

CT_EX 宏集手册

ctex.org

2016/04/25 v2.4*

简介

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

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

目录

第 1 节 介绍	2	第 8 节 实用命令	23
第 2 节 简明教程	3	8.1 字号与间距	23
2.1 CT _E X 宏集的组成	3	8.2 中文数字转换	23
2.2 CT _E X 宏集的安装和更新	3	8.3 杂项	24
2.3 使用 CT _E X 文档类	4	第 9 节 LuaL ^A T _E X 下的中文支持方式	24
2.4 使用 ctex 宏包	4	9.1 LuaL ^A T _E X 下替代字体的设置	25
第 3 节 宏包选项与 \ctexset 命令	5	第 10 节 CT _E X 宏集的配置文	26
第 4 节 编译方式、编码与中文字库	5	10.1 修改宏包默认选项	26
4.1 编译方式	5	10.2 宏包载入后的配置	27
4.2 中文编码	6	10.3 配置标题中文翻译	27
4.3 中文字库	6	10.4 自定义字体集	27
第 5 节 排版格式设定	8	第 11 节 对旧版本的兼容性	27
5.1 文档默认字号	8	11.1 CT _E X 0.8a 及以前的版本	27
5.2 章节标题风格	9	11.2 CT _E X 0.9–CT _E X 1.0d	28
5.3 排版方案选项	9	11.3 CT _E X 1.02c 以后的 SVN 开发版	29
第 6 节 文档汉化	11	11.4 CT _E X 2.2 之前的版本	30
6.1 日期汉化	11	第 12 节 宏集依赖情况与手工安装方法	30
6.2 文档标题汉化	11	第 13 节 开发人员	31
6.3 页面格式设置与汉化	12	第 14 节 参考文献	31
第 7 节 章节标题格式设置	14	第 15 节 代码实现	32
		版本历史	135
		代码索引	136

*ctex-kit rev. 1fcc459.

第 1 节 介绍

历史

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

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

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

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

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

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

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

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

关于宏集名字の説明

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

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

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

¹<http://code.google.com/p/ctex-kit/>

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

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

第 2 节 简明教程

2.1 C_TE_X 宏集组成

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

表 1 C_TE_X 宏集的组成

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

2.2 C_TE_X 宏集的安装和更新

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

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

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 C_TE_X 宏集⁵。

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

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

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

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

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

⁵*nix 用户可能需要超级用户权限才能正确安装宏集。

```
tlmgr update --all
```

来完整更新已安装的宏包。

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

2.3 使用 CT_EX 文档类

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

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

例 1

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

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

以下是使用 `ctexbeamer` 文档类编写中文演示文稿的一个示例。

例 2

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

2.4 使用 ctex 宏包

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

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

例 3

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

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

第 3 节 宏包选项与 `\ctexset` 命令

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

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

另一方面,这些选项可以分为以下三类:

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

`\ctexset {<键值列表>}`

New: 2014-03-18

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

`\ctexset` 的参数是一组由逗号分隔的选项列表,列表中的选项通常是一个 `<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 L^AT_EX3 风格的键值设置,支持不同类型的选项与层次化的选项设置,相关示例见 7 节。

第 4 节 编译方式、编码与中文字库

4.1 编译方式

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

不同的编译方式和中文支持方式会在一定程度上影响 CT_EX 宏集的行为,比如宏包对编码的处理。在用户使用 X_YL^AT_EX、Lua^AT_EX 及 up^AL^AT_EX 编译时,CT_EX 宏集使用(且仅能使用) UTF-8 编码;而因为历史原因,在用户使用 L^AT_EX 及 pdf^AL^AT_EX 编译时,宏包默认使用 GBK 编码。用户需要保证编译方式、源文件编码、宏包编码选项三者一致。关于宏包编码选项,可以参考 4.2 节。

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

⁷L^AT_EX、pdf^AL^AT_EX、X_YL^AT_EX、Lua^AT_EX 及 up^AL^AT_EX。

表 2 CT_EX 宏集的中文支持方式

编译方式	支持宏包
(pdf)La _T _E X	CJK
X _Y La _T _E X	xeCJK
LuaLa _T _E X	LuaTeX-ja
upLa _T _E X*	原生

* pLa_T_EX-ng (或称 ApLa_T_EX) 与 upLa_T_EX 兼容。使用 pLa_T_EX-ng 编译时, ctex 采用与 upLa_T_EX 相同的设置。

4.2 中文编码

GBK ☆
UTF8 ☆

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

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

4.3 中文字库

以往,为 La_T_EX 文档配置中文支持是一件相当繁琐的事情。默认情况下,CT_EX 宏集能自动检测用户使用的编译方式(参见 4.1 节)和操作系统⁸,选择合适的底层支持和字库,从而简化配置过程。自动配置的情况参见表 3。

通常,由 CT_EX 宏集进行的自动配置已经足够使用,无需用户手工干预;但是 CT_EX 仍然提供了一系列选项,供在 CT_EX 的自动选择机制因为意外情况失效,或者在用户有特殊需求的情况下使用。除非必要,用户不应使用这些选项。

zhmap ☆

zhmap = <true|false|zhmCJK>

Updated: 2014-03-08

指定字体映射机制。本选项只在使用 pdfLa_T_EX/La_T_EX 编译时有意义。

true

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

false

使用传统的 CJK 字库(Type 1)⁹。

zhmCJK

载入 zhmCJK 宏包¹⁰,由 zhmCJK 宏包提供从 CJK 字库到 .ttf 的映射。

fontset

fontset = <none|adobe|fandol|founder|mac|ubuntu|windows|windowsnew|windowsold|...>

New: 2014-03-08

指定 CT_EX 宏集加载的字库。

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

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

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

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

表 3 CT_EX 宏集自动配置字体策略

	Mac OS X	Windows New ¹	Windows Old ²	其他
X _E L ^A T _E X	xeCJK 华文字库	xeCJK 中易字库 + 微软雅黑	xeCJK 中易字库	xeCJK Fandol 字库 ³
LuaL ^A T _E X ⁴	LuaTeX-ja 华文字库	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja 中易字库	LuaTeX-ja Fandol 字库
pdfL ^A T _E X	不可用 ⁵	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	不可用 ⁵
L ^A T _E X + DVIPDFMx	不可用 ⁶	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	CJK + zhmetrics Fandol 字库
upL ^A T _E X + DVIPDFMx	不可用 ⁶	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex 中易字库	zhmetrics-uptex Fandol 字库

¹ Windows Vista 及以后的 Windows 操作系统。

² Windows XP 及以前的 Windows 操作系统。

³ 由马起园、苏杰、黄晨成等人开发的开源中文字体, 参见: <https://github.com/clerkma/fandol-fonts>。

⁴ LuaL^AT_EX 编译时使用 LuaTeX-ja 宏包。对此, 第 9 节有特别说明。

⁵ 受 pdfL^AT_EX 的限制, 无法嵌入 OpenType 字体。

⁶ 目前受 DVIPDFMx 的限制, Mac OS X 系统上的黑体和仿宋无法读取。

CT_EX 预定义了以下六种中文字库。

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

注意: 使用 (pdf)L^AT_EX 编译的时候, 若设置 `zhmap = false` (比如需要使用 L^AT_EX + Dvips 编译), 则需要按照传统方式¹¹在本地安装好 CJK 字体。

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

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

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

例 5

```
\documentclass[fontset = none]{ctexart}
\ctexset{fontset = founder}
\begin{document}
```

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

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}
```

CT_EX 宏集预定义的中文字库还定义了一些字体命令。

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

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

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

在 windowsnew 字库中,还有 `\yahei`。

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

第 5 节 排版格式设定

5.1 文档默认字号

`zihao`

☆

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

New: 2015-05-06

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

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

表 4 标准字体命令与字号的对应

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

10pt ☆

11pt ☆

12pt ☆

CT_EX 文档类是在 L^AT_EX 标准文档类之上开发的。因此,除了可以使用 CT_EX 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,CT_EX 文档类的字号选项会被抑制。亦即,在 `zihao` 选项之后设置 10pt 选项,zihao 选项将不再起作用。

标准文档类的其他选项在 C_TE_X 文档类中依旧有效。例如，设置纸张大小和方向的 a4paper 和 landscape，设置单双面的 oneside 和 twoside 等。C_TE_X 会将这些选项传给标准文档类¹²。

5.2 章节标题风格

heading ☆ heading = <true|false>

New: 2014-03-08

本选项只能在调用 ctex.sty 时作为宏包选项使用。

C_TE_X 宏集提供了一套用于修改文档章节标题格式的接口。该选项用于选择是否启用该功能。详细的设置方法请参见 7 节和 6.3 节。

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

sub3section ☆
sub4section ☆

修改 \paragraph 和 \subparagraph 的格式。

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

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

具体格式可参考 7 节中的 runin 和 afterskip 选项。

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

5.3 排版方案选项

scheme ☆ scheme = <chinese|plain>

New: 2015-04-15

选择文章的排版方案，预设有 chinese 和 plain 两种方案。

chinese 调整默认字号为五号字，并调整行距为 1.3；汉化文档中的标题名字（如“图”、“表”、“目录”和“参考文献”等，见 6.2 节）；在 heading = true 的情况下¹³（5.2 节），还会将章节标题的风格修改为中文样式（见 7 节）。

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

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

punct punct = <quanjiao|banjiao|kaiming|CCT|plain>

Updated: 2014-04-11

设置标点处理格式。预定义好的格式有：

¹²事实上，L_AT_EX 在文档类中的选项是全局设定的，除了对使用的文档类有影响外，也可能影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中，那么该选项将会被自动激活。

¹³使用 C_TE_X 文档类，或者使用 ctex 宏包且开启该选项时。

quanjiao	全角式:所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;
banjiao	半角式:所有标点占半个汉字宽度;
kaiming	开明式:句末点号 ¹⁴ 用占一个汉字宽度,标号和句内点号占半个汉字宽度;
CCT	CCT 式:所有标点符号的宽度略小于一个汉字宽度;
plain	原样(不调整标点间距)。

space

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

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。

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

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

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

linespread ☆

linespread = $\langle \text{数值} \rangle$

New: 2014-04-23

接受一个浮点数值,设置行距倍数。本选项的初始值与 scheme 有关。

scheme = chinese	初始值为 1.3,即 1.3 倍行距。此时,相邻两行的基线(\baselineskip)距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。
scheme = plain	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 = $\langle true|false| \text{数值} | \text{带单位的数值} \rangle$

New: 2014-03-13

在字体大小发生变化时,是否自动调整段首缩进(\parindent)的大小。

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

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

¹⁴标点符号分为标号与点号。点号分为两类,一共七种:句末点号有句号、问号和叹号;句内点号有逗号、顿号、冒号和分号。

¹⁵LaTeX 将单个换行视作一个空格。

例 8

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

linestretch ★ `linestretch = <数值或长度>`

New: 2014-03-26

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

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

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

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

第 6 节 文档汉化

6.1 日期汉化

CT_EX 宏包对显示当前日期的 `\today` 命令进行了汉化, 使之以中文的方式显示今天的日期。如编译本文档的日期就是“2016 年 4 月 25 日”。

today ★ `today = <small|big|old>`

该选项用来控制 `\today` 命令的输出格式:

`small` 效果为“2016 年 4 月 25 日”。使用阿拉伯数字和汉字的日期格式。
`big` 效果为“二〇一六年四月二十五日”。使用全汉字的日期格式。
`old` 效果为“April 25, 2016”。使用文档原来的(英文)日期格式。

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

例 9

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

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

6.2 文档标题汉化

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

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

contentsname ★ `contentsname = <名字>`

设置目录标题名 `\contentsname`。中文默认为“目录”。

<hr/> <code>listfigurename</code> ★ <hr/>	<code>listfigurename = <名字></code> 设置插图目录标题名 <code>\listfigurename</code> 。中文默认为“插图”。
<hr/> <code>listtablename</code> ★ <hr/>	<code>listtablename = <名字></code> 设置表格目录标题名 <code>\listtablename</code> 。中文默认为“表格”。
<hr/> <code>figurename</code> ★ <hr/>	<code>figurename = <名字></code> 设置图片环境标题名 <code>\figurename</code> 。中文默认为“图”。
<hr/> <code>tablename</code> ★ <hr/>	<code>tablename = <名字></code> 设置表格环境标题名 <code>\tablename</code> 。中文默认为“表”。
<hr/> <code>abstractname</code> ★ <hr/>	<code>abstractname = <名字></code> 设置摘要 <code>abstract</code> 环境标题名 <code>\abstractname</code> 。中文默认为“摘要”。注意 <code>book</code> 类没有摘要, 该选项无效。
<hr/> <code>indexname</code> ★ <hr/>	<code>indexname = <名字></code> 设置索引标题名 <code>\indexname</code> 。中文默认为“索引”。
<hr/> <code>appendixname</code> ★ <hr/>	<code>appendixname = <名字></code> 设置附录标题名 <code>\appendixname</code> 。中文默认为“附录”。
<hr/> <code>bibname</code> ★ <hr/>	<code>bibname = <名字></code> 设置参考文献标题名 <code>\refname</code> (对 <code>article</code>) 或 <code>\bibname</code> (对 <code>report</code> 、 <code>book</code> 和 <code>beamer</code>)。中文默认为“参考文献”。
<hr/> <code>proofname</code> ★ <hr/>	<code>proofname = <名字></code> 设置证明环境的名称 <code>\proofname</code> 。中文默认为“证明”。 如果使用 <code>ctexbeamer</code> 文档类或者在 <code>beamer</code> 文档类下使用 <code>ctex</code> 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。
<hr/> <code>refname</code> ★ <hr/>	<code>refname = <名字></code> 设置参考文献标题名 <code>\refname</code> 。中文默认为“参考文献”。
<hr/> <code>algorithmname</code> ★ <hr/>	<code>algorithmname = <名字></code> 设置算法环境标题名 <code>\algorithmname</code> 。中文默认为“算法”。
<hr/> <code>continuation</code> ★ <hr/>	<code>continuation = <名字></code> 设置 <code>beamer</code> 可断页的帧在续页标题中的延续标识 <code>\insertcontinuationtext</code> 。中文默认为“(续)”。

6.3 页面格式设置与汉化

当使用了 C_TE_X 的文档类或是用 `ctex` 宏包加载了 `heading` 选项时, 会设置整个文档的页面格式(`page style`)为 `headings`, 即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

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

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

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

- \CTEXthechapter、\CTEXthesection 等章节编号(见 7 节)。它们用来代替英文文档类中的 \thechapter、\thesection 等宏。
- \leftmark、\rightmark,它们是在使用章节标题命令后,自动设置的宏。它们实际是在与章节标题命令对应的标记命令 \chaptermark、\sectionmark 中调用 \markright 或 \markboth 生成的。

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

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

例 10

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

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

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

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

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

例 11

```
\documentclass{ctexart}
\ctexset{section={
```

```

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

本例的页眉效果大致如下(有页眉线):

第 1 节——天地玄黄

— 1 —

第 7 节 章节标题格式设置

CT_EX 宏包对 L^AT_EX 的标准文档类(article、report 和 book)和 beamer 进行了扩充。当以 heading 选项调用 CT_EX 宏包时(5.2 节),则会启用章节标题的格式设置功能。本节就来介绍有关章节标题的格式选项,所有选项使用 \ctexset 命令设置。

第 6.3 节和本节介绍的功能已经被提取到 ctexheading 宏包之中,可以在 ctex 宏包和 ctexart 等文档类之外独立使用。各选项的默认值与 scheme = plain 时的情形相同。

