750, 2751, 2817, 2823
\small 4387, 4426
\songti 5452
\space 406, 3024, 3078, 3859, 3921, 3959, 3990, 4091
space 11, 36, 259, 2320
\special 1956, 1957, 1960, 1961, 1969, 1970, 1973, 1974, 2087, 2111, 5553, 5554, 5555, 5556, 5557, 5558, 5559, 5560, 5561, 5562, 5563, 5564, 5565, 5566, 5567, 5568, 5569, 5570, 5577, 5578, 5579, 5580, 5581, 5582, 5583, 5584, 5585, 5586, 5587, 5588, 5589, 5590, 5591, 5592, 5593, 5594, 5626, 5627, 5628, 5629, 5630, 5631, 5632, 5633, 5634, 5635, 5636, 5637, 5638, 5639, 5640, 5641, 5642, 5643, 5644, 5645, 5646, 5647, 5648, 5649, 5650, 5651, 5675, 5676, 5677, 5678, 5679, 5680, 5681, 5682, 5683, 5684, 5685, 5686, 5687, 5688, 5689, 5690, 5691, 5692, 5699, 5700, 5701, 5702, 5703, 5704, 5705, 5706, 5707, 5708, 5709, 5710, 5711, 5712, 5713, 5714, 5715, 5716, 5717, 5718, 5719, 5720, 5721, 5722, 5723, 5724, 5731, 5732, 5733, 5734, 5735, 5736, 5737, 5738, 5739, 5740, 5741, 5742, 5743, 5744, 5745, 5746, 5747, 5748, 5780, 5781, 5782, 5783, 5784, 5785, 5786, 5787, 5788, 5789, 5790, 5791, 5792, 5793, 5794, 5795, 5796, 5797, 5798, 5799, 5800, 5801, 5802, 5803, 5804, 5805
\SplitArgument 1753, 1853, 2733
str commands:
\c_backslash_str 6326
\c_colon_str 1187, 1188, 1189
\c_dollar_str 5380
\c_left_brace_str 6327
\str_case:nnTF 2039, 4238, 6297, 6312
\str_case_e:nnTF 1401, 1412
\str_const:Nn 23, 30
\str_if_empty:NTF 6383, 6419, 6428
\str_if_eq:nnTF 512, 574, 581, 647, 1570, 2075, 2084, 4155, 4181, 4195, 4551, 4554, 4571, 4576, 6303
\str_if_eq_p:nn 4518, 4519
\str_lowercase:n 475
\str_new:N 6332, 6333, 6334
\str_set:Nn 6350, 6351, 6352
\strutbox 4462
sub3section 11, 268
sub4section 11, 268
subparagraph/afterskip 37
subparagraph/beforeskip 37
subparagraph/numbering 17
subparagraph/runin 23
subsection/afterskip 37
subsection/beforeskip 37
subsection/numbering 17
subsection/runin 23
\subsectionname 3733, 3990
subsubsection/afterskip 37
subsubsection/beforeskip 37

subsubsection/numbering 17
subsubsection/runin 23
\symmincho 1926
sys commands:
\c_sys_engine_str 24
\sys_get_shell:nnN 5869, 5871
\sys_if_engine_luatex:TF 5013, 5151
\sys_if_engine_pdftex:TF . . . 217, 300, 375, 4490, 4496
\sys_if_engine_pdftex_p: 5464
\sys_if_engine_uptex:TF 4492, 4498
\sys_if_engine_uptex_p: 5465
\sys_if_engine_xetex:TF 294, 5814
\sys_if_output_pdf:TF 94, 490, 2078, 4497
\sys_if_platform_windows:TF 4502

T

\tablename 2600
tablename 14
\TeX 4466
TEX and LATEX 2ε commands:
\@@ifdefinable 900
\@@italiccorr 1911
\@@par 3166, 3943, 3976, 4010, 4054, 4070
\@afterheading 2909, 2966, 3055, 3065, 3224
\@afterindentfalse 2844, 3009, 3013, 3118, 3124
\@afterindenttrue 2842, 3011, 3116, 3122
\@Alph 4087, 4092
\@break@tfor 414
\@chapapp 3024, 3078, 3623, 3644, 3673
\@chapter 3015, 3016
\@chinese 2551, 2560
\@classoptionslist 49, 409, 415, 417
\@clubpenalty 3239
\@currentHref 3396
\@currentlabel 3786, 3790
\@currext 160, 6055, 6213, 6219
\@currname 160, 6055, 6112, 6214, 6220
\@dblarg 3137
\@defaultfamilyhook 2007
\@empty 117, 3146, 3155
\@endpart 2945, 2982, 2984
\@EverySelectfont@Init 2210, 2225, 2237
\@firstofone 63
\@firstoftwo 3560
\@gobble 63
\@hangfrom 3165, 3203
\@ifclassloaded 4100
\@ifdefinable 900
\@ifl@t@r 6069
\@ifpackagelater 21, 824, 843, 846, 872, 2125
\@ifpackageloaded 833, 954, 2061, 2072, 2099, 3470, 3586, 3813, 6109, 6145, 6158
\@ifpackagewith 3415, 3424, 3475, 3650, 3815
\@ifstar 1907, 3135
\@listI 4385, 4424
\@listi 4385, 4392, 4403, 4424, 4431, 4442
\@M 2891, 2933, 2953, 2974, 3098, 3166, 3204, 3232
\@makechapterhead 3052, 3054, 3067
\@makeother 1904
\@makeschapterhead 3062, 3064, 3092
\@minus 3944, 3977, 4011, 4027, 4041, 4381, 4383, 4389, 4391, 4393, 4394, 4400, 4402, 4404, 4405, 4420, 4422, 4428, 4430, 4432, 4433, 4439, 4441, 4443, 4444
\@namedef 1898, 3521, 3523, 3527, 3529, 3533, 3535, 3543, 3548
\@nameuse 3521, 3523, 3533, 3535, 3543
\@ne 411, 3468, 3609, 3641, 3659
\@nil 1141
\@nobreakfalse 3226
\@noligs 1905
\@noskipsecfalse 3230
\@noskipsectrue 3227
\@onlypreamble 43, 445, 523, 1558, 1559, 1560, 1561, 1562, 1563, 1985, 1986, 1987, 1988, 4544
\@part 2846, 2868, 2870
\@pkgextension 6121, 6130, 6131, 6213, 6219
\@plus 3888, 3889, 3944, 3945, 3977, 3978, 4011, 4012, 4027, 4041, 4055, 4071, 4381, 4382, 4383, 4389, 4390, 4391, 4393, 4394, 4400, 4401, 4402, 4404, 4405, 4420, 4421, 4422, 4428, 4429, 4430, 4432, 4433, 4439, 4440, 4441, 4443, 4444
\@popfilename 160, 6216
\@pushfilename 160
\@reset@ptions 160, 6210
\@rmfamilyhook 2004
\@schapter 3015, 3057
\@seccntformat 3138, 3151
\@secondoftwo 3561
\@secpenalty 3129, 3947, 3980, 4014, 4028, 4042
\@sect 3137, 3142
\@setfontsize 4371
\@sffamilyhook 2005
\@spart 2846, 2868, 2947
\@ssect 3136, 3194
\@startsection 107, 110, 119, 3112, 3320
\@sverb 1907
\@svsec 3146, 3151, 3155, 3165, 3167, 3181, 3182
\@svsechd 3179, 3210, 3211, 3233
\@tempdima 3569, 3571
\@tempskipa 3115, 3117, 3118, 3129, 3159, 3160, 3197, 3198, 3216, 3217, 3220, 3235, 3236
\@tempswafalse 2862
\@tempswatrue 2860
\@text@composite@x 60, 950
\@tfor 412
\@topnewpage 3052, 3062
\@topnum 3008
\@ttfamilyhook 2006
\@verb 1907
\@xsect 3193, 3214, 3215
\@zhdig 2559
\@zhnum 2551, 2558