章节标题的格式选项是分层设置的。顶层的选项是章节标题名称,次一级的选项是章节标题的格式。章节标题名包括 part, chapter, section, subsection, subsubsection, paragraph, subparagraph;而可用的格式包括 numbering, name, number, format, nameformat, numberformat, aftername, titleformat, aftertitle, runin, afterindent, beforeskip, afterskip, indent, pagestyle 等。

注意,对 article 及其衍生的 ctexart 等文档类,没有 chapter 级别的标题。而对于 beamer 文档类,这些选项控制的是由 \partpage、\sectionpage 和 \subsectionpage 产生的标题格式,此时只有 part、section 和 subsection 这三层级别,并且 runin 和 afterindent 这两个格式无效。

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

使用 \ctexset 设置多级选项时,可以在同一个上级选项下设置多个下级选项。

例如,同时设置 part 一级标题的 pagestyle 选项,chapter 一级标题的 format 与 pagestyle 选项和 section 一级标题的 name 与 number 选项:

例 12

```

\ctexset {
  part/pagestyle = empty,
  chapter = {
    format      = \raggedright,
    pagestyle = empty,

```



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

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

New: 2015-06-21

numbering = true|false
控制是否对章节标题编号。对各级标题的默认值均为 true。

我们知道， \LaTeX 带星号的章节标题命令（如 \section* ）不会对标题编号，但也不会将该没编号的标题编入目录中。本选项控制的是不带星号的标题命令是否编号。设置本选项为 false，除了不对标题编号以外，功能与正常标题一致，比如可以编目录，正确的 \hyperref 目录超链接位置和页眉标记。

注意，章节标题的编号深度受 \LaTeX 计数器 secnumdepth 的控制。numbering 选项在 secnumdepth 的控制下起作用。

如果没有特别说明，以下将用“...”代表各级章节标题名。

.../name ★
Updated: 2014-03-08

name = {<前名字>,<后名字>}
name = {<前名字>}

设置章节的名字。名字可以分为前后两部分，即章节编号前后的词语，两个词之间用一个半角逗号分开；也可以只有一部分，表示只有章节编号之前的名字。例如：

例 13

```
\ctexset{
  chapter/name = {第,章},
  section/name = {\S},
}
```

会使得 \chapter 标题使用形如“第一章”的名字，而 \section 标题则使用形如“§1”的名字。

表 5 name 选项的默认设置

标题名	scheme = chinese	scheme = plain	注
part	{第,部分}	{\partname\space}	原 \partname 为 Part
chapter	{第,章}	{\chaptername\space}	原 \chaptername 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 \sectionname 为 \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 14

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

`number` 选项的定义同时将控制对章节计数器的交叉引用。在引用计数器时,记录在 L^AT_EX 辅助文件中的是 `number` 选项的定义。

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

表 6 `number` 选项的默认设置

标题名	scheme = chinese	scheme = plain	原 <code>\the⟨标题⟩</code> 等价定义
part (beamer)	<code>\chinese{part}</code>	<code>\insertromanpartnumber</code>	意义为 <code>\Roman{part}</code>
part	<code>\chinese{part}</code>	<code>\thepart</code>	<code>\Roman{part}</code>
chapter	<code>\chinese{chapter}</code>	<code>\thechapter</code>	<code>\arabic{chapter}</code>
section (beamer)	同右	<code>\insertsectionnumber</code>	意义为 <code>\arabic{section}</code>
section	同右	<code>\thesection</code>	<code>\arabic{section}</code>
subsection (beamer)	<code>\arabic{section}.</code> <code>\arabic{subsection}</code>	<code>\insertsubsectionnumber</code>	意义为 <code>\arabic{subsection}</code>
subsection	同右	<code>\thesubsection</code>	<code>\thesection.\arabic{subsection}</code>
subsubsection	同右	<code>\thesubsubsection</code>	<code>\thesubsection.\arabic{subsubsection}</code>
paragraph	同右	<code>\theparagraph</code>	<code>\thesubsubsection.\arabic{paragraph}</code>
subparagraph	同右	<code>\thesubparagraph</code>	<code>\theparagraph.\arabic{subparagraph}</code>

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

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

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

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

Updated: 2015-06-30

`format` 选项用于控制章节标题的全局格式,作用域为章节名字和随后的标题内容。可以用于控制章节标题的对齐方式、整体字体字号等格式。

带加号的 `format+` 选项则用于在已有格式之后追加新的格式命令。

例如,设置章格式为无衬线字体左对齐,为节格式增加无衬线字体设置:

例 15

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

表 7 format 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	\Large\bfseries\centering	\raggedright
part (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

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

Updated: 2015-06-30

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

nameformat+ 用于在已有的章节名字格式后附加内容。

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

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

表 8 nameformat 选项的默认设置

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

```
.../numberformat * numberformat = {<格式命令>}
.../numberformat+ * numberformat+= {<格式命令>}
```

Updated: 2015-06-19

numberformat 选项用于控制章节编号的格式,作用域仅为编号数字本身。对各级标题默认均为空,当你需要编号的格式和前后的章节名字不一样时可以使用。

numberformat+ 选项用于在原有编号格式后面附加格式命令。

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

例如,我们可以使用 numberformat 特别强调章标题中的数字:

例 16

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

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

第 4 章

```
.../aftername  ★ aftername = {{代码}}
.../aftername+ ★ aftername+= {{代码}}
```

Updated: 2014-03-08

`aftername` 选项的参数 `{代码}` 将被插入到章节编号与其后的标题内容之间,用于控制格式变换。常用于控制章节编号与标题内容之间的距离,或者控制标题是否另起一行。

`aftername+` 选项用于在原有代码后面附加代码。

表 9 `aftername` 选项的默认设置

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

```
.../titleformat  ★ titleformat = {{格式命令}}
.../titleformat+ ★ titleformat+= {{格式命令}}
```

Updated: 2015-06-30

`titleformat` 选项用于控制标题内容的格式,作用域为章节标题内容。

`titleformat+` 选项用于在原有标题格式后面附加格式命令。

需要注意的是,`titleformat` 选项的最后一个格式命令可以是一个有一个参数的命令。这个命令的参数接受标题内容,用于实现特殊效果。例如,实现多行标题的居中悬挂对齐:

例 17

```
\usepackage{varwidth} %% 提供 varwidth 环境
\ctexset{
  chapter/name = {第,回},
  chapter/titleformat = \chaptertitleformat
}
\newcommand\chaptertitleformat[1]{%% 以标题内容为参数
  \begin{varwidth}[t]{.7\linewidth}\#1\end{varwidth}}
.....
\chapter{情中情因情感妹妹\错里错以错劝哥哥}
```

上面的代码可以做出类似这样的章标题效果：

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

表 10 titleformat 选项的默认设置

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

```
.../aftertitle  ★ aftertitle = {{代码}}
.../aftertitle+ ★ aftertitle+= {{代码}}
```

New: 2015-06-19

aftertitle 选项的参数 {代码} 将被插入到章节标题内容之后。

aftertitle+ 选项用于在原有代码后面附加代码。

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

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

表 11 aftertitle 选项的默认设置

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

表 12 runin 选项的默认设置

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

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

```
subsection/runin  ★
```

```
subsubsection/runin  ★
```

```
paragraph/runin  ★
```

```
subparagraph/runin  ★
```

New: 2015-06-27

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

runin 选项的默认值, 在 scheme 选项的不同取值下相同。

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

.../afterindent ★ afterindent = true|false

New: 2015-06-27

afterindent 选项用于设置章节标题后首段的缩进。

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

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

表 13 afterindent 选项的默认设置

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

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

Updated: 2015-06-27

beforeskip 选项用于设置章节标题前的垂直间距。

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

表 14 beforeskip 选项的默认设置

标题名	默认值
part (article)	4ex
part (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

.../afterskip ★ afterskip = {<弹性间距>}

Updated: 2015-06-27

afterskip 选项控制章节标题与后面下方之间的距离。

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

afterskip 选项的默认值,在 scheme 选项的不同取值下相同。

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

表 15 `afterskip` 选项的默认设置

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

表 16 `indent` 选项的默认设置

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

`.../indent` ★ `indent = {<缩进间距>}`

Updated: 2015-06-27 `indent` 选项用于设置章节标题本身的首行缩进。

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

例如, 设置 `\section` 标题缩进 20pt:

例 18

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

`part/pagestyle` ★ `pagestyle = {<页面格式>}`

`chapter/pagestyle` ★

New: 2014-03-21

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

表 17 `pagestyle` 选项的默认设置

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

`appendix/numbering` ★ `numbering = true|false`

New: 2015-06-21

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

用法与普通章节 `numbering` 选项类似。

appendix/name
Updated: 2014-03-08

name = {(前名字),(后名字)}
name = {(前名字)}

设置附录章(对 **book** 与 **report**)或附录节(对 **article**)的名字。

用法与普通章节 **name** 选项类似。

注意该选项与 **appendixname** 选项 (6.2 节) 在意义上有些重叠, 但意义不同。**appendixname** 选项只用来重定义 `\appendixname`, 而不管 `\appendixname` 如何使用; 该选项则决定在章节标题中输出的名字, 可以调用 `\appendixname` 设置。

表 18 appendix/name 选项的默认设置

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

appendix/number ★

number = {(数字输出命令)}

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

用法与普通章节的 **number** 选项类似。

该选项也同时控制附录章节计数器的交叉引用。

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

表 19 appendix/number 选项的默认设置

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

我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 19

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

本例的设置效果大致如下:

第 8 节 实用命令

8.1 字号与间距

<code>\zihao</code>	<code>\zihao {<字号>}</code>
Updated: 2014-03-08	用于调整字号大小。其中 <字号> 的有效值共有 16 个，如表 20 所示。使用 <code>\zihao</code> 命令调整字体大小时，西文字号大小会始终和中文字号保持一致。

表 20 中文字号

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

<code>\ziju</code>	<code>\ziju {<中文字符宽度的倍数>}</code>
Updated: 2014-03-28	用于调整相邻汉字之间的间距，即（在正常中文行文中）前一个汉字的右边缘与后一个汉字的左边缘之间的距离。其中参数可以是任意浮点数值；而中文字符宽度指的是实际汉字的宽度，不包含当前字距。

这个命令会影响 `\ccwd` 的值，但不会影响英文字距。

<code>\ccwd</code>	当前汉字的字宽保存在长度寄存器 <code>\ccwd</code> 之中。汉字字宽是相邻两个汉字中心之间的距离，包含字距在内。因此修改字距会间接修改字宽。
Updated: 2014-03-27	

8.2 中文数字转换

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

$\backslash\text{chinese}$	$\backslash\text{chinese} \{ \langle \text{counter} \rangle \}$
Updated: 2014-03-08	$\backslash\text{chinese}$ 命令与 $\backslash\text{roman}$ 等命令的用法类似, 作用在一个 \LaTeX 计数器上, 将计数器的值以中文数字的形式输出。
$\backslash\text{zhnumber}$	$\backslash\text{zhnumber} \{ \langle \text{number} \rangle \}$
New: 2014-03-08	以中文格式输出数字。这里的数字可以是整数、小数和分数。
$\backslash\text{zhdigits}$	$\backslash\text{zhdigits} \{ \langle \text{number} \rangle \}$
New: 2014-03-08	将阿拉伯数字转换为中文数字串。
$\backslash\text{CTEXnumber}$	$\backslash\text{CTEXnumber} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$
	$\backslash \langle \text{macro} \rangle$ 必须是一个 \TeX 宏, 不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 $\langle \text{number} \rangle$ 转为中文数字, 最后将结果存储在 $\backslash \langle \text{macro} \rangle$ 里。对 $\backslash \langle \text{macro} \rangle$ 的定义是局部的, 将它展开一次就可以得到转换结果。
	一般来说, 并不需要使用 $\backslash\text{CTEXnumber}$, 直接使用 $\backslash\text{zhnumber}$ 即可。但是, 如果在文档中需要多次使用同一个数字 $\langle \text{number} \rangle$ 的中文形式, 就可以先用 $\backslash\text{CTEXnumber}$ 将结果保存起来备用, 而不是每次使用时都用 $\backslash\text{zhnumber}$ 现场转换一次。
$\backslash\text{CTEXdigits}$	$\backslash\text{CTEXdigits} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$
	$\backslash\text{CTEXdigits}$ 与 $\backslash\text{CTEXnumber}$ 类似, 但其转换的结果是中文数字串, 而不是中文数字。

8.3 杂项

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

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

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

9.1 Lua_{TeX} 下替代字体的设置

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

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

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

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

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

例 20

```

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

```

declarecharrange ★ <hr/> New: 2014-04-14	<pre> \ctexset { declarecharrange = { {<name₁>} {<character range₁>} , {<name₂>} {<character range₂>} , ... } } </pre>
--	--

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

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

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

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

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

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

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

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

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

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

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

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

10.1 修改宏包默认选项

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

例 21

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


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

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

10.2 宏包载入后的配置

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

例 22

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

例 23

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

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

第 11 节 对旧版本的兼容性

11.1 C_TE_X 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

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

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

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

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

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

<hr/> cs4size c5size <hr/>	分别相当于 <code>zihao=-4</code> 和 <code>zihao=5</code> , 过时选项。
<hr/> CCT CCTfont <hr/>	相关选项已删除。
<hr/> indent noindent <hr/>	<code>indent</code> 和 <code>noindent</code> 什么也不做, 过时选项。 在中文版式下, <code>ctex</code> 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开。C _T E _X 文档类的相关功能由章节标题的 <code>afterindent</code> 选项的值来确定。
<hr/> zhmap nozhmap <hr/>	<code>zhmap</code> 宏包选项增加了参数, 扩充了功能, 除了支持真假值参数外, 还支持选择 <code>zhmCJK</code> 作为底层中文处理宏包。(4.3 节) <code>nozhmap</code> 选项相当于 <code>zhmap=false</code> 。过时选项。
<hr/> winfonts adobefonts nofonts <hr/>	宏包选项 <code>winfonts</code> 相当于 <code>fontset=windows</code> , <code>adobefonts</code> 相当于 <code>fontset=adobe</code> , <code>nofonts</code> 相当于 <code>fontset=none</code> 。这几个选项是过时选项, 对于新文档, 应使用 <code>fontset</code> 选项设置不同字体集。 另外, 第 2 版 C _T E _X 宏包的默认字体不再是 Windows 系统字体, 而是根据检测到的操作系统选择使用 Windows、Mac 的系统字体还是 Fandol 字体(4.3 节)。
<hr/> punct nopunct <hr/>	旧版本中宏包 <code>punct</code> 选项没有参数, 现在可以用参数设定标点风格(5.3 节)。原有无参形式的 <code>punct</code> 选项相当于 <code>punct=quanjiao</code> 。 旧版宏包中 <code>nopunct</code> 选项的效果大致相当于 <code>punct=plain</code> 。过时选项, 不推荐使用。
<hr/> cap nocap <hr/>	原有的 <code>cap</code> 和 <code>nocap</code> 选项由新的 <code>scheme</code> 选项代替。(5.3 节) <code>cap</code> 选项相当于 <code>scheme = chinese</code> , <code>nocap</code> 选项相当于 <code>scheme = plain</code> 。它们均已过时, 仅因兼容性而保留。
<hr/> space nospace <hr/>	新版本宏包 <code>space</code> 选项增加真假值参数。(5.3 节) <code>nospace</code> 选项相当于 <code>space=false</code> , 成为过时选项。
<hr/> fancyhdr <hr/>	新版本宏包中总是自动处理对 <code>fancyhdr</code> 宏包的兼容性, 而由用户自己使用 <code>\usepackage</code> 载入 <code>fancyhdr</code> 宏包。 <code>fancyhdr</code> 选项过时, 因兼容性保留, 功能是载入 <code>fancyhdr</code> 宏包。