\abstractname *14*
\add@unicode@accent *60*, 951
\addCJKfontfeature *78*
\AfterEndPreamble *159*
\algorithmname *14*
\appendixname *14*, *28*
\AtBeginDocument *43*, *53*
\AtBeginDvi *50*
\AtBeginShipoutFirst *50*
\AtEndOfClass *43*
\AtEndOfPackage *43*
\AtEndOfPackageFile* *159*
\AtEndPreamble *159*
\baselineskip *12*, *89*, *92*
\bfseries *57*
\bibname *14*
\c@chapter 4092
\c@secnumdepth . . 2872, 2894, 2913, 2936, 3017, 3076, 3143, 3173, 3187, 3606, 3609, 3614, 3621, 3622, 3631, 3638, 3641, 3644, 3647, 3656, 3659, 3664, 3671, 3672, 3681
\c@section 4087
\caption *37*
\catcode *59*, *147*, *160*
\catcodetable *59*
\ccwd *12*, *25*, *31*, *36*, *43*, *86*, *87*, *91*–*93*
\chapter *10*, *18*, *23*, *28*, *29*, *99*, *106*, *123*
\chaptermark *15*
\chaptername *18*
\char *61*
\chardef@text@cmd *60*, 948
\CheckCommand *87*
\Chinese *37*
\chinese *31*, *37*, *94*
\CJK@@ignorespaces *54*, 591
\CJK@addcmap 561
\CJK@encoding *68*, *70*, *77*, 1044, 1048, 1135, 1252, 1306, 1601, 1604, 1606, 1610, 1750, 1760, 1761, 1779, 1780
\CJK@envStart *53*, 586
\CJK@family *63*, 1035, 1049, 1103, 1135, 1359, 1812, 1822, 1824, 2271, 2272
\CJK@ignorespaces *57*, 687, 693, 699, 705, 756, 760
\CJK@input 532
\CJK@loadBinding 582, 583
\CJK@makeActive *53*
\CJK@plane *52*, *88*, 462, 463, 2260
\CJK@surr *52*, 540
\CJK@upperReset *53*
\CJK@X *54*, 592
\CJK@XX 592
\CJK@XXX 592
\CJK@XXXX 592
\CJKfamily *70*, *81*
\CJKfamilydefault *53*, *74*, *83*, *84*
\CJKglue *92*, *129*
\CJKhook *57*
\CJKrmdefault *52*
\CJKsymbol *88*
\CJKunderdotbasesep *36*
\CJKunderline *36*
\clearpage *53*
\contentsname *13*
\CS *64*
\CTEX@add 411, 414, 415
\CTEX@add@unicode@accent 919, 951
\CTEX@addloflotskip 2813, 3050
\CTEX@addtocline . . . 2830, 2888, 2929, 3046, 3177, 3191
\CTEX@afterindent 3121, 3281, 3288
\CTEX@aftertitle 3207, 3212, 3280, 3287
\CTEX@alchar 888, 911
\CTEX@appendix@number 3335, 3347, 3353
\CTEX@appendix@numbering 3336, 3349, 3355
\CTEX@beginallalchar 888, 915, 921, 936, 960
\CTEX@chapter@afterindent 3010
\CTEX@chapter@aftername 3082
\CTEX@chapter@afterskip 3089, 3108
\CTEX@chapter@aftertitle 3085, 3104
\CTEX@chapter@beforeskip 3069, 3094
\CTEX@chapter@break 3006
\CTEX@chapter@fixskip 3070, 3090, 3095, 3109
\CTEX@chapter@format 3075, 3100
\CTEX@chapter@lofskip 2748
\CTEX@chapter@lotskip 2749
\CTEX@chapter@numbering 3021, 3355
\CTEX@chapter@pagestyle 2747
\CTEX@chapter@titleformat 3084, 3103
\CTEX@chapter@tocline 3292
\CTEX@chaptername 3082
\CTEX@char@n 593, 683, 718, 749
\CTEX@char@nn *56*, 602, 674, 683, 707, 723
\CTEX@char@nn@n *56*, 707
\CTEX@char@nnn 618, 683, 708, 724
\CTEX@char@nnn@n 707
\CTEX@char@nnnn 635, 683, 709, 725
\CTEX@char@nnnn@n 707
\CTEX@chardef@text@cmd 898, 948
\CTEX@defaultfamilyhook 2017
\CTEX@document@left@hook 6183, 6189
\CTEX@document@right@hook 6185, 6191
\CTEX@endallalchar 888, 917, 924, 940, 961
\CTEX@Family@CMap 459, 496, 501
\CTEX@fixheadingskip 2787, 2792, 2839, 2907, 2964, 2988, 3090, 3109, 3132, 3222
\CTEX@fixskip 3132, 3222, 3282, 3289
\CTEX@fixtopskip 2784, 2866, 3070, 3095
\CTEX@gettitle 2887, 2928, 2951, 2972, 3045, 3060, 3158, 3196, 3402
\CTEX@hang 3251, 3283, 3290
\CTEX@heading@format@initial 2765, 2892, 2934, 2954, 2975, 3074, 3099, 3163, 3201, 3698, 3717, 3736
\CTEX@heading@glue 3237, 3268
\CTEX@headinghang 2897, 2939, 2956, 2977, 3081, 3101, 3244