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

例 24

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

11.3 CT_EX 1.02c 以后的 SVN 开发版

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

CT_EX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 C_TE_X 2.2 之前的版本

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

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

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

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

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

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

第 12 节 宏集依赖情况与手工安装方法

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

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

- `expl3`、`xparse` 和 `l3keys2e` 宏包。它们属于 `l3kernel` 和 `l3packages` 宏集, 它们还依赖
 - `etex` 宏包。
- `indentfirst` 宏包, 属于 `tools` 宏集。
- `everysel` 宏包, 属于 `ms` 宏集。
- `zhnumber` 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- `CJK` 宏集, 它的下划线功能依赖 `ulem` 宏包。
- `CJKpunct` 宏包。
- `xCJK2uni` 宏包。
- `zhmetrics` 宏包。
- `zhmCJK` 宏包。
- ➡ 以上是使用 `pdfLATEX` 或 `LATEX + DVIPDFMx` 的编译方式所需要的依赖项, 其中 `zhm-CJK` 是可选的。
- `xeCJK` 宏包, 它还依赖
 - `fontspec` 宏包, 它还依赖
 - * `euenc` 宏包。
 - * `xunicode` 宏包, 它还依赖 `tipa` 宏包。
- ➡ 以上是使用 `XYLATEX` 编译时的依赖项。
- `LuaTeX-ja` 宏集, 它还依赖
 - `oberdiek` 宏集。
 - `xkeyval` 宏包。
 - `lualibs` 宏包。
 - `luaotfload` 宏包, 它还依赖 `luatexbase` 宏包。

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

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

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

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

若要手工安装, 请遵循如下步骤:

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

第 13 节 开发人员

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

目前比较活跃的开发维护人员是刘海洋、李清和黄晨成。

第 14 节 参考文献

- [1] DONALD ERVIN KNUTH. *The \TeX book, Computers & Typesetting*, volume A. Addison-Wesley, 1986
- [2] FRANK MITTELBAACH and MICHEL GOOSSENS. *The \LaTeX Companion*. Tools and Techniques for Computer Typesetting. Boston: Addison-Wesley, second edition, 2004

第 15 节 代码实现

1 <@@=ctex>

宏包载入检查。

```

2 <*class|ctex>
3 \tl_const:Nx \c__ctex_version_tl
4 { \cs_if_exist_use:cF { ver@ \@currname . \@currentext } { 9999/99/99 } }
5 <*class>
6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl
7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl
8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl
9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl
10 </class>
11 <*ctex>
12 \msg_new:nnnn { ctex } { subpackage-loaded }
13 { Package~`#1'~can~not~be~loaded~with~`ctex'. }
14 {
15   `#1'~is~actually~a~part~of~`ctex'.\\
16   It~is~not~necessary~to~load~it~separately.
17 }
18 \ifpackageloaded { ctexsize }
19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }
20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }
21 \ifpackageloaded { ctexheading }
22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }
23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }
24 </ctex>
25 </class|ctex>
26 <*class|style>
27 \RequirePackage { xparse , l3keys2e }
28 </class|style>
29 <*class|ctex>

```

检查 expl3 和 l3keys2e 的版本。

```

30 \msg_new:nnnn { ctex } { l3-too-old }
31 { Support~package~`#1'~too~old. }
32 {
33   Please~update~an~up~to~date~version~of~the~bundles\\
34   `l3kernel'~and~`l3packages'\\
35   using~your~TeX~package~manager~or~from~CTAN.
36 }
37 \ifpackagelater { expl3 } { 2015/12/20 } { }
38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
39 <*class>
40 \ifpackagelater { l3keys2e } { 2015/12/20 } { }
41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
42 </class>

```

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

```

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

```

(End definition for \c__ctex_engine_file_str.)

52 </class|ctex>

53 <*class|ctex|ctexheading>


```

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

```

15.1 内部函数与变量

```

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

```

(End definition for \l__ctex_tmp_tl and others.)

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

```

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

```

(End definition for \ctex_file_wrapper:nnn.)

`\ctex_file_input:n` 输入文件。

```

75 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
76 { \ctex_file_wrapper:nnn { } { \file_input:n {#1} } { } }

```

(End definition for \ctex_file_input:n.)

`\ctex_scheme_input:n` 输入 scheme 文件。先查找当前文档类下的 (scheme),找不到再查找一般的文件。

```

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

```

(End definition for \ctex_scheme_input:n.)

`\g__ctex_section_depth_flag` 若大于 3,则 `\paragraph` 和 `\subparagraph` 标题单独占一行;若为 3,则 `\paragraph` 单独占一行。

```

92 <!*beamer>
93 \cs_new_eq:NN \g__ctex_section_depth_flag \c_two
94 <!/beamer>

```

(End definition for \g__ctex_section_depth_flag.)

```

95 </class|ctex|ctexheading>
96 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

97 \msg_new:nnnn { ctex } { package-too-old }
98 { Support~package~`#1'~too~old. }
99 {
100   Please~update~an~up~to~date~version~of~the~package~`#1'\
101   using~your~TeX~package~manager~or~from~CTAN.
102 }

```

\ifctexpdf 在 zhmetrics 映射文件中使用。

```

103 \sys_if_output_pdf:TF
104 { \cs_new_eq:NN \ifctexpdf \if_true: }
105 { \cs_new_eq:NN \ifctexpdf \if_false: }

```

(End definition for \ifctexpdf.)

\ctex_if_preamble:TF 测试是否在 L^AT_EX 2_ε 的导言区。在宏包内部初始为真,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。

```

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

```

(End definition for \ctex_if_preamble:TF.)

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

```

108 \cs_new:Npn \ctex_set_default_ccwd:Nn #1#2
109 {
110   \dim_compare:nNnTF
111     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
112     { \tl_set:Nn #1 {#2} }
113     { \tl_set:Nn #1 { #2 \ccwd } }
114 }

```

(End definition for \ctex_set_default_ccwd:Nn.)

\ctex_default_pt:n 最新版本的 expl3 已经不允许 \dim_to_decimal:n 的参数带额外的单位。然而我们需要这个特性实现可展的 \@defaultunits。

```

115 \cs_new:Npn \ctex_default_pt:n #1
116 {
117   \exp_after:wN \__ctex_default_pt:w
118   \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
119 }
120 \use:x
121 {
122   \cs_new:Npn \exp_not:N \__ctex_default_pt:w
123     ##1 \tl_to_str:n { pt } ##2 \exp_not:N \q_stop
124     { ##1 \tl_to_str:n { pt } }
125 }

```

(End definition for \ctex_default_pt:n.)

`\l__ctex_encoding_tl` (pdf)LaTeX 初始化编码为 GBK, 其它则是 UTF8。

```
126 \tl_new:N \l__ctex_encoding_tl
127 \tl_set:Nx \l__ctex_encoding_tl
128 { \sys_if_engine_pdftex:TF { GBK } { UTF8 } }
```

(End definition for `\l__ctex_encoding_tl`.)

`\g__ctex_zhmCJK_bool` 是否使用 zhmCJK 宏包。

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

(End definition for `\g__ctex_zhmCJK_bool`.)

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

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

(End definition for `\l__ctex_autoindent_tl`.)

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

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

(End definition for `\ctex_if_autoindent_touched:F`.)

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

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

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

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

```
133 <class>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
134 <ctex>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfPackage }
```

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

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

```
135 <*class>
136 \clist_new:N \g__ctex_std_options_clist
137 </class>
```

(End definition for `\g__ctex_std_options_clist`.)

对无效选项给出警告。

```
138 \msg_new:nnn { ctex } { invalid-option }
139 { Option~`\l_keys_key_tl'~is~invalid~in~current~mode. }
140 \msg_new:nnn { ctex } { invalid-value }
141 { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_tl'. }
```

对过时选项或命令给出警告。

```
142 \msg_new:nnn { ctex } { deprecated-option }
143 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
144 \msg_new:nnn { ctex } { deprecated-command }
145 { Command~ #1 is~ deprecated.\\ #2 }
146 \msg_new:nnn { ctex } { deprecated-environment }
147 { Environment~`#1'~is~ deprecated.\\ #2 }
```

```
148 </class|ctex>
```

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

```

149 <*class|ctex|ctexsize>
150 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one
151 </class|ctex|ctexsize>

```

(End definition for `\g__ctex_font_size_flag`.)

15.2 宏包选项

```

152 <*class|style>
153 \keys_define:nn { ctex / option }
154 {
155 </class|style>

```

zihao

```

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

```

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

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

```

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

```

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

autoindent 自动调整段落的首行缩进功能。

```

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

```

```

195     {
196       \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
197       \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
198     } ,

```

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

indent 仅为兼容性保留, 已过时。

```

199     indent .code:n =
200     {
201       \msg_warning:nnn { ctex } { deprecated-option }
202       {
203         The~ functionality~ has~ been~ removed.\\
204         It's~ better~ to~ set~ the~ heading~ styles~ via~ `afterindent'~
205         options.~
206       }
207     } ,
208     indent .value_forbidden:n = true ,
209     noindent .code:n =
210     {
211       \msg_warning:nnn { ctex } { deprecated-option }
212       {
213         The~ functionality~ has~ been~ removed.\\
214         It's~ better~ to~ set~ the~ heading~ styles~ via~ `afterindent'~
215         options.~
216       }
217     } ,
218     noindent .value_forbidden:n = true ,

```

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

GBK

UTF8

```

219     GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
220     UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
221     GBK .value_forbidden:n = true ,
222     UTF8 .value_forbidden:n = true ,

```

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

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

```

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

```

```

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

```

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

zhmap

```

254 zhmap .choice: ,
255 zhmap .default:n = { true } ,
256 zhmap / zhmCJK .code:n =
257 {
258     \bool_gset_true:N \g__ctex_zhmCJK_bool
259     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
260 },
261 zhmap / true .code:n =
262 {
263     \bool_gset_false:N \g__ctex_zhmCJK_bool
264     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
265 },
266 zhmap / false .code:n =
267 {
268     \bool_gset_false:N \g__ctex_zhmCJK_bool
269     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
270 },
271 nozhmap .code:n =
272 {
273     \msg_warning:nnn { ctex } { deprecated-option }
274     { Option~`zhmap=false'~ is~ set. }
275     \keys_set:nn { ctex / option } { zhmap = false }
276 },
277 nozhmap .value_forbidden:n = true ,

```

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

punct 设置标点符号输出格式。

```

278 punct .tl_set:N = \l__ctex_punct_tl ,
279 punct .default:n = { quanjiao } ,
280 punct .initial:n = { quanjiao } ,
281 nopunct .code:n =
282 {
283     \msg_warning:nnn { ctex } { deprecated-option }
284     { Option~`punct=plain'~ is~ set. }
285     \keys_set:nn { ctex / option } { punct = plain }
286 },
287 nopunct .value_forbidden:n = true ,

```

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

space

```

288 space .choices:nn =
289 { true , auto , false }
290 {
291     \exp_args:Nx \ctex_at_end:n
292     { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
293 },
294 space .default:n = { true } ,
295 nospace .code:n =

```

```

296     {
297       \msg_warning:nnn { ctex } { deprecated-option }
298       { Option~`space=false'~ is~ set. }
299       \keys_set:nn { ctex / option } { space = false }
300     } ,
301     nospace .value_forbidden:n = true ,

```

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

heading

```

302     heading .bool_set:N = \l__ctex_heading_bool ,

```

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

```

303 </class|ctex>
304 <*class|ctex|ctexheading>

```

sub3section

sub4section

```

305 <!*beamer>
306     sub3section .code:n =
307     { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
308     sub4section .code:n =
309     { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
310     sub3section .value_forbidden:n = true ,
311     sub4section .value_forbidden:n = true ,
312 <!/beamer>

```

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

scheme

```

313     scheme .tl_set:N = \l__ctex_scheme_tl ,
314 <*ctexheading>
315     scheme .default:n = { plain } ,
316     scheme .initial:n = { plain }
317   }
318 </ctexheading>
319 <!*ctexheading>
320     scheme .default:n = { chinese } ,
321     scheme .initial:n = { chinese } ,
322 </!ctexheading>

323 </class|ctex|ctexheading>
324 <*class|ctex>

325     cap .code:n =
326     {
327       \msg_warning:nnn { ctex } { deprecated-option }
328       { Option~`scheme = chinese' ~ is~ set. }
329       \keys_set:nn { ctex / option } { scheme = chinese }
330     } ,
331     nocap .code:n =
332     {
333       \msg_warning:nnn { ctex } { deprecated-option }
334       { Option~`scheme = plain' ~ is~ set. }
335       \keys_set:nn { ctex / option } { scheme = plain }
336     } ,
337     cap .value_forbidden:n = true ,
338     nocap .value_forbidden:n = true ,

```

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

fntef 这些都是过时的宏包兼容选项,原选项功能总是打开的。
fancyhdr
hyperref

```

339     fntef .code:n =

```



```

340     {
341       \msg_warning:nnn { ctex } { deprecated-option }
342       { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
343     } ,
344     fancyhdr .code:n =
345     {
346       \msg_warning:nnn { ctex } { deprecated-option }
347       { `fancyhdr'~ package~ is~ loaded. }
348       \RequirePackage { fancyhdr }
349     } ,
350     hyperref .code:n =
351     {
352       \msg_warning:nnn { ctex } { deprecated-option }
353       { `hyperref'~ package~ will~ be~ loaded. }
354       \ctex_at_end:n
355       {
356         \cs_if_exist:NF \hypersetup
357         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
358       }
359       \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
360     } ,
361   }

```

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

```

362 </class>ctex>
363 <*class>ctex|ctexsize>

```

10pt 使 **ctex** 和 **ctexsize** 可以接受文档类的全局选项,不修改默认字体大小。在文档类下还将参数
 11pt 传给标准文档类。
 12pt

```

364 \tl_clear:N \l__ctex_tmp_tl
365 \clist_map_inline:nn
366 {
367   10pt , 11pt , 12pt ,
368   8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
369 }
370 {
371   \tl_put_right:Nn \l__ctex_tmp_tl
372   {
373     #1 .code:n =
374     <!*class>
375     { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
376     </!class>
377     <*class>
378     {
379       \cs_gset_eq:NN \g__ctex_font_size_flag \c_two
380       \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
381     } ,
382     </class>
383     #1 .value_forbidden:n = true ,
384   }
385 }
386 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
387 \tl_clear:N \l__ctex_tmp_tl

```

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

将未知选项传给标准文档类。

```

388 <*class>
389 \keys_define:nn { ctex / option }
390 {
391   unknown .code:n =
392   { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
393 }
394 </class>

```

```

395 <|ctexsize>\ctex_file_input:n { ctexopts.cfg }
396 </class|ctex|ctexsize>

397 <*class|style>
398 \ProcessKeysOptions { ctex / option }
399 </class|style>

400 <*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。

```

401 \if_case:w \g__ctex_font_size_flag
402 \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
403 \or:
404 \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
405 \fi:

```

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

```

406 <*article>
407 \tl_const:Nn \c__ctex_class_tl { article }
408 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
409 \LoadClass { article }
410 </article>
411 <*book>
412 \tl_const:Nn \c__ctex_class_tl { book }
413 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
414 \LoadClass { book }
415 </book>
416 <*report>
417 \tl_const:Nn \c__ctex_class_tl { report }
418 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
419 \LoadClass { report }
420 </report>
421 <*beamer>
422 \tl_const:Nn \c__ctex_class_tl { beamer }
423 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
424 \LoadClass { beamer }
425 </beamer>

426 </class>

```

15.3 用户设置接口

`\ctexset`

```

427 <*class|ctex|ctexheading>
428 \NewDocumentCommand \ctexset { } { \keys_set:nn { ctex } }
429 </class|ctex|ctexheading>

```

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

`\CTEXsetup` 过时命令。
`\CTEXoptions`