\CTEX@headingskip 2801, 2840, 2867, 2908, 2965, 2989, 3071, 3091, 3096, 3110, 3133, 3223
\CTEX@hyperheadinghook 3368, 3400
\CTEX@ifnamefalse . . . 2808, 2878, 2883, 2919, 2924, 2949, 2970, 3029, 3035, 3041, 3058, 3144, 3153, 3195, 3514
\CTEX@ifnametrue . . . 2808, 2874, 2915, 3022, 3149, 3514
\CTEX@indentbox 2782, 3182, 3211, 3700, 3719, 3738
\CTEX@makeanchor 2879, 2884, 2920, 2925, 3030, 3154, 3358, 3366, 3379, 3385, 3391
\CTEX@makeanchor@chapter 3036, 3042
\CTEX@makeanchor@schapter 3059
\CTEX@makeanchor@sect 3145
\CTEX@makeanchor@spart 2950, 2971
\CTEX@makeanchor@ssect 3136
\CTEX@part@afterindent 2841
\CTEX@part@aftername 2898, 2940, 3702
\CTEX@part@afterskip 2906, 2963, 2987, 3708
\CTEX@part@aftertitle . . . 2902, 2943, 2959, 2980, 3706
\CTEX@part@beforeskip 2838, 2865, 3697
\CTEX@part@break 2836, 2855
\CTEX@part@fixskip 2839, 2866, 2907, 2964, 2988
\CTEX@part@format 2893, 2935, 2955, 2976, 3699
\CTEX@part@indent 3700
\CTEX@part@numbering 2873, 2914, 3701
\CTEX@part@pagestyle 2746
\CTEX@part@titleformat . . 2900, 2942, 2958, 2979, 3706
\CTEX@part@tocline 3292
\CTEX@partname 2898, 2940, 3702
\CTEX@patch@text@cmd 926, 952
\CTEX@patch@tunec 946
\CTEX@popfilename@hook 6216
\CTEX@postappendix 3340, 3348, 3354
\CTEX@postchapter 3354
\CTEX@postsection 3348
\CTEX@preappendix 3339, 3346, 3352
\CTEX@prechapter 3352
\CTEX@presection 3346
\CTEX@reset@ptions@hook 6210
\CTEX@rmfamilyhook *83*, 1994, 2014, 2023
\CTEX@runin 3161, 3199, 3218, 3284, 3291
\CTEX@save@appendix 3341, 3344
\CTEX@save@refstepcounter 3841, 3844
\CTEX@section@aftername 3721
\CTEX@section@afterskip 3727
\CTEX@section@aftertitle 3725
\CTEX@section@beforeskip 3716
\CTEX@section@format 3718
\CTEX@section@indent 3719
\CTEX@section@numbering 3349, 3720
\CTEX@section@titleformat 3725
\CTEX@sectionhang 3167, 3205, 3244
\CTEX@sectionname 3721
\CTEX@selectfont@hook 2202, 2260, 2263
\CTEX@selectfont@save 2216, 2231, 2235, 2238
\CTEX@setcurrentlabel@n 3784, 3845
\CTEX@setheadingskip 2799, 2838, 2865, 2906, 2963, 2987, 3069, 3089, 3094, 3108, 3131, 3221
\CTEX@setthispagestyle 2802, 2857, 3007
\CTEX@sffamilyhook 2015, 2024
\CTEX@subsection@aftername 3740
\CTEX@subsection@afterskip 3746
\CTEX@subsection@aftertitle 3744
\CTEX@subsection@beforeskip 3735
\CTEX@subsection@format 3737
\CTEX@subsection@indent 3738
\CTEX@subsection@numbering 3739
\CTEX@subsection@titleformat 3744
\CTEX@subsectionname 3740
\CTEX@text@composite@x 913, 950
\CTEX@textcmd 933, 934, 938, 941
\CTEX@thechapter 3353
\CTEX@thesection 3347
\CTEX@titleformat@n *107*, 3206, 3212, 3279, 3286
\CTEX@titlepslabel@clear 3530, 3540
\CTEX@titlepslabel@set 3524, 3536, 3540
\CTEX@toc@width@n 3566, 3581, 3593
\CTEX@todayold 2567, 2572
\CTEX@ttfamilyhook 2016, 2025
\CTEX@update@sectionformat@n 3120, 3277
\CTEX@updatettlifname 3511, 3513
\CTEX@verbatim@font@hook 1892, 1896
\ctex@zhmap@endinput 5808
\CTEXdigits *31*
\CTEXifname *27*, *29*, *111*
\CTEXnumber *31*
\CTEXnumberline *27*
\CTEXoptions *37*
\ctexset *5*, *6*, *8*, *9*, *13*, *16*, *17*, *32*, *33*, *35*, *37*, *131*
\CTEXsetup *37*
\CTEXthechapter *15*, *28*, *111*
\CTEXthesection *15*, *19*, *114*
\CTEXunderdotbasesep *36*
\CTEXunderline *36*
\curr@fontshape 1051, 1085, 1094, 1100, 1142
\declare@file@substitution *158*, 6129
\DeclareAlternateKanjiFont *77*
\DeclareFontEncoding *66*
\DeclareFontFamily *50*, *65*
\DeclareFontShape *65*
\DeclareFontShape@ *77*, 1711
\DeclareRobustCommand *161*
\DeclareTextCommand *60*
\DeclareTextSymbol *60*
\DeclareUnicodeAccent *60*
\DeclareUnicodeComposite *60*
\define@newfont *64*, 1077
\disable@package@load *158*, 6116, 6118
\do@noligs 1910
\do@subst@correction *64*, *65*, 1076
\em *81*
\endlinechar *52*