```

430 <*class|ctex>
431 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
432 {
433   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
434   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
435   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
436 }
437 \NewDocumentCommand \CTEXoptions { +o }
438 {
439   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
440   { \ctexset~ {~ #1~ }~ is~ set. }
441   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
442 }
443 </class|ctex>

```

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

15.4 特定引擎支持与设置

15.4.1 ctex-engine-pdfTeX.def

`\ctex_set_zhmap:n` 设置 upTeX 字体映射,同时作用于 `\AtBeginDvi` 与 `\AtBeginShipoutFirst`。该宏对 pdfTeX 和 upTeX 均有用。

```

444 <*pdfTeX|uptex>
445 \cs_new_protected_nopar:Npn \ctex_set_zhmap:n #1
446 {
447   \AtBeginDvi { #1 }
448   \ctex_at_end_package:nn { atbegshi }
449   { \AtBeginShipoutFirst { #1 } }
450 }
451 \@onlypreamble \ctex_set_zhmap:n
452 </pdfTeX|uptex>

```

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

```

453 <*pdfTeX>

```

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

```

454 \seq_new:N \c__ctex_cmap_encoding_seq
455 \seq_set_from_clist:Nn \c__ctex_cmap_encoding_seq
456 { C19 , C10 , C00 , C09 , C40 , C60 }

```

(End definition for `\c__ctex_cmap_encoding_seq`.)

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

```

457 \cs_new_protected_nopar:Npn \ctex_family_cmap:nn #1#2
458 {
459   \cs_if_free:cF { #1 + #2 }
460   {
461     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
462     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
463   }
464 }
465 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
466 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn

```

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

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

```

467 \cs_new_protected_nopar:Npn \ctex_add_cmap:n #1
468 {
469   \cs_if_free:NF \CJK@plane
470   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
471 }
472 \cs_new_protected_nopar:Npn \ctex_add_cmap:Nn #1#2
473 {
474   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
475   #1
476 }
477 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
478 \cs_new_protected_nopar:Npn \__ctex_save_cmap:Nn #1#2
479 {
480   \tl_set:Nx \l__ctex_tmp_tl { \str_lower_case:n {#2} \CJK@plane }
481   \tex_immediate:D \pdfTeX_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
482   \cs_new_protected_nopar:Npx #1

```

```

483     {
484         \exp_not:N \pdfTEX_pdffontattr:D \exp_not:N \tex_font:D
485         { /ToUnicode ~ \int_use:N \pdfTEX_pdflastobj:D \c_space_tl 0 ~ R }
486     }
487 }

```

(End definition for \ctex_add_cmap:n.)

`\DeclareFontFamily` 只在 pdf_{La}T_EX 下加 CMap。如 cmap 宏包被引入,则不重复设置。

```

488 \group_begin:
489 \char_set_catcode_other:N \#
490 \sys_if_output_pdf:TF
491 {
492     \group_end:
493     \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
494     { \CTEX@Family@CMap {#1} {#2} }
495     {
496         \ctex_at_end_package:nn { cmap }
497         { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
498     }
499     { \ctex_patch_failure:N \DeclareFontFamily }
500 }
501 { \group_end: }

```

(End definition for \DeclareFontFamily.)

首先检查选项,决定是否载入 zhmCJK 宏包。

```

502 \if_bool:N \g__ctex_zhmCJK_bool
503 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
504 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

505 \else:
506 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
507 { \RequirePackage { CJK } }
508 { \RequirePackage { CJKutf8 } }
509 \RequirePackage { CJKpunct , CJKspace }

```

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

```

510 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
511 {
512     \tl_set:Nn \CJKrmdefault {#1}
513     \tl_set:Nn \CJKsfdefault {#2}
514     \tl_set:Nn \CJKttdefault {#3}
515     \ctex_set_zhmap:n { \file_input:n {#4} }
516 }
517 \@onlypreamble \ctex_load_zhmap:nnnn

```

(End definition for \ctex_load_zhmap:nnnn.)

```

518 \tl_if_exist:NF \CJKfamilydefault
519 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
520 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
521 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
522 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
523 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
524 { \CJKfamily { \CJKrmdefault } }
525 { }
526 { \ctex_patch_failure:N \rmfamily }
527 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
528 { \CJKfamily { \CJKsfdefault } }
529 { }
530 { \ctex_patch_failure:N \sffamily }
531 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }

```

```

532 { \CJKfamily { \CJKttdefault } }
533 { }
534 { \ctex_patch_failure:N \ttfamily }
535 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
536 { \CJKfamily { \CJKfamilydefault } }
537 { \cs_set_eq:NN \reset@font \normalfont }
538 { \ctex_patch_failure:N \normalfont }

```

zhmCJK 判断结束。

```
539 \fi:
```

`\ctex_CJK_input:n` `\CJK@input` breqn 包可能会在正文中将 `ˆ` 的 `\catcode` 改为 12 或 13, 这将破坏 CJK 对汉字的首字节的定义 (`\CJK@loadBinding` 和 `\CJK@loadEncoding`)。因此需要确保载入 `.enc` 和 `.bdg` 文件时, `ˆ` 的 `\catcode` 为 7。

```

540 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
541 {
542   \ctex_file_wrapper:nnn
543   {
544     \char_set_catcode_other:n { 60 } % <
545     \char_set_catcode_math_superscript:n { 94 } % ˆ
546     \int_set_eq:NN \tex_endlinechar:D \c_minus_one
547   }
548   { \file_input:n {#1} }
549   {
550     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
551     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
552     \int_set:NN \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
553   }
554 }
555 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

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

`\ctex_plane_to_utfxvibe:Nn` `\CJK@surr` fancyhdr 宏包的 `\nouppercase` 会将 `\uppercase` 定义为 `\relax`, 而 `\CJK@surr` 需要用它将 `\CJK@plane` 转化成大写字母, 这就造成了冲突¹⁶。我们在这里给出 `\CJK@surr` 的一个不依赖 `\uppercase` 的实现。

```

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

```

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

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

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

```

577 \AtBeginDocument
578 {
579     \str_if_eq_x:nnF { \l__ctex_punct_tl } { quanjiao }
580     { \punctstyle { \l__ctex_punct_tl } }
581 }

```

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

```

582 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

启用中文字符功能。GBK 编码时, 将汉字的首字节设置为活动字符, 并对这些字符初始化; UTF8 编码时, 上游宏包已经处理好。 `\CJK@makeActive` 应该先于 `ctex-name-gbk.cfg` 等文件的载入。注意 `\CJK@loadBinding` 需要调用补丁后的 `\CJK@input`。使用 `zhmCJK` 时, 此功能已经被启用。

```

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

```

在导言区结束时调用 `\CJK@envStart` 启用完整的中文功能。

`\CJK@envStart` 的定义是

```

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

```

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

```

589 \exp_args:Nx \ctex_at_end_preamble:n
590 {
591     \exp_not:N \CJK@envStart
592     { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
593     \exp_not:N \CJKtilde
594 }

```

`zhmCJK` 判断结束。

```

595 \fi:

```

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

```

596 <@@=>
597 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces

```

恢复名字空间,要把它放在一个 macrocode 环境中,l3doc 才能正确工作。

```
598 <@@=ctex>
```

(End definition for \ctex_auto_ignorespaces:.)

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

```
599 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
600 {
601   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
602     { \cs_set_eq:NN \CJK@ignorespaces #1 }
603   \ctex_set_ignorespaces:
604 }
605 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
606 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }
```

(End definition for \ctex_ignorespaces_case:N and \ctex_set_ignorespaces:.)

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

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

(End definition for \CJKhook.)

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

```
608 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
609 {
610   \clist_map_inline:Nn \c__ctex_punct_family_clist
611     {
612       \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
613       {
614         \cs_set_eq:cc
615           { CJKpunct@ ##1 @spaces }
616           { c__ctex_ #1 ##1 _punct_spaces_tl }
617       }
618     }
619 }
620 \clist_const:Nn \c__ctex_punct_family_clist
621 {
622   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
623   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
624 }
```

(End definition for \ctex_punct_set:n.)

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

```
625 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
626 {
627   \cs_if_free:cF { CJKpunct@ #2 @spaces }
628   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
629 }
```

(End definition for \ctex_punct_map_family:nn.)

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

```
630 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
631 {
632   \clist_map_inline:nn {#1}
```



```

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

```

(End definition for \ctex_punct_map_bfseries:nn.)

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

```

647 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
648 {
649     \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
650     \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
651     \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
652     \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
653     \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
654     \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
655 }

```

(End definition for \ctex_punct_map_itshape:nn.)

\ctex_punct_space:nn 定义标点的边界信息。

\ctexspadef

```

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

```

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

载入边界信息文件。

```

659 \ctex_file_input:n { ctexspa.def }

```

660 </pdfTeX>

15.4.2 ctex-engine-xetex.def

```

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

```

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。

```

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

```

671 </xetex>

15.4.3 ctex-engine-luatex.def

672 <*luatex>

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

```

673 \msg_new:nnn { ctex } { luatexja-loaded }
674 {
675   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
676   Loading~file~`#1'~will~abort!
677 }
678 \@ifpackageloaded { luatexja }
679 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
680 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }

681 \RequirePackage { luatexja }
682 \@ifpackagelater { luatexja } { 2015/09/21 } { }
683 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }

684 \RequirePackage { fontspec }
685 \@ifpackagelater { fontspec } { 2014/05/25 } { }
686 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

15.4.3.1 LuaTeX-j_a 的默认设置

```

687 \ExplSyntaxOff

```

以下设置抄录自 ltjdefs.sty。

```

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

```

以下设置抄录自 ltj-latex.sty。

```

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

```

15.4.3.2 LuaTeX-j_a 的补丁

```

717 <@@=ctex_ltx>

```

在 L_AT_EX 下, LuaTeX-j_a 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。

```

718 \RequirePackage { xunicode-addon }
719 \AtBeginUTFCommand
720 {

```

```

721 \group_begin:
722 \lua_now_x:n { tex.globaldefs = 0 }
723 \ltj@allalchar
724 }
725 \AtEndUTFCommand { \group_end: }

```

对 `fontspec` 沿用 `LuaTeX-j`a 的补丁。

```

726 \RequirePackage { lltjp-fontspec }

```

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

```

727 \cs_new_protected_nopar:Npn \__ctex_ltj_um_define_math_chars:
728 {
729 \group_begin:
730 \cs_set_protected:Npn \__um_sym:nnn ##1##2##3
731 {
732 \tl_if_in:nnT
733 {
734 \mathord \mathalpha \mathbin \mathrel
735 \mathpunct \mathop \mathfence
736 }
737 { ##3 }
738 { \__ctex_ltj_um_char:Nn ##2 { ##1 } }
739 }
740 \__um_input_math_symbol_table:
741 \group_end:
742 }
743 \cs_new_protected:Npn \__ctex_ltj_um_char:Nn #1#2
744 {
745 \__ctex_ltj_um_char_aux:Nx #1 { \char_generate:nn {#2} { 12 } }
746 \ltjsetmathletter {#2}
747 }
748 \cs_new_protected:Npn \__ctex_ltj_um_char_aux:Nn #1#2
749 {
750 \cs_gset_protected_nopar:Npn #1
751 {
752 \mode_if_math:TF
753 {#2}
754 {
755 {
756 \lua_now_x:n { tex.globaldefs = 0 }
757 \ltj@allalchar #2
758 }
759 }
760 }
761 }
762 \cs_generate_variant:Nn \__ctex_ltj_um_char_aux:Nn { Nx }
763 \ctex_at_end_package:nn { unicode-math }
764 {
765 \cs_set_eq:NN \__um_define_math_chars: \__ctex_ltj_um_define_math_chars:
766 \@ifpackagelater { unicode-math } { 2015/06/28 }
767 {
768 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
769 \cs_set_protected_nopar:Npn \ctex_ltj_math_group_hook:
770 { \__um_switchto_literal: }
771 }
772 { }
773 }

```

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

```

774 \ctex_at_end_package:nn { listings }
775 {
776 \use:x
777 {

```

```

778         \exp_not:N \RequirePackage { lltpj-listings }
779         \tl_set:Nn \exp_not:N \lstlistingname
780         { \exp_not:o { \lstlistingname } }
781         \tl_set:Nn \exp_not:N \lstlistlistingname
782         { \exp_not:o { \lstlistlistingname } }
783     }
784 }

```

15.4.3.3 字体切换方式

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

```

785 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
786 {
787     \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
788     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
789 }
790 \tl_new:N \CJK@family
791 \tl_new:N \l__ctex_ltj_current_font_tl
792 \tl_set:Nn \l__ctex_ltj_current_font_tl
793 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

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

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

```

794 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
795 {
796     \group_begin:
797     \tl_set_eq:NN \f@encoding \CJK@encoding
798     \tl_set_eq:NN \f@family \CJK@family
799     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
800     \ctex_ltj_pickup_font:
801     \group_end:
802     \font@name
803     \__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。

```

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

```

(End definition for `__ctex_ltj_select_font_aux:`.)

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

```

814 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
815 {
816     \exp_after:wN \cs_if_exist:NF \font@name
817     {
818         \group_begin:
819         \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
820         \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
821         \define@newfont
822         \group_end:

```

```

823     }
824   }
825   \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

(End definition for \ctex_ltj_pickup_font:.)

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

```

%%\font\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX(false, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\protected\def\globaljfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX(true, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\def\ltj@@jfont{\directlua{luatexja.jfont.jfontdefY()}}

```

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

```

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

```

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

```

826 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
827   {
828     \get@external@font
829     \ctex_ltj_if_alterate_shape_exist:nT { \curr@fontshape }
830     {
831       \tl_set:Nx \external@font
832       { \exp_after:wN __ctex_ltj_patch_external_font:w \external@font }
833     }
834     \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。

```

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

```

(End definition for \ctex_ltj_extract_font:.)

\ctex_ltj_subst_font: \do@subst@correction 在设置通过 sub 或者 ssub 函数定义的字体时会用到。如果没有设置 SlantedFont, fontspec 会设置 \itdefault 作为 \sldefault 的替代字形, 因而会用到这个函数。它的本来定义是:

```

\def\do@subst@correction{%
  \xdef\subst@correction{%
    \font@name
    \global\expandafter\font
    \csname \curr@fontshape/\f@size\endcsname
    \noexpand\fontname\font
    \relax}%
  \aftergroup\subst@correction
}

```

我们在这里不需要定义新字体, 而是设置对应字体的命令。

```

840 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
841   {

```

```

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

```

(End definition for \ctex_ltj_subst_font:.)

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

```

860 \prg_new_conditional:Npnn \ctex_ltj_if_alter_nate_shape_exist:n #1 { T , F , TF }
861 {
862   \lua_now_x:n { luatexja.jfont.does_alt_set (' \lua_escape_x:n { #1 } ') }
863   \prg_return_true: \else: \prg_return_false: \fi:
864 }

```

(End definition for \ctex_ltj_if_alter_nate_shape_exist:nTF.)

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

```

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

```

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

```

865 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
866 { #1 \lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for __ctex_ltj_patch_external_font:w.)

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

```

867 \cs_new_protected_nopar:Npn \ctex_ltj_select_alter_nate_font:
868 {
869   \ctex_ltj_if_alter_nate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
870   {
871     \lua_now_x:n
872     {
873       luatexja.jfont.output_alt_font_cmd
874       ('y', ' \lua_escape_x:n { \l__ctex_ltj_current_shape_tl } ')
875     }
876     \lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size' ) }
877   }
878 }
879 \tl_new:N \l__ctex_ltj_current_shape_tl
880 \tl_set:Nn \l__ctex_ltj_current_shape_tl
881 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for \ctex_ltj_select_alter_nate_font:.)

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

```

882 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
883 {
884   \cs_if_exist:cF { #1/\f@size }
885   {
886     \group_begin:
887     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
888     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
889     \ctex_ltj_pickup_font:
890     \group_end:
891     \__ctex_ltj_pop_fontname:
892   }
893 }

```

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

894 `<@@=`

`\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。

```

895 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
896 {
897   \ltj@@getjfontnumber #1
898   \lua_now_x:n
899   {
900     luatexja.jfont.pickup_alt_font_b
901     ( \int_use:N \ltj@tempcntc, '\lua_escape_x:n {#2}' )
902   }
903 }

```

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

15.4.3.4 数学字体族

904 `<@@=ctex_ltj`

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

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

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

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

```

905 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
906 {
907   \lua_now_x:n
908   { luatexja.jfont.is_kenc( string.match('\lua_escape_x:n {#1}', '[~/]+') ) }
909   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
910 }

```

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

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

```

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

```


(End definition for \ctex_ltj_if_jfont_math:NTF.)

918 <@@=>

\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。

```

919 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
920 {
921   \ctex_ltj_if_jfont:NTF { \token_to_str:N #2 }
922   { \ctex_ltj_get_and_define_fonts_ja:nN }
923   { \ctex_ltj_get_and_define_fonts_al:nN }
924   {#1} #2
925 }
926 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
927 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
928 \cs_new_protected_nopar:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
929 {
930   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
931   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
932   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
933   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
934   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
935   \ctex_ltj_pickup_font:
936   \tl_put_right:Nx \math@fonts
937   {
938     \ltj@setpar@global
939     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
940     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
941     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
942   }
943 }
```

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

944 <@@=ctex_ltj>

\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。

```

945 \cs_new_protected_nopar:Npn \ctex_ltj_use_math_group:Nn #1#2
946 {
947   \mode_if_math:T
948   {
949     \math@bgroup
950     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
951     \ctex_ltj_math_group_hook:
952     \ctex_ltj_if_jfont_math:NTF #1
953     { \jfam } { \mathgroup } #2 \scan_stop:
954     \math@egroup
955   }
956 }
957 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
958 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
```

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

15.4.3.5 字体族的定义与使用

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl

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

```

959 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
960 {
961   \str_if_eq:nnTF {#1} { plain }
962   { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
963   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
964 }
```

```

965 \tl_new:N \l__ctex_ltj_jfm_tl
966 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
967 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

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

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

```

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

```

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

这是 $\text{\luatexja-fontspec}$ 中新增的一些字体选项。

```

990 \newfontfeature { CID } { cid = #1 }
991 \newfontfeature { JFM } { jfm = #1 }
992 \newfontfeature { JFM-var } { jfmvar = #1 }

```

在新版本的 \fontspec 中, $\text{_fontspec_namewrap:n}$ 变成了私有函数。

```

993 \keys_define:nn { fontspec-parse-external }
994 {
995   NoEmbed .code:n =
996   { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
997 }
998 \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。

```

999 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1000 {
1001   \group_begin:
1002   \clist_clear:N \l__ctex_ltj_char_range_clist
1003   \seq_clear:N \l__ctex_ltj_alternate_seq
1004   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1005   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1006   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1007   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1008   {
1009     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1010     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1011     {#1} { \l__ctex_ltj_font_options_clist }
1012     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist

```

```

1013     \l__ctex_ltj_font_options_clist
1014     \l__ctex_ltj_gset_family_cs:nn {#1} {#3}
1015   }
1016   \group_end:
1017 }
1018 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
1019 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
1020 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1021 \clist_new:N \l__ctex_ltj_font_options_clist
1022 \cs_new_protected_nopar:Npn \l__ctex_ltj_use_global_options:N #1
1023 {
1024   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1025   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
1026 }

```

(End definition for \ctex_ltj_set_family:nnn.)

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

```

1027 \prop_new:N \g__ctex_ltj_family_name_prop
1028 \prop_new:N \g__ctex_ltj_family_font_name_prop
1029 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

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

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

```

1030 \cs_new_protected_nopar:Npn \l__ctex_ltj_check_family:n #1
1031 {
1032   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1033   {
1034     \cs_undefine:c { \l__ctex_ltj_family_csname:n {#1} }
1035     \cs_undefine:c { \l__ctex_ltj_altername_cs:n {#1} }
1036     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1037     {
1038       \use:c { \l__ctex_ltj_altername_cs:n { clear / #1 } }
1039       \cs_undefine:c { \l__ctex_ltj_altername_cs:n { clear / #1 } }
1040       \cs_undefine:c { \l__ctex_ltj_altername_cs:n { reset / #1 } }
1041       \prop_gremove:Nn \g__ctex_ltj_reset_altername_prop {#1}
1042     }
1043     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1044   }
1045 }
1046 \tl_new:N \l__ctex_ltj_tmp_tl
1047 \msg_new:nnn { ctex } { redefine-family }
1048 { Redefining~CJKfamily~\l__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

(End definition for \l__ctex_ltj_check_family:n.)

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

```

1049 \cs_new_protected_nopar:Npn \l__ctex_ltj_gset_family_cs:nn #1#2
1050 {
1051   \cs_gset_protected_nopar:cpx { \l__ctex_ltj_family_csname:n {#1} }
1052   {
1053     \group_begin:
1054     \l__ctex_ltj_change_encoding:
1055     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1056     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1057     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1058     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1059     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1060     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1061     \exp_not:N \g__ctex_ltj_fontspec_family_tl

```

```

1062         \__ctex_ltj_set_alternate_family:n {#1}
1063     \group_end:
1064 }
1065 }
1066 \tl_new:N \l__ctex_ltj_base_family_tl
1067 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1068 \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1069 \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1070 {
1071     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1072     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1073     \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1074     \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1075 }
1076 \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

(End definition for __ctex_ltj_gset_family_cs:nn.)

\CJKfamily 切换字体。

```

1077 \NewDocumentCommand \CJKfamily { m }
1078 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1079 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1080 {
1081     \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1082     {
1083         \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1084         \selectfont
1085     }
1086     { \__ctex_ltj_family_unknown_warning:n {#1} }
1087 }
1088 \tl_new:N \l_ctex_ltj_family_tl
1089 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

(End definition for \CJKfamily.)

\ctex_ltj_family_if_exist:nTF 判断 CJK 字体族 #1 是否存在, 若存在则把实际族名保存到 #2 中。

```

1090 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1091 {
1092     \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1093     { \prg_return_true: }
1094     {
1095         \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1096         {
1097             \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1098             \prg_return_true:
1099         }
1100         { \prg_return_false: }
1101     }
1102 }
1103 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

__ctex_ltj_family_unknown_warning:n

```

1104 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1105 {
1106     \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1107     {
1108         \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1109         {
1110             \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1111             \msg_warning:nnn { ctex } { family-unknown } {#1}
1112         }

```

```

1113     }
1114 }
1115 \seq_new:N \g__ctex_ltj_unknown_family_seq
1116 \msg_new:nnn { ctex } { family-unknown }
1117 {
1118     Unknown~CJK~family~`__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1119     Try~to~use~`__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1120 }
1121 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1122 {
1123     \str_case_x:nnF {#1}
1124     {
1125         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1126         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1127         \CJKttdefault { \token_to_str:N \setCJKmonofont }
1128     }
1129     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1130     [...] \{ ... \}
1131 }
1132 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1133 {
1134     \str_case_x:nnF {#1}
1135     {
1136         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1137         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1138         \CJKttdefault { \token_to_str:N \CJKttdefault }
1139     }
1140     {#1}
1141 }

```

(End definition for `__ctex_ltj_family_unknown_warning:n`.)

`\ctex_ltj_fontspec:nn`

```

1142 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1143 {
1144     \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1145     { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1146     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1147     {
1148         \int_gincr:N \g__ctex_ltj_family_int
1149         \__ctex_ltj_fontspec:xnn
1150         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1151         {#1} {#2}
1152     }
1153 }
1154 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1155 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1156 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1157 {
1158     \bool_if:NT \l__ctex_ltj_add_alternate_bool
1159     {
1160         \cs_if_free:cF
1161         { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1162         {
1163             \cs_gset_eq:cc
1164             { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1165             { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1166             \cs_gset_eq:cc
1167             { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1168             { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1169         }
1170         \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1171     }
1172     \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1173     \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1174     \ctex_ltj_switch_family:n {#1}

```

```

1175 }
1176 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1177 \prop_new:N \g__ctex_ltj_fontspec_prop

(End definition for \ctex_ltj_fontspec:nn.)

\ctex_ltj_add_font_features:n
\ctex_ltj_add_font_features:nn
1178 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1179 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1180 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1181 {
1182   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1183   {#1} \l__ctex_ltj_tmp_tl
1184   {
1185     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1186     {#1} \l__ctex_ltj_font_options_clist
1187     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1188     \bool_set_true:N \l__ctex_ltj_add_alternate_bool
1189     \ctex_ltj_fontspec:xx
1190     { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1191     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1192   }
1193   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1194 }
1195 \bool_new:N \l__ctex_ltj_add_alternate_bool
1196 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1197 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1198 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1199 {
1200   \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1201   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1202 }

```

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

```

\setCJKfamilyfont
\newCJKfontfamily
\CJKfontspec
\addCJKfontfeatures
1203 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
1204 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1205 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
1206 {
1207   \tl_set:Nx \l__ctex_ltj_tmp_tl
1208   { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1209   \cs_new_protected_nopar:Npx #2
1210   { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1211   \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1212 }
1213 \NewDocumentCommand \CJKfontspec { O { } m }
1214 {
1215   \ctex_ltj_fontspec:xx {#1} {#2}
1216   \tex_ignorespaces:D
1217 }
1218 \NewDocumentCommand \addCJKfontfeatures { m }
1219 {
1220   \ctex_ltj_add_font_features:x {#1}
1221   \tex_ignorespaces:D
1222 }
1223 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

(End definition for \setCJKfamilyfont and others.)

```

\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1224 \NewDocumentCommand \setCJKmainfont { O { } m }
1225 {

```

```

1226 \ctex_ltj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1227 \normalfont
1228 }
1229 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1230 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1231 {
1232 \ctex_ltj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1233 \normalfont
1234 }
1235 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1236 {
1237 \ctex_ltj_set_family:xxx { \CJKttdefault } {#1} {#2}
1238 \normalfont
1239 }
1240 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1241 { \ctex_ltj_set_family:xxx { \c__ctex_ltj_math_tl } {#1} {#2} }
1242 \NewDocumentCommand \defaultCJKfontfeatures { m }
1243 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1244 \clist_new:N \g__ctex_ltj_default_features_clist
1245 \@onlypreamble \setCJKmainfont
1246 \@onlypreamble \setCJKsansfont
1247 \@onlypreamble \setCJKmonofont
1248 \@onlypreamble \setCJKmathfont
1249 \@onlypreamble \setCJKromanfont
1250 \@onlypreamble \defaultCJKfontfeatures

(End definition for \setCJKmainfont and others.)

1251 \tl_if_exist:NF \CJKfamilydefault
1252 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1253 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1254 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1255 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1256 \ctex_preto_cmd:NnnTF \rmfamily { \ExplSyntaxOff }
1257 { \CJKfamily { \CJKrmdefault } }
1258 { }
1259 { \ctex_patch_failure:N \rmfamily }
1260 \ctex_preto_cmd:NnnTF \sffamily { \ExplSyntaxOff }
1261 { \CJKfamily { \CJKsfdefault } }
1262 { }
1263 { \ctex_patch_failure:N \sffamily }
1264 \ctex_preto_cmd:NnnTF \ttfamily { \ExplSyntaxOff }
1265 { \CJKfamily { \CJKttdefault } }
1266 { }
1267 { \ctex_patch_failure:N \ttfamily }
1268 \ctex_preto_cmd:NnnTF \normalfont { \ExplSyntaxOff }
1269 { \CJKfamily { \CJKfamilydefault } }
1270 { \cs_set_eq:NN \reset@font \normalfont }
1271 { \ctex_patch_failure:N \normalfont }

```

在导言区末尾更新 \CJKfamilydefault。

```

1272 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

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

```

1273 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1274 {
1275 \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1276 {
1277 \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tpa_tl
1278 {
1279 \str_if_eq:x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1280 { \use:n }
1281 {
1282 \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tpa_tl
1283 { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }

```



```

1284         { \use:n }
1285     }
1286     {
1287         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1288         {
1289             \prop_map_break:n
1290             { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1291         }
1292     }
1293 }
1294 \normalfont
1295 \ctex_ltj_update_mathfont:
1296 }
1297 }

```

(End definition for \ctex_ltj_ensure_default_family:.)

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

```

1298 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1299 {
1300     \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1301     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1302     {
1303         \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1304         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1305     }
1306 }
1307 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1308 {
1309     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1310     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1311     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1312     \cs_if_free:cTF
1313     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1314     {
1315         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1316         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1317     }
1318     {
1319         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1320         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1321     }
1322 }

```

(End definition for \ctex_ltj_update_mathfont:.)

15.4.3.6 替代字体的设置

AlternateFont 设置替代字体的选项。

CharRange

```

1323 \keys_define:nn { ctex_ltj / fontspec }
1324 {
1325     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1326     AlternateFont .value_required:n = true ,
1327     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1328     CharRange .value_required:n = true
1329 }

```

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

\ctex_ltj_set_alternate_seq:n 我们使用 || 作为替代字体序列的分隔标志。它可能被设置为活动字符，为此需要先“消毒”，同时过滤掉空元素。

```

1330 \group_begin:

```

```

1331 \char_set_catcode_other:N \l
1332 \cs_set:Npn \__ctex_ltj_tmp:w #1
1333 {
1334   \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n ##1
1335   {
1336     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1337     {
1338       \tl_set:Nn \l__ctex_ltj_tmp_tl { ##1 }
1339       \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl {#1} { || }
1340       \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1341       \seq_set_filter:NnN \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1342       { ! \tl_if_blank_p:n { ##### } }
1343       \seq_concat:NNN \l__ctex_ltj_alternate_seq
1344       \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1345     }
1346   }
1347 }
1348 \char_set_catcode_active:N \l
1349 \__ctex_ltj_tmp:w { || }
1350 \group_end:
1351 \seq_new:N \l__ctex_ltj_tmp_seq
1352 \seq_new:N \l__ctex_ltj_alternate_seq

(End definition for \ctex_ltj_set_alternate_seq:n)

```

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

```

1353 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1354 {
1355   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1356   {
1357     \__ctex_ltj_check_family:n {#1}
1358     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1359     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1360     #3
1361   }
1362   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1363 }

(End definition for \ctex_ltj_set_alternate_family:nnF)

```

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

```

\ctex_ltj_save_alternate_seq:Nnnwn
1364 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1365 {
1366   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1367   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1368 }
1369 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1370 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1371 { m m m +O{ } m u{ \q_stop } }
1372 {
1373   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1374   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1375   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1376   \tl_if_blank:nTF {#5}
1377   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1378   {
1379     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1380     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1381   }
1382   \use:x
1383   {
1384     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1385     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1386     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1387     { \exp_not:o { \l__ctex_ltj_tmp_tl } }

```

```

1388     }
1389   }
1390   \clist_new:N \l__ctex_ltj_altername_options_clist

```

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

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