\escapechar 67
\EverySelectfont 87
\expandafter 35
\external@font 1087, 1088, 1090
\extract@font 64, 1075
\f@baselineskip 2461
\f@encoding 1048, 1093, 1197
\f@family 63, 1049, 1093, 1103
\f@nch@initialise 3635
\f@series 1044, 1135
\f@shape 1044, 1135
\f@size 1044, 1051, 1130, 1138, 1141, 1142, 2145, 2199, 3572
\fangsong 8
\fbox 20
\figurename 14
\font 65
\font@name 63, 65, 564, 566, 567, 1054, 1061, 1062, 1067, 1072, 1090, 1091, 1110, 1178, 1179, 1180, 1181, 1182, 1189
\fontcharwd 87
\footnotesep 129
\footnotesize 129
\get@external@font 1084
\getanddefine@fonts 1167
\globaldefs 63
\globaljfont 64
\H@old@chapter 3373
\H@refstepcounter 3817
\hbox 24, 99
\heiti 8, 146
\hrule 99
\Hy@chapapp 3030, 3036, 3042, 3059
\Hy@driver 2084
\Hy@MakeCurrentHrefAuto 3393
\Hy@org@chapter 3373
\Hy@raisedlink 3394
\hyper@anchorend 3397
\hyper@anchorstart 3396
\hypersetup 84
\if@compatibility 1899
\if@mainmatter 3019, 3077, 3621, 3671, 4118, 4119
\if@nobreak 3126
\if@noskipsec 2834, 3113, 3229
\if@openright 2850, 2992, 3004, 3891, 3919
\if@tempswa 2998
\if@twocolumn 2858, 3051, 3061
\if@twoside 2991
\iffalse 66
\IfFormatAtLeastTF 157
\ifin@ 66, 414
\iftrue 66
\ifttl@label 3508, 3510
\ifttl@toclabel 3508, 3510
\in@ 413
\indexname 14
\input 157
\InputIfFileExists 157
\insertcontinuationtext 14
\itdefault 65
\itshape 58
\jfam 81
\kaishu 8
\kanjifamilydefault 83
\kanjiskip 89
\l@nohyphenation 1906
\labelformat 34, 35, 117, 118
\leftmark 15
\linewidth 92
\lishu 8
\listfigurename 13
\listtablename 14
\ltj@@does@alt@set 65
\ltj@@getjfontnumber 66, 1150
\ltj@@set@stackfont 1187, 1188, 1189
\ltj@allalchar 894
\ltj@curjfnt 64, 88, 1097
\ltj@pickup@altfont@auxy 63, 1136
\ltj@pickup@altfont@copy 63, 1148
\ltj@setpar@global 80, 1186, 1913, 1915
\ltj@tempcntc 66, 1151
\ltjalchar 61
\ltjlineendcomment 59
\ltjsetkanjiskip 80
\ltjsetparameter 80
\ltjsetxkanjiskip 80
\m@ne 2872, 2894, 3017, 3076, 3614, 3621, 3622, 3644, 3671, 3672, 4596, 4597, 4598, 4599, 4600, 4601
\MakeUppercase 54
\markboth 15
\markright 15
\math@bgroup 1196
\math@egroup 1201
\math@fonts 1184
\mathbb 129
\maxdimen 12, 92
\meaning 161–163
\mule@arg 671
\newcommand 161
\newrobustcmd 161
\newtitlemark 112, 113
\nfss@catcodes 68
\noindent 98
\normalfont 83
\normalsize 10, 124
\nouppercase 52
\NR@gettitle 3406
\null 80
\numberline 113
\p@ 2145, 2199, 2937, 3048, 3049, 3068, 3078, 3088, 3093, 3107, 3572, 3887, 3915, 3916, 3917, 3918, 3925, 4381, 4382, 4383, 4389, 4390, 4391, 4393, 4394, 4400, 4401, 4402, 4404, 4405, 4420, 4421, 4422, 4428, 4429, 4430, 4432, 4433, 4439, 4440, 4441, 4443, 4444

\pagenumbering 31
\pagestyle 11
\par@update 88, 2222, 2226, 2230
\paragraph 11, 23, 42
\parindent 12, 36, 91, 93, 98, 107, 129
\parskip 99
\part 17, 23, 25, 26, 99, 106
\partmark 99
\partname 18
\partpage 16, 115
\PassOptionsToClass 43
\PassOptionsToPackage 84, 94
\pdfmapline 135
\pickup@font 63, 1050
\pickup@jfont 1081
\pingfang 8, 146
\ProcessKeysOptions 34, 43
\proofname 14, 95
\protected 54, 112
\protected@edef 2564, 2566, 3151, 3786
\ProvidesFile 148
\ps@headings 114, 3602, 3655, 3658, 3663, 3669, 3680
\refname 14
\refstepcounter 35, 117
\refstepcounter@noarg 3820
\refstepcounter@optarg 3821
\relax 52, 92
\rightmark 15
\rmfamily 58, 82, 83
\roman 31
\scantokens 161
\scr@new@selectfont 2219, 2234
\scr@selectfont 2231, 2235
\scriptfont@name 1181, 1188
\section 11, 17–19, 23–26, 28, 37, 96, 106, 114
\section* 17
\sectionmark 15
\sectionname 18
\sectionpage 16, 115
\selectfont 66, 82, 87, 88, 92
\setCJKfamilyfont 32
\setCJKmainfont 7, 32
\setcounter 116
\sf@size 1180
\sffamily 58
\size@update 92, 2246, 2247, 2421, 2435
\sldefault 65
\songti 8
\special 7, 50, 84
\split@name 1141
\SplitArgument 98
\ssf@size 1182
\strutbox 129
\subparagraph 11, 23, 42
\subsectionname 18
\subsectionpage 16, 116
\subst@correction 1106, 1113
\tablename 14
\textfont@name 1179, 1187
\textlangle 52
\textrangle 52
\tf@size 1178
\thechapter 15, 28
\thesection 15, 19, 28, 114
\thispagestyle 99
\titleformat 110
\titlespacing 25, 110
\today 13
\trans@languagepath 4167, 4192
\ttfamily 58
\ttl@a 3488, 3496
\ttl@chapterout 3483
\ttl@extract 110
\ttl@labelling 3507
\ttl@setifthe 3553
\ttl@setsubmark 112, 3526, 3532
\ttl@settopmark 112, 3520, 3551
\ttl@tocpart 3485, 3487
\ttlh@hang 3449, 3459
\ttlh@runin 3448, 3458
\tw@ 414
\uppercase 52
\use@mathgroup 1192, 1223
\usepackage 36, 38
\verb 80
\verb@eol@error 1904
\verbatim@font 1890, 1894, 1905
\vskip 24
\wrong@fontshape 64
\x@protect 6300
\xdef 112
\XeTeXglyphbounds 152
\xkanjiskip 88, 89
\yahei 8, 146
\youyuan 8
\z@ 879, 882, 1910, 2891, 2953, 3008, 3073, 3098, 3117, 3160, 3198, 3217, 3231, 3468, 3606, 3631, 3638, 3647, 3656, 3664, 3681, 4382, 4390, 4401, 4421, 4429, 4440
\zhdig 94
\zhdigits 31
\zhnum 94
\zhnumber 31
\zihao 29
\ziju 31
\zw 87

tex commands:

\tex_afterassignment:D 902
\tex_chardef:D 903
\tex_def:D 6376
\tex_dimexpr:D 105, 2472
\tex_divide:D 92, 2473, 2513
\tex_edef:D 6430