```

1391   \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_family:nn #1#2
1392   {
1393     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1394     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1395     \ctex_ltj_set_altername_family:coonn
1396     { \__ctex_ltj_altername_cs:n {#1} }
1397     { \l__ctex_ltj_char_range_clist }
1398     { \l__ctex_ltj_font_options_clist } {#2} {#1}
1399   }
1400   \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_family:Nnnnn #1#2#3#4#5
1401   {
1402     \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1403     { \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} }
1404     \ctex_ltj_save_altername_family:Nnnn #1 {#2} {#3} {#4}
1405   }
1406   \cs_generate_variant:Nn \ctex_ltj_set_altername_family:Nnnnn { coo }

```

(End definition for \ctex_ltj_set_altername_family:nn.)

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

```

1407   \cs_new_protected_nopar:Npn \ctex_ltj_save_altername_family:Nnnn #1#2#3#4
1408   {
1409     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1410     \cs_gset_protected_nopar:Npx #1
1411     { \exp_not:o { #1 \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} } }
1412   }

```

(End definition for \ctex_ltj_save_altername_family:Nnnn.)

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

```

1413   \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_family:nnn #1#2#3
1414   {
1415     \group_begin:
1416     \__ctex_ltj_change_encoding:
1417     \cs_set_eq:NN \CJKfamily \use_none:n
1418     \ctex_ltj_swap_cs:NN
1419     \DeclareFontShape@ \ctex_ltj_declare_altername_shape:nnnnnn
1420     \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1421     \fontspec_set_family:Nnn \l__ctex_ltj_altername_family_tl {#2} {#3}
1422     \group_end:
1423   }
1424   \tl_new:N \l__ctex_ltj_altername_family_tl

```

(End definition for \ctex_ltj_set_altername_family:nnn.)

\ctex_ltj_swap_cs:NN 交换两个控制序列的意义。

```

1425   \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1426   {
1427     \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1428     \cs_set_eq:NN #1 #2
1429     \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1430     \cs_undefine:N \__ctex_ltj_tmp:w
1431   }

```

(End definition for \ctex_ltj_swap_cs:NN.)

LTJFONTUID
_ctex_ltj_update_family_uid:N

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

```
1432 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1433 \cs_new_protected_nopar:Npn \_ctex_ltj_update_family_uid:N #1
1434 {
1435   \int_gincr:N \g__ctex_ltj_family_int
1436   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1437 }
1438 \int_new:N \g__ctex_ltj_family_int
```

(End definition for LTJFONTUID and _ctex_ltj_update_family_uid:N.)

\ctex_ltj_declare_alter_nate_shape:nnnnnn

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

```
1439 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1440 {
1441   \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1442   \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1443   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1444   { \l_fontspec_family_tl } {#3} {#4}
1445 }
```

(End definition for \ctex_ltj_declare_alter_nate_shape:nnnnnn.)

\ctex_ltj_set_alter_nate_shape:Nnnnnnn

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

```
1446 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1447 {
1448   \clist_map_inline:Nn #1
1449   {
1450     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1451     {
1452       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1453       \l__ctex_ltj_char_range_tl
1454     }
1455     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1456   }
1457   \_ctex_ltj_save_alter_nate_shape:cnn
1458   { \_ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1459   { luatexja.jfont.clear_alt_font_latex }
1460   { '\lua_escape_x:n { \CJK@encoding/#2/#3/#4 }' }
1461 }
```

(End definition for \ctex_ltj_set_alter_nate_shape:Nnnnnnn.)

\ctex_ltj_set_alter_nate_shape:nnn

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

```
1462 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1463 { m m > { \SplitArgument { \c_one } { -> } } m }
1464 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1465 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1466 {
1467   \ctex_ltj_set_alter_nate_shape:n
1468   {
1469     \IfNoValueTF {#4}
1470     { \int_eval:n {#3} , \int_eval:n {#3} , }
```

```

1471         {
1472             \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1473             \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1474         }
1475         '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1476         '\lua_escape_x:n { \CJK@encoding/#1 }'
1477     }
1478 }
1479 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1480 {
1481     \lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1482     \__ctex_ltj_save_alternate_shape:cn
1483     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1484     { luatexja.jfont.set_alt_font_latex } {#1}
1485 }

```

(End definition for \ctex_ltj_set_alternate_shape:nnn.)

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

```

1486 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1487 {
1488     \tl_map_inline:Nn #3
1489     {
1490         \ctex_ltj_set_alternate_shape:n
1491         {
1492             ##1 ,
1493             '\lua_escape_x:n { \CJK@encoding/#2 }' ,
1494             '\lua_escape_x:n { \CJK@encoding/#1 }'
1495         }
1496     }
1497 }

```

(End definition for \ctex_ltj_set_alternate_shape:nnN.)

__ctex_ltj_save_alternate_shape:Nnn 将实际设置的替换字形保存起来用于清除或恢复。暂时令 \l__ctex_ltj_base_family_tl 为 \scan_stop: 是让它不被展开,使得替换字体的设置可以在 \addCJKfontfeature 中直接使用。

```

1498 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1499 {
1500     \group_begin:
1501     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1502     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1503     \cs_set_eq:NN \lua_escape_x:n \scan_stop:
1504     \cs_gset_protected_nopar:Npx #1
1505     { \exp_not:o {#1} \exp_not:N \lua_now_x:n { #2 ( #3 ) } }
1506     \group_end:
1507 }
1508 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nnn { c }

```

(End definition for __ctex_ltj_save_alternate_shape:Nnn.)

clearalternatefont 清除和重置操作总是全局的。

resetalternatefont

```

1509 \keys_define:nn { ctex }
1510 {
1511     clearalternatefont .code:n =
1512     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1513     resetalternatefont .code:n =
1514     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1515     clearalternatefont .default:n = \l__ctex_ltj_family_tl ,
1516     resetalternatefont .default:n = \l__ctex_ltj_family_tl
1517 }
1518 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1519 {

```

```

1520 \group_begin:
1521 \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1522 {
1523 \cs_if_exist_use:cT { \l__ctex_ltj_alterate_cs:n { clear / #1 } }
1524 {
1525 \prop_gput:Nno \g__ctex_ltj_reset_alterate_prop
1526 {#1} { \l__ctex_ltj_base_family_tl }
1527 \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1528 \selectfont
1529 }
1530 }
1531 { \l__ctex_ltj_family_unknown_warning:n {#1} }
1532 \group_end:
1533 }
1534 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alterate_font:n #1
1535 {
1536 \group_begin:
1537 \prop_gpop:NnNT \g__ctex_ltj_reset_alterate_prop {#1} \CJK@family
1538 {
1539 \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1540 \use:c { \l__ctex_ltj_alterate_cs:n { reset / #1 } }
1541 \selectfont
1542 }
1543 \group_end:
1544 }
1545 \prop_new:N \g__ctex_ltj_reset_alterate_prop
1546 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

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

declarecharrange 预先声明字符范围。

```

1547 \keys_define:nn { ctex }
1548 {
1549 declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1550 declarecharrange .value_required:n = true
1551 }
1552 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1553 { \clist_map_inline:nn {#1} { \l__ctex_ltj_declare_char_range:nn ##1 } }
1554 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1555 \cs_new_protected_nopar:Npn \l__ctex_ltj_declare_char_range:nn #1#2
1556 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

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

```

\ctex_ltj_declare_char_range:nn #1 是名字,#2 是范围。
\g__ctex_ltj_char_range_prop
1557 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1558 {
1559 \tl_clear:N \l__ctex_ltj_char_range_tl
1560 \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1561 \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1562 \ctex_ltj_def_char_range_key:n {#1}
1563 \tl_clear:N \l__ctex_ltj_char_range_tl
1564 }
1565 \tl_new:N \l__ctex_ltj_char_range_tl
1566 \prop_new:N \g__ctex_ltj_char_range_prop

```

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

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

```

1567 \NewDocumentCommand \ctex_ltj_save_char_range:n
1568 { > { \SplitArgument { \c_one } { -> } } m }
1569 { \ctex_ltj_save_char_range:nn #1 }
1570 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2

```

```

1571 {
1572   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1573   { {
1574     \IfNoValueTF {#2}
1575     { \int_eval:n {#1} , \int_eval:n {#1} }
1576     {
1577       \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1578       \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1579     }
1580   } }
1581 }

```

(End definition for \ctex_ltj_save_char_range:n.)

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

```

1582 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1583 {
1584   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1585   {
1586     \keys_define:nn { ctex_ltj / fontspec }
1587     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1588   }
1589 }

```

(End definition for \ctex_ltj_def_char_range_key:n.)

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

```

1590 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1591 {
1592   \tl_if_blank:nTF {#2}
1593   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1594   {
1595     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1596     {
1597       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1598       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1599     }
1600   }
1601 }

```

(End definition for \ctex_ltj_char_range_key:nn.)

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

```

1602 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1603 { +o o u { \q_stop } }
1604 {
1605   \exp_args:NNf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1606   {
1607     \IfNoValueTF {#1} { {#3} }
1608     {
1609       \IfNoValueTF {#2}
1610       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] { #3 } } }
1611       { [ { #1 } ] { [ #2 ] } }
1612     }
1613   }
1614   \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1615 }

```

(End definition for __ctex_ltj_char_range_parse_feature:w.)

15.4.3.7 其它设置

`\ctex_ltj_update_xkanjiskip:` `\ltjsetparameter` 对 `xkanjiskip` 是即时赋值。`\zw` 与字体相关,因此需要每次 `\selectfont` 的时候更新一次 `xkanjiskip`。如果用户设置过 `xkanjiskip`, 就不更新。注意, 同 \TeX 的 `\baselineskip` 一样, 如果在一个段落内多次设置了 `kanjiskip` 或 `xkanjiskip`, 最后的设置会影响全段。

```

1616 \cs_new_protected_nopar:Npn \ctex_ltj_update_xkanjiskip:
1617 {
1618   \skip_if_eq:nnT
1619     { \ltjsetparameter { xkanjiskip } } { \l__ctex_ltj_xkanjiskip_skip }
1620   {
1621     \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }
1622     \ltjsetxkanjiskip \l__ctex_ltj_xkanjiskip_skip
1623   }
1624 }

```

注意, 此时还没有实际设置字体, 所以 `\zw` 是 0pt。

```

1625 \skip_new:N \l__ctex_ltj_xkanjiskip_skip
1626 \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }

```

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

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

```

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

1627 \AtBeginDocument
1628 {
1629   \ctex_appto_cmd:NnnTF \verbatim@font
1630     { \char_set_catcode_letter:n { 64 } }
1631     { \CTEX@verbatim@font@hook }
1632     { }
1633     { \ctex_patch_failure:N \verbatim@font }
1634 }
1635 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1636 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

1637 <@@=

```

`\@@italiccorr` \LaTeX 的倾斜校正也要重新定义。

```

1638 \cs_set_eq:NN \@@italiccorr \/

```

(End definition for `\@@italiccorr.`)

```

1639 <@@=ctex>

```

```

1640 </luatex>

```

15.4.4 `ctex-engine-uptex.def`

```

1641 <*uptex>

```


按 CJK 的命名习惯模拟部分命令,并设置默认字体。

```

1642 \tl_set:Nn \CJKrmdefault { rm }
1643 \tl_set:Nn \CJKsfdefault { sf }
1644 \tl_set:Nn \CJKttdefault { tt }
1645 \tl_set:Nn \CJKfamilydefault { \CJKrmdefault }
1646 \tl_put_left:Nn \rmfamily { \kanjifamily { \CJKrmdefault } }
1647 \tl_put_left:Nn \sffamily { \kanjifamily { \CJKsfdefault } }
1648 \tl_put_left:Nn \ttfamily { \kanjifamily { \CJKttdefault } }
1649 \NewDocumentCommand \CJKfamily { m }
1650 {
1651   \kanjifamily { #1 }
1652   \selectfont
1653 }
1654 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }

```

`\em` 取消 `up \TeX` 对 `\em` 使用 `\mcfamily`、`\gtfamily` 命令的重定义,恢复 `L \TeX 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。

```

1655 \ctex_patch_cmd:Nnn \em
1656 { \mcfamily \upshape \else \gtfamily \itshape }
1657 { \eminnershapes \else \itshape }

```

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

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

```

1658 \cs_new_protected_nopar:Npn \ctex_set_upfamily:nnn #1 #2 #3
1659 {
1660   \DeclareKanjiFamily{JY2}{#1}{#1}{}
1661   \DeclareKanjiFamily{JT2}{#1}{#1}{}
1662   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1663   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1664   \tl_if_empty:NF { #3 }
1665   {
1666     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1667     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1668   }
1669 }
1670 \@onlypreamble \ctex_set_upfamily:nnn

```

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

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

```

1671 \cs_new_protected_nopar:Npn \ctex_set_upmap:nnn #1 #2 #3
1672 {
1673   \ctex_set_zhmap:n
1674   {
1675     \special{ pdf:mapline~ #1-h~   UniGB-UTF16-H~ #2 }
1676     \special{ pdf:mapline~ #1-v~   UniGB-UTF16-V~ #2 }
1677     \tl_if_empty:NF { #3 }
1678     {
1679       \special{ pdf:mapline~ #1b-h~  UniGB-UTF16-H~ #3 }
1680       \special{ pdf:mapline~ #1b-v~  UniGB-UTF16-V~ #3 }
1681     }
1682   }
1683 }
1684 \@onlypreamble \ctex_set_upmap:nnn

```

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

`\ctex_set_upfonts:nnnnnn` 设置 $\text{up}\text{T}_{\text{E}}\text{X}$ 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。

```

1685 \cs_new_protected_nopar:Npn \ctex_set_upfonts:nnnnnn #1 #2 #3 #4 #5 #6
1686 {
1687   \ctex_set_upmap:nnn { upserif } { #1 } { #2 }
1688   \ctex_set_upmap:nnn { upserifit } { #3 } {}
1689   \ctex_set_upmap:nnn { upsans } { #4 } { #5 }
1690   \ctex_set_upmap:nnn { upmono } { #6 } {}
1691 }
1692 \@onlypreamble \ctex_set_upfonts:nnnnnn

(End definition for \ctex_set_upfonts:nnnnnn.)

```

`\@EverySelectfont@Init` `everyysel` 宏包 (2011/10/28) 未考虑 $\text{up}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 对 `\selectfont` 的修改, 因此这里需要对其给出正确的定义。因为不会使用 `CJK`, 这里忽略有关 `CJK` 的定义。

```

1693 \cs_set_nopar:Npn \@EverySelectfont@Init
1694 {
1695   \ctex_patch_cmd:Nnn \selectfont
1696   { \enc@update }
1697   {
1698     \@EverySelectfont@EveryHook
1699     \@EverySelectfont@AtNextHook
1700     \gdef \@EverySelectfont@AtNextHook { }
1701     \enc@update
1702   }
1703   \@ifpackageloaded { tracefmt }
1704   {
1705     \EverySelectfont
1706     {
1707       \ifnum \tracingfonts > \tw@
1708         \@font@info { Switching~ to~ \font@name }
1709         \fi
1710     }
1711   }
1712   { }
1713   \let \@EverySelectfont@Init \undefined
1714 }

(End definition for \@EverySelectfont@Init.)

```

1715 `</uptex>`

15.4.5 调整 `\CJKfamilydefault`

`\ctex_update_default_family:` 在导言区结束, 如果 `\CJKfamilydefault` 没有被更改, 则在此时根据西文字体的情况更新 `\CJKfamilydefault`。 `xeCJK` 已经有这个功能, 不需要再调整。

```

1716 <*pdfTeX|luatex>
1717 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1718 {
1719   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1720   {
1721     \group_begin:
1722     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1723     \tl_gset:Nx \CJKfamilydefault
1724     {
1725       \str_case:onF { \familydefault }
1726       {
1727         { \rmdefault } { \exp_not:N \CJKrmdefault }
1728         { \sfdefault } { \exp_not:N \CJKsfdefault }
1729         { \ttdefault } { \exp_not:N \CJKttdefault }
1730       }
1731       { \CJKfamilydefault }

```

```

1732     }
1733     \group_end:
1734 }

```

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

```

1735 <*\uotex>
1736     \ctex_ltj_ensure_default_family:
1737 </\uotex>
1738 }

```

(End definition for `\ctex_update_default_family:.`)

`\l_ctex_family_default_init_tl` 往 `\CJKfamilydefault` 中加入标志, 用于判断它是否被更改。

```

1739 \tl_new:N \l_ctex_family_default_init_tl
1740 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1741 \tl_set:Nx \l_ctex_family_default_init_tl
1742 {
1743     \exp_not:N \__ctex_family_default_wrap:n
1744     { \exp_not:o { \CJKfamilydefault } }
1745 }
1746 \tl_gset_eq:NN \CJKfamilydefault \l_ctex_family_default_init_tl
1747 </pdf\te|x|uotex>

```

(End definition for `\l_ctex_family_default_init_tl`.)

15.4.6 操作系统的判断

`\ctex_detected_platform:` 在 Lua_{TeX} 下直接用调用 `os.name` 来判断。

```

1748 <*\uotex>
1749 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1750 {
1751     \tl_gset:Nx \g__ctex_fontset_tl
1752     {
1753         \lua_now_x:n
1754         {
1755             if ~ os.name == 'windows' then ~
1756                 tex.sprint ( 'windows' )
1757             elseif ~ os.name == 'macosx' then ~
1758                 tex.sprint ( 'mac' )
1759             else ~
1760                 tex.sprint ( 'fandol' )
1761             end
1762         }
1763     }
1764 }
1765 </\uotex>

```

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

```

1766 <*\xetex|pdf\te|x|uptex>
1767 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1768 {
1769     \file_if_exist:nTF { /dev/null }
1770     {
1771         \file_if_exist:nTF { nul: }
1772         {
1773             \file_if_exist:nTF { \c__ctex_upper_case_file_str }
1774             { \ctex_if_macosx:TF { mac } { windows } }
1775             { \ctex_if_macosx:TF { mac } { fandol } }

```

```

1776         }
1777         { \ctex_if_macosx:TF { mac } { fandol } }
1778     }
1779     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1780 }
1781 \str_const:Nx \c__ctex_upper_case_file_str
1782 { \exp_args:No \str_upper_case:n { \g_file_current_name_tl } }

```

(End definition for \ctex_detected_platform:.)

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

```

1783 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1784 {
1785     \file_if_exist:nTF { \c__ctex_macosx_file_str }
1786     { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1787     { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1788 }
1789 \str_const:Nn \c__ctex_macosx_file_str { /Library/Fonts/Songti.ttc }
1790 </xetex|pdfTeX|uptex>

```

(End definition for \ctex_if_macosx:TF.)

15.4.7 hyperref 兼容性处理

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

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

```

1791 <*class|ctex>
1792 \@ifpackageloaded { hyperref }
1793 {
1794     \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1795     { \hypersetup {#1} }
1796 }
1797 {
1798     \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1799     { \PassOptionsToPackage {#1} { hyperref } }
1800 }
1801 </class|ctex>

```

(End definition for \ctex_hypersetup:n.)

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

```

1802 <*pdfTeX>
1803 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1804 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1805 {
1806     \ctex_hypersetup:n { CJKbookmarks = true }
1807     \sys_if_output_pdf:TF
1808     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1809     {
1810         \ctex_at_end_package:nn { hyperref }
1811         {
1812             \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1813             {

```

```

1814             \AtBeginShipoutFirst
1815             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1816         }
1817         { \RequirePackage { xCJK2uni } }
1818     }
1819 }
1820 }
1821 { \ctex_hypersetup:n { pdfencoding = unicode } }
1822 </pdfTeX>

```

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

```

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

```

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

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

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

```

1823 <*xetex|luatex>
1824 \ctex_hypersetup:n { pdfencoding = unicode }
1825 </xetex|luatex>

1826 <*uptex>
1827 \PassOptionsToPackage { dvipdfmx } { hyperref }
1828 \ctex_at_end_package:nn { hyperref }
1829 {
1830     \AtBeginShipoutFirst{ \special{ pdf:tounicode~ UTF8-UTF16 } }
1831 }
1832 </uptex>

1833 <*pdfTeX|xetex|luatex|uptex>

```

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

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

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

15.4.8 CJKfntef、xeCJKfntef 相关设置

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

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

```

1834 < *pdfTEX >
1835 \RequirePackage { CJKfntef }
1836 \normalem
1837 \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1838 { \tl_clear:c { CJK#1color } }
1839 < /pdfTEX >
1840 < *xeTEX >
1841 \RequirePackage { xeCJKfntef }
1842 \ifpackagelater { xeCJKfntef } { 2014/11/04 }
1843 {
1844   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1845   { \xeCJKsetup { #1 / format = { } } }
1846 }
1847 {
1848   \cs_new_protected_nopar:Npn \__ctex_clear_fntef_color:n #1
1849   { \tl_clear:c { CJK#1color } }
1850 }
1851 < /xeTEX >
1852 < *luaTEX >
1853 \msg_new:nnn { ctex } { fntef-not-available }
1854 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1855 \msg_warning:nn { ctex } { fntef-not-available }
1856 < /luaTEX >
1857 \clist_map_inline:nn
1858 { underdot , underline , underdblline , underwave , sout , xout }
1859 < *pdfTEX|xeTEX >
1860 {
1861   \__ctex_clear_fntef_color:n {#1}
1862   \cs_new_protected_nopar:cpx { CTEX#1 }
1863   {
1864     \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1865     { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1866     \exp_not:c { CJK#1 }
1867   }
1868 }
1869 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1870 {
1871   \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1872   { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1873   \CJKfilltwosides
1874 }
1875 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1876 < /pdfTEX|xeTEX >
1877 < *luaTEX|uptEX >
1878 { \cs_new_eq:cN { CTEX#1 } \use:n }
1879 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1880 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1881 < /luaTEX|uptEX >
1882 < *pdfTEX >
1883 \clist_map_inline:nn
1884 {
1885   underdotbasesep , underdotsep , underlinebasesep ,
1886   underlinesep , underdbllinesep , underdbllinebasesep ,
1887   underwavebasesep , underwavesep , southeight ,
1888   underdotcolor , underwavecolor , underlinecolor ,
1889   underdbllinecolor , soutcolor , xoutcolor
1890 }
1891 {

```

```

1892     \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1893     \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1894 }
1895 </pdfTeX>

```

15.4.9 \ccwd 的更新

\ctex_update_ccwd:

```

\ccwd
1896 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1897 <*pdfTeX|xetex>
1898 {
1899     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1900     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1901 }
1902 </pdfTeX|xetex>
1903 <*luatex>
1904 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1905 </luatex>
1906 <*uptex>
1907 { \skip_set:Nn \ccwd { 1zw } }
1908 </uptex>
1909 \dim_new:N \ccwd

```

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

\ctex_update_ccglue: 更新字间距。

```

1910 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1911 <*pdfTeX>
1912 {
1913     \cs_set_protected_nopar:Npn \CJKglue
1914     { \skip_horizontal:N \l__ctex_ccglue_skip }
1915 }
1916 </pdfTeX>
1917 <*xetex>
1918 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1919 </xetex>
1920 <*luatex>
1921 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1922 </luatex>
1923 <*uptex>
1924 { \kanjiskip \l__ctex_ccglue_skip }
1925 </uptex>
1926 \skip_new:N \l__ctex_ccglue_skip

```

(End definition for \ctex_update_ccglue:.)

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

\ctex_if_ccglue_touched: TF

```

1927 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1928 {
1929 <*pdfTeX|xetex>
1930     \if_meaning:w \CJKglue \__ctex_ccglue:
1931     \prg_return_false: \else: \prg_return_true: \fi:
1932 </pdfTeX|xetex>
1933 <*luatex>
1934     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1935     { \prg_return_false: } { \prg_return_true: }
1936 </luatex>
1937 <*uptex>
1938     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \kanjiskip }
1939     { \prg_return_false: } { \prg_return_true: }
1940 </uptex>
1941 }

```

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

```
1942 <*pdf|xetex>
1943 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1944 </pdf|xetex>
```

(End definition for \ctex_if_ccglue_touched:TF.)

`\ctex_update_em_unit:` 将当前汉字的宽度保存到 `\ccwd` 中备用。不采用 `1 em`, 因为这时的 `1 em` 实际上来自西文字体的信息, 未必等于汉字的宽度, 这似乎在传统的 `.tfm` 字体上表现更明显。在 `pdfTeX` 和 `XgTeX` 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 `XgTeX` 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`。

```
1945 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1946 <pdf|xetex> { \dim_set:Nn \ccwd { \f@size \p@ } }
1947 <luatex> { \dim_set:Nn \ccwd { \zw } }
1948 <uptex> { \dim_set:Nn \ccwd { 1zw } }
```

(End definition for \ctex_update_em_unit:.)

15.4.10 其它

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

```
1949 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1950 {
1951   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1952   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1953 }
1954 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1955 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1956 \ctex_preto_cmd:NnnTF \selectfont { \ExplSyntaxOff }
1957 { \CTEX@selectfont@hook }
1958 {
1959   \tl_put_left:Nn \@EverySelectfont@Init
1960   { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1961 }
1962 { \ctex_patch_failure:N \selectfont }
```

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

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

```
1963 <*pdf|xetex>
1964 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1965 </pdf|xetex>
1966 <*xetex|luatex|uptex>
1967 \EverySelectfont { \CTEX@selectfont@hook }
1968 </xetex|luatex|uptex>
```

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。

```
1969 <*luatex>
1970 \ctex_add_to_selectfont:n
1971 {
1972   \ctex_ltj_select_font:
```



```

1973 \ctex_ljtj_select_alternate_font:
1974 \ctex_ljtj_update_xkanjiskip:
1975 }
1976 \tl_set:Nn \CJK@family { song } \selectfont
1977 \tl_clear:N \CJK@family
1978 </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。

```

1979 \keys_define:nn { ctex }
1980 {
1981 <*pdf|xe|lua|up>
1982   space .choice: ,
1983   space / true .code:n =
1984 <pdf|xe|lua|up> { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
1985 <xe|lua|up> { \xeCJKsetup { CJKspace = true } } ,
1986   space / auto .code:n =
1987 <pdf|xe|lua|up> { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
1988 <xe|lua|up> { \xeCJKsetup { CJKspace = false } } ,
1989   space / false .code:n =
1990 <pdf|xe|lua|up> { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
1991 <xe|lua|up> { \xeCJKsetup { CJKspace = false } } ,
1992   space .default:n = { true } ,
1993   space .initial:n = { auto }
1994 </pdf|xe|lua|up>
1995 <*lua|up>
1996   space .code:n =
1997   { \msg_warning:nn { ctex } { invalid-option } }
1998 </lua|up>
1999 }

```

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

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-jan 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。

```

2000 \keys_define:nn { ctex }
2001 {
2002   punct .code:n =
2003   {
2004     \tl_set:Nx \l__ctex_punct_tl { #1 }
2005 <pdf|xe|lua|up> \punctstyle { \l__ctex_punct_tl }
2006 <xe|lua|up> \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2007 <lua|up> \ctex_mono_jfm:o { \l__ctex_punct_tl }
2008 <up> \msg_warning:nn { ctex } { invalid-option }
2009   } ,
2010   punct .default:n = { quanjiao } ,
2011 }

```

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

X_{La}TeX、Lua_{La}TeX 和 up_{La}TeX 总是使用 UTF8 编码。

```

2012 <*xe|lua|up>
2013 \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
2014 <xe|lua|up>
2015 </pdf|xe|lua|up>
2016 <*class|ctex>

```

15.4.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

2017 \ctex_file_input:n { \c__ctex_engine_file_str }

```

15.5 字距与缩进

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

```

2018 \keys_define:nn { ctex }
2019 {
2020   autoindent .choice: ,
2021   autoindent .default:n = { true } ,
2022   autoindent / true .code:n =
2023   {
2024     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2025     \ctex_select_size:
2026   } ,
2027   autoindent / false .code:n =
2028   { \tl_clear:N \l__ctex_autoindent_tl } ,
2029   autoindent / unknown .code:n =
2030   {
2031     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2032     \ctex_select_size:
2033   }
2034 }

```

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

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2035 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2036 \cs_new_protected_nopar:Npn \ctex_select_size:
2037 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

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

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

```

2038 \cs_new_protected_nopar:Npn \ctex_update_size:
2039 {
2040   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2041   {
2042     \ctex_update_stretch:
2043     \ctex_update_parindent:
2044   }
2045   { \ctex_update_ziju: }
2046 }
2047 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2048 \tl_new:N \l__ctex_ziju_tl
2049 \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。

```

2050 \ctex_add_to_selectfont:n
2051 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

(End definition for `\ctex_update_size:`.)

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

```

2052 \keys_define:nn { ctex }
2053 {
2054   linestretch .code:n =
2055   {
2056     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2057     \ctex_select_size:
2058   } ,

```

```

2059     linestretch .value_required:n = true
2060   }
2061   \tl_new:N \l__ctex_line_stretch_tl
2062   \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

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

`\ctex_update_stretch:` 首先计算一行上汉字的字数, `\CJKglue` 相当于将 `\linewidth` 与汉字总宽度之差均匀地填充到汉字之间。 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。

```

2063   \cs_new_protected_nopar:Npn \ctex_update_stretch:
2064   {
2065     \ctex_update_em_unit:
2066     \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2067     \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2068     {
2069       \ctex_if_ccglue_touched:TF
2070       { \ctex_update_ccwd: }
2071       {
2072         \dim_set:Nn \l__ctex_tmp_dim
2073         { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
2074         \skip_set:Nn \l__ctex_ccglue_skip
2075         { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2076         \ctex_update_ccglue:
2077       }
2078     }
2079     {
2080       \int_set:Nn \l__ctex_tmp_int
2081       { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2082       \tex_divide:D \l__ctex_tmp_int \ccwd
2083       \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
2084       {
2085         \skip_set:Nn \l__ctex_ccglue_skip
2086         {
2087           \c_zero_dim plus \dim_eval:n
2088           {
2089             ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2090             \l__ctex_tmp_int
2091           }
2092         }
2093       }
2094       { \skip_zero:N \l__ctex_ccglue_skip }
2095       \ctex_update_ccglue:
2096     }
2097   }

```

(End definition for `\ctex_update_stretch:`.)

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

```

2098   \cs_new_protected_nopar:Npn \ctex_update_parindent:
2099   {
2100     \tl_if_empty:NF \l__ctex_autoindent_tl
2101     {
2102       \dim_compare:nNnF \parindent = \c_zero_dim
2103       { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2104     }
2105   }

```

(End definition for `\ctex_update_parindent:`.)