\tex_endlinechar:D 535
\tex_font:D 482, 483, 5841, 5842
\tex_glueexpr:D 2461
\tex_hangindent:D 3264
\tex_iftrue:D 4119
\tex_ignorespaces:D 1356, 1522, 1527, 2331, 2497, 4209, 6186
\tex_indent:D 2780
\tex_interlinepenalty:D 2769
\tex_kanjiskip:D 2152, 2168, 2182
\tex_noindent:D 2770, 3262
\tex_numexpr:D 923
\tex_pagegoal:D 2788
\tex_parindent:D 2774, 2779, 3254
\tex_parskip:D 2796
\tex_pdffontattr:D 483
\tex_prevdepth:D 2795
\tex_setfontid:D 1097
\tex_topskip:D 2789
\tex_Uchar:D 911, 922, 923
\tex_undefined:D 526
\tex_XeTeXcharglyph:D 5848
\tex_XeTeXglyphbounds:D 5897
\tex_xkanjiskip:D 2279, 2283, 2293, 2367
text commands:
\l_text_case_exclude_arg_tl 716
\text_declare_expand_equivalent:Nn . . . 723, 724, 725
\text_declare_purify_equivalent:Nn 749, 750, 751, 752
\text_expand:n *54*, *56*
\text_purify:n *56*
\text_uppercase:n *54*
\textasteriskcentered 952
\the 5490, 5493, 5494, 5513, 5517
\thechapter 3024, 3027, 3078, 3623, 3644, 3673, 3922
\theparagraph 4024
\thepart 2876, 2895, 2917, 2937, 3861
\thesection 3606, 3614, 3631, 3638, 3647, 3656, 3664, 3681, 3941
\thesubparagraph 4038
\thesubsection 3609, 3641, 3659, 3973
\thesubsubsection 4007
\thispagestyle 2804, 2856, 2994, 3005
\tiny 4410, 4449
tl commands:
\c_novalue_tl 6165
\tl_clear:N 189, 352, 1844, 1848, 2120, 2132, 2272, 2412, 2739, 2757, 2768, 4336, 5859, 5860, 6384
\tl_clear_new:N 55, 329, 5879, 5880
\tl_const:Nn 120, 813, 1252, 1265, 1600, 2009, 2011, 2012, 2013, 2020, 2021, 2022, 2431, 2662, 2668, 4101, 4120, 4122, 5381, 5384, 5385, 6121
\tl_gput_right:Nn 423, 455, 762, 2014, 2015, 2016, 2017, 2023, 2024, 2025, 6180, 6182, 6200
\tl_gset:Nn 218, 221, 224, 377, 547, 550, 1062, 1178, 1180, 1182, 1574, 2026, 2037, 4503, 4506, 4507, 4523, 4556, 5028
\tl_gset_eq:NN 1067, 2058
\tl_gset_rescan:Nnn 1581
\tl_head:N 4340
\tl_head:n 1625
\tl_if_blank:nTF 922, 1659, 1668, 1868, 1869, 1884, 2384, 2394
\tl_if_blank_p:n 5875, 5876
\tl_if_empty:NTF 373, 1039, 2490, 2674, 3801, 4514, 5062, 5249, 5268, 5282, 5290, 5300, 5305
\tl_if_empty:nTF 109, 1944, 1958, 1971, 6347
\tl_if_eq:NNTF 2033, 2424
\tl_if_eq:NnTF 4179
\tl_if_exist:NTF . . . 72, 761, 2008, 2011, 2012, 2013, 2020, 2021, 2022, 2589, 2620, 2631, 2634, 4113, 4202, 6199
\tl_if_in:NnTF 6411
\tl_if_in:nnTF 5032
\tl_if_novalue:nTF 1483, 1506, 1862, 2738
\tl_map_inline:Nn 1774
\tl_map_inline:nn 4337
\tl_new:N 110, 112, 444, 489, 761, 905, 1041, 1042, 1133, 1250, 1300, 1327, 1344, 1345, 1366, 1716, 1850, 2001, 2051, 2286, 2432, 2445, 2591, 2660, 2661, 3339, 3340, 5010, 6093, 6193, 6194, 6199, 6387, 6388, 6389
\tl_put_left:Nn 2237, 6216
\tl_put_right:Nn 336, 716, 1184, 1857, 2225, 2226, 2230, 2690, 2728, 2730, 4340, 4341, 6210
\tl_remove_all:Nn 1626
\tl_replace_all:Nnn 1672
\tl_rescan:nn *163*, 6429, 6438
\tl_set:Nn 100, 184, 518, 519, 520, 901, 1043, 1087, 1134, 1234, 1286, 1349, 1361, 1505, 1669, 1671, 1712, 2053, 2215, 2271, 2287, 2345, 2364, 2408, 2446, 2500, 2592, 2737, 2740, 2744, 3445, 3464, 3483, 3488, 3496, 4468, 5025, 5864, 5866
\tl_set_eq:NN 1048, 1049, 1103, 1179, 1181, 1350, 1375, 1812, 1824, 2433
\tl_set_rescan:Nnn 6385
\tl_tail:N 3450
\tl_tail:n 3460
\tl_to_str:n . . 108, 653, 660, 667, 6328, 6357, 6398, 6399
\tl_trim_spaces:n 1671
\tl_trim_spaces_apply:nN 1841
\tl_use:N 425
\l_tmpa_tl 5025, 5027, 5028
tocdepth *27*, 3753
\today 2567, 2572, 2575, 2580
today *13*, 2567
token commands:
\c_parameter_token *113*, 3584, 3586, 3589
\token_if_eq_meaning:NNTF 596, 610, 627
\token_to_meaning:N 6342
\token_to_str:N *67*, 144, 929, 1161, 1169, 1178, 1180, 1182, 1403, 1404, 1405, 1407, 1414, 1415, 1416, 1478, 1509, 2617, 2618, 4217, 5852, 6060, 6139, 6154, 6246
\toks 5490, 5491, 5493, 5494, 5513, 5517
\topsep 4393, 4404, 4432, 4443
\TrimSpaces 2382