`\ziju` 若参数为 0, 则恢复正常间距。

```

2106 \NewDocumentCommand \ziju { m }
2107   { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2108 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
2109   {
2110     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2111     \ctex_select_size:
2112   }

```

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

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

```

2113 \cs_new_protected_nopar:Npn \ctex_update_ziju:
2114   {
2115     \ctex_update_em_unit:
2116     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2117     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2118     \dim_compare:nNnTF \ccwd > \c_zero_dim

```

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

```

2119     {
2120       \dim_set:Nn \l__ctex_tmp_dim
2121         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2122       \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2123       \tex_divide:D \l__ctex_tmp_int \ccwd
2124       \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。

```

2125       \dim_compare:nNnF \parindent = \c_zero_dim
2126       {
2127         \int_compare:nNnF \l__ctex_tmp_int < \c_three
2128         { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
2129       }
2130       \skip_set:Nn \l__ctex_ccglue_skip
2131       {
2132         \l__ctex_ziju_dim
2133         plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2134         minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2135         { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2136       }
2137     }
2138     { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2139     \ctex_update_ccglue:

```

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

```

2140     \ctex_update_parindent:
2141   }
2142 \dim_new:N \l__ctex_ziju_dim

```

(End definition for `\ctex_update_ziju:`)

`\CTEXindent` 过时命令。
`\CTEXnoindent`

```

2143 \NewDocumentCommand \CTEXindent { }
2144   {
2145     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }

```

```

2146     { \parindent is~ set~ to~ 2\ccwd. }
2147     \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2148   }
2149   \NewDocumentCommand \CTEXnoindent { }
2150   {
2151     \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2152     { \parindent is~ set~ to~ Opt. }
2153     \dim_zero:N \parindent
2154   }

```

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

15.6 中文数字与日期

```

2155 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2156 \RequirePackage { zhnumber }

```

\chinese

```

2157 \cs_new_eq:NN \chinese \zhnum
2158 \cs_new_eq:NN \Chinese \chinese
2159 \cs_new_eq:NN \CTEXcounter \use_none:n

```

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

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

```

2160 \ctex_at_end_package:nn { enumitem }
2161 {
2162   \cs_if_free:NF \AddEnumerateCounter
2163   {
2164     \AddEnumerateCounter * { \zhnum } { \zhnum_int:n } { 1 }
2165     \AddEnumerateCounter * { \chinese } { \zhnum_int:n } { 1 }
2166   }
2167 }

```

\CTEXnumber

\CTEXdigits

```

2168 \NewDocumentCommand \CTEXnumber { m m }
2169 { \protected@edef #1 { \zhnumber {#2} } }
2170 \NewDocumentCommand \CTEXdigits { m m }
2171 { \protected@edef #1 { \zhdigits {#2} } }

```

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

today

```

2172 \cs_set_eq:NN \CTEX@todayold \today
2173 \keys_define:nn { ctex }
2174 {
2175   today .choice: ,
2176   today / old .code:n =
2177     { \cs_set_eq:NN \today \CTEX@todayold } ,
2178   today / small .code:n =
2179     {
2180       \cs_set_eq:NN \today \zhtoday
2181       \zhnumsetup { time = Arabic }
2182     } ,
2183   today / big .code:n =
2184     {
2185       \cs_set_eq:NN \today \zhtoday
2186       \zhnumsetup { time = Chinese }
2187     } ,
2188   today / unknown .code:n =
2189     { \msg_error:nnx { ctex } { today-undef } {#1} }
2190 }

```

```

2191 \msg_new:nnnn { ctex } { today-undef }
2192 { Today~format~`#1'~is~undefined. }
2193 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

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

15.7 其它中文标题定义

```

2194 \keys_define:nn { ctex }
2195 {
2196   contentsname .tl_set:N = \contentsname ,
2197   listfigurename .tl_set:N = \listfigurename ,
2198   listtablename .tl_set:N = \listtablename ,
2199   figurename .tl_set:N = \figurename ,
2200   tablename .tl_set:N = \tablename ,
2201   abstractname .tl_set:N = \abstractname ,
2202   indexname .tl_set:N = \indexname ,
2203   appendixname .tl_set:N = \appendixname ,
2204   proofname .tl_set:N = \proofname ,
2205   <article> bibname .tl_set:N = \refname
2206   <book|report> bibname .tl_set:N = \bibname
2207   <*beamer>
2208     algorithmname .tl_set:N = \algorithmname ,
2209     bibname .tl_set:N = \bibname ,
2210     refname .tl_set:N = \refname ,
2211     continuation .tl_set:N = \insertcontinuationtext
2212   </beamer>
2213 }
2214 <*ctex>
2215 \msg_new:nnn { ctex } { ctexbibname }
2216 {
2217   Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2218   The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2219 }
2220 \tl_if_exist:NTF \insertcontinuationtext
2221 {
2222   \keys_define:nn { ctex }
2223   {
2224     algorithmname .tl_set:N = \algorithmname ,
2225     bibname .tl_set:N = \bibname ,
2226     refname .tl_set:N = \refname ,
2227     continuation .tl_set:N = \insertcontinuationtext
2228   }
2229 }
2230 {
2231   \tl_if_exist:NTF \bibname
2232   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2233   {
2234     \tl_if_exist:NTF \refname
2235     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2236     {
2237       \msg_warning:nn { ctex } { ctexbibname }
2238       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2239     }
2240   }
2241 }
2242 </ctex>
2243 </class|ctex>

```

15.8 中文文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```

2244 <*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。

```

2245 <*article|book|report>
2246 \seq_new:N \c__ctex_section_headings_seq
2247 \seq_gset_from_clist:Nn \c__ctex_section_headings_seq
2248   { section , subsection , subsubsection , paragraph , subparagraph }
2249 </article|book|report>

(End definition for \c__ctex_section_headings_seq.)

```

`\c__ctex_headings_seq`

```

2250 \seq_new:N \c__ctex_headings_seq
2251 <*article|book|report>
2252 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
2253 <book|report>\seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2254 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2255 </article|book|report>
2256 <*beamer>
2257 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2258   { part , section , subsection }
2259 </beamer>

(End definition for \c__ctex_headings_seq.)

```

`__ctex_initial_heading:n`

```

2260 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2261 {
2262   \tl_new:c { CTEX@pre#1 }
2263   \tl_new:c { CTEX@post#1 }
2264   \tl_const:cx { CTEX@the#1 }
2265   {
2266     \exp_not:c { CTEX@pre#1 }
2267     \exp_not:c { CTEX@the#1 }
2268     \exp_not:c { CTEX@post#1 }
2269   }
2270   \tl_const:cx { CTEX@#1name }
2271   {
2272     \group_begin:
2273     \exp_not:c { CTEX@#1@nameformat }
2274     {
2275       \exp_not:c { CTEX@pre#1 }
2276       \exp_not:N \tl_if_empty:NTF
2277       \exp_not:c { CTEX@#1@numberformat }
2278       { \exp_not:c { CTEX@the#1 } }
2279       {
2280         \group_begin:
2281         \exp_not:c { CTEX@#1@numberformat }
2282         \exp_not:c { CTEX@the#1 }
2283         \group_end:
2284       }
2285       \exp_not:c { CTEX@post#1 }
2286     }
2287   \group_end:
2288 }
2289 }

```

(End definition for `__ctex_initial_heading:n`.)

`__ctex_def_heading_keys:n`

```

2290 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2291 {

```

```

2292 \tl_put_right:Nx \l__ctex_tmp_tl
2293 {
2294   #1 .meta:nn = { ctex / #1 } { #####1 } ,
2295   #1 / name .code:n =
2296     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2297   #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2298   #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2299   #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2300   #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2301   #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2302   #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2303   #1 / aftertitle .tl_set:N = \exp_not:c { CTEX@#1@aftertitle } ,
2304   #1 / before skip .tl_set:N = \exp_not:c { CTEX@#1@before skip } ,
2305   #1 / after skip .tl_set:N = \exp_not:c { CTEX@#1@after skip } ,
2306   #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2307   #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2308   < *article|book|report >
2309   #1 / after indent .bool_set:N = \exp_not:c { CTEX@#1@after indent } ,
2310   #1 / run in .bool_set:N = \exp_not:c { CTEX@#1@run in } ,
2311   < /article|book|report >
2312   #1 / format+ .code:n =
2313     { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2314   #1 / nameformat+ .code:n =
2315     { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2316   #1 / numberformat+ .code:n =
2317     { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2318   #1 / aftername+ .code:n =
2319     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2320   #1 / titleformat+ .code:n =
2321     { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2322   #1 / aftertitle+ .code:n =
2323     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftertitle } { #####1 } } ,
2324   #1 / numbering .initial:n = true ,
2325   #1 / before skip .initial:n = \c_zero_skip ,
2326   #1 / after skip .initial:n = \c_zero_skip ,
2327   #1 / indent .initial:n = \c_zero_dim ,
2328   #1 / before skip .value_required:n = true ,
2329   #1 / after skip .value_required:n = true ,
2330   #1 / indent .value_required:n = true ,
2331 }
2332 }

```

(End definition for __ctex_def_heading_keys:n.)

`ctex_assign_heading_name:nn` `name` 的值是一个至多两个元素的逗号分隔列表。由于 \LaTeX 3 的 `clist` 总是会自动忽略空元素，所以设置 `name={,章}` 后，第一个元素将会是“章”，必须用空的分组保护空元素：`name={ {},章}`，这在使用中有些许不便。我们可以改用 `seq` 或者手写函数解析参数来加以改进。为实现的简单起见，这里用了 `xparse` 的 `\SplitArgument`，它带有参数的长度检查。

```

2333 \NewDocumentCommand \ctex_assign_heading_name:nn
2334 { m > { \SplitArgument { \c_one } { , } } +m }
2335 { \__ctex_assign_heading_name:nnn {#1} #2 }
2336 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2337 {
2338   \tl_set:cn { CTEX@pre#1 } {#2}
2339   \IfNoValueTF {#3}
2340   { \tl_clear:c { CTEX@post#1 } }
2341   { \tl_set:cn { CTEX@post#1 } {#3} }
2342 }

```

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

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

```

2343 < *book|report >
2344 \tl_set:Nn \l__ctex_tmp_tl

```



```

2345 {
2346   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2347   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2348 }
2349 </book|report>

```

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

定义标题键值选项。

```

2350 \seq_map_inline:Nn \c__ctex_headings_seq
2351 {
2352   \__ctex_initial_heading:n {#1}
2353   \__ctex_def_heading_keys:n {#1}
2354 }
2355 \use:x { \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } } }
2356 \tl_clear:N \l__ctex_tmp_tl
2357 <*article|book|report>

```

15.8.2 标准标题命令的修改

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

```

2358 \cs_new_protected_nopar:Npn \CTEX@makeheadinganchor { }
2359 \ctex_at_end_package:nn { hyperref }
2360 { \cs_gset_protected_nopar:Npn \CTEX@makeheadinganchor { \phantomsection } }

```

(End definition for \CTEX@makeheadinganchor.)

15.8.2.1 part 的标题

```

2361 <@@=@>

```

\part

```

2362 <*article>
2363 \renewcommand\part{%
2364   \if@noskipsec \leavevmode \fi
2365   \par
2366   % \addvspace{4ex}%
2367   \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
2368   \ifodd \CTEX@part@afterindent
2369     \@afterindenttrue
2370   \else
2371     \@afterindentfalse
2372   \fi
2373   \secdef\@part\@spart}
2374 </article>
2375 <*book|report>
2376 \renewcommand\part{%
2377   \if@openright
2378     \cleardoublepage
2379   \else
2380     \clearpage
2381   \fi
2382   % \thispagestyle{plain}%
2383   \thispagestyle{\CTEX@part@pagestyle}%
2384   \if@twocolumn
2385     \onecolumn
2386     \@tempwatrue
2387   \else
2388     \@tempwafalse
2389   \fi
2390   % \null\vfil
2391   \null\vskip \glueexpr \CTEX@part@beforeskip \relax

```

```

2392 \secdef\@part\@spart}
2393 </book|report>

```

(End definition for \part.)

\@part

```

2394 <*article>
2395 \def\@part[#1]#2{%
2396   \ifnum \c@secnumdepth >\m@ne
2397     \ifodd \CTEX@part@numbering
2398       \refstepcounter{part}%
2399       % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2400       \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2401     \else
2402       \CTEX@makeheadinganchor
2403       \addcontentsline{toc}{part}{#1}%
2404     \fi
2405   \else
2406     \addcontentsline{toc}{part}{#1}%
2407   \fi
2408   {\interlinepenalty \@M
2409   % \normalfont \parindent \z@ \raggedright
2410   \normalfont \parindent \dimexpr \CTEX@part@indent \relax \CTEX@part@format
2411   \ifnum \c@secnumdepth >\m@ne \ifodd \CTEX@part@numbering
2412     % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2413     \CTEX@partname \CTEX@part@aftername
2414   \fi \fi
2415   % \huge\bfseries #2%
2416   \CTEX@part@titleformat{#2}%
2417   \markboth{}{}%
2418   \CTEX@part@aftertitle}%
2419   \nobreak
2420   % \vskip 3ex
2421   \vskip \glueexpr \CTEX@part@afterskip \relax
2422   \@afterheading}
2423 </article>
2424 <*book|report>
2425 \def\@part[#1]#2{%
2426   \ifnum \c@secnumdepth >-2\relax
2427     \ifodd \CTEX@part@numbering
2428       \refstepcounter{part}%
2429       % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2430       \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2431     \else
2432       \CTEX@makeheadinganchor
2433       \addcontentsline{toc}{part}{#1}%
2434     \fi
2435   \else
2436     \addcontentsline{toc}{part}{#1}%
2437   \fi
2438   \markboth{}{}%
2439   {\interlinepenalty \@M
2440   % \normalfont \centering
2441   \normalfont \CTEX@part@format
2442   \ifnum \c@secnumdepth >-2\relax \ifodd \CTEX@part@numbering
2443     % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2444     \CTEX@partname \CTEX@part@aftername
2445   \fi \fi
2446   % \Huge\bfseries #2\par}%
2447   \CTEX@part@titleformat{#2}%
2448   \CTEX@part@aftertitle}%
2449   \@endpart}
2450 </book|report>

```

(End definition for \@part.)

\@spart

```

2451 <*article>
2452 \def\@spart#1{%
2453   {\interlinepenalty \@M
2454    % \normalfont \parindent \z@ \raggedright
2455     \normalfont \parindent \dimexpr \CTEX@part@indent \relax
2456     \CTEX@part@format
2457    % \huge \bfseries #1\par}%
2458     \CTEX@part@titleformat{#1}%
2459     \CTEX@part@aftertitle}%
2460    \nobreak
2461    % \vskip 3ex
2462     \vskip \glueexpr \CTEX@part@afterskip \relax
2463     \@afterheading}
2464 </article>
2465 <*book|report>
2466 \def\@spart#1{%
2467   {\interlinepenalty \@M
2468    % \normalfont \centering
2469     \normalfont \CTEX@part@format
2470    % \Huge \bfseries #1\par}%
2471     \CTEX@part@titleformat{#1}%
2472     \CTEX@part@aftertitle}%
2473   \@endpart}
2474 </book|report>

```

(End definition for \@spart.)

\@endpart

```

2475 <*book|report>
2476 \def\@endpart{%
2477   % \vfil
2478     \vskip \glueexpr \CTEX@part@afterskip \relax
2479     \newpage
2480     \if@twoside
2481       \if@openright
2482         \null
2483         \thispagestyle{empty}%
2484         \newpage
2485       \fi
2486     \fi
2487     \if@tempswa
2488       \twocolumn
2489     \fi}
2490 </book|report>

```

(End definition for \@endpart.)

15.8.2.2 chapter 的标题

2491 <*book|report>

\chapter

```

2492 \renewcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
2493 % \thispagestyle{plain}%
2494     \thispagestyle{\CTEX@chapter@pagestyle}%
2495     \global\@topnum\z@
2496 % \@afterindentfalse
2497     \ifodd \CTEX@chapter@afterindent
2498       \@afterindenttrue
2499     \else
2500       \@afterindentfalse
2501     \fi
2502     \secdef\@chapter\@schapter}

```

(End definition for \chapter.)

\@chapter

```

2503 \def\@chapter[#1]#2{%
2504   \ifnum \c@secnumdepth >\m@ne
2505   <book>   \if@mainmatter
2506             \ifodd \CTEX@chapter@numbering
2507               \refstepcounter{chapter}%
2508               \typeout{\@chapapp\space\thechapter.}%
2509               \typeout{\CTEXthechapter}%
2510               \addcontentsline{toc}{chapter}
2511               {\protect