\ttdefault 2043
\ttfamily 2006
\twocolumn 2999
\typeout 3024, 3025, 3483

U

ubuntu 38
um commands:
\um_input_math_symbol_table: 1216, 1220
\um_sym:nnn 1219
um internal commands:
__um_input_math_symbol_table: 1228
__um_switchto_literal: 1225
__um_sym:nnn 1227
\undefined 5515
\UnicodeEncodingName 929
\unless 3161, 3199, 3218
\unskip 3234
\upshape 1934, 1937
use commands:
\use:N 686, 692, 698, 704, 1093, 1094, 1178, 1180, 1182, 1270, 1319, 1825, 2804, 2816, 2822, 2831, 3248, 3311, 6412, 6432, 6446
\use:n 113, 492, 572, 649, 656, 663, 746, 747, 748, 1141, 1254, 1433, 1486, 1492, 1571, 1575, 1674, 2052, 3257, 3286, 3577, 4111, 4528, 6159, 6281, 6323, 6374, 6395
\use:nn 6440
\use_i:nn 97, 2810, 6227, 6232, 6252, 6257, 6267
\use_i:nnn 239
\use_ii:nn ... 98, 2808, 2812, 6227, 6232, 6252, 6257, 6268
\use_ii:nnn 114, 244
\use_iii:nnn 249
\use_none:n 185, 190, 195, 500, 1120, 1335, 1574, 1709, 2553
\use_none:nn 501
\use_none_delimit_by_q_stop:w 139, 142, 5038, 5058, 5078, 5120, 5135, 5195, 5315
\usebeamercolor 3695, 3714, 3733, 3904, 3964, 3998
\usebeamerfont 3694, 3705, 3713, 3724, 3732, 3743, 3903, 3906, 3963, 3966, 3997, 4000
\usefont 2007
\uselanguage 4158, 4163, 4183, 4188
\usepackage 6060
UTF8 ..... 6, 38, 215

V

\vadjust 1902, 1910
\verb 1900
\verse 4478
\vfil 2864, 2986
\vskip 2905, 2908, 2937, 2962, 2965, 2989, 3078, 3088, 3091, 3107, 3110, 3220, 3223, 3696, 3715, 3734, 3887, 3905, 3925, 3965, 3999
\vspace 2867, 3068, 3071, 3093, 3096

W

windows 38
windowsnew 38
windowsold 38
winfonts 36
\write 5530

X

\x 407, 413, 417, 5492, 5495, 5496, 5497, 5498, 5499, 5500, 5501, 5502, 5503, 5504, 5505, 5506, 5507, 5508, 5509, 5510, 5511, 5518, 5528, 5532
x commands:
\x: 412
\xdef 5531
\xeCJKsetup 819, 2128, 2326, 2329, 2332, 2347, 2359

Y

\yahei 5452
\youyuan 5452

Z

\zhdig 2559
\zhdigits 31, 2566
zhmap 7, 36, 234
\zhnum 2558
zhnum commands:
\zhnum_counter:n 2550
\zhnumber 31, 2564
\zhnumsetup 2576, 2581
\zhtoday 2575, 2580
\zihao 29, 4208, 4217
zihao 10, 164
\ziju 31, 2496
\zw 880, 2149, 2200, 2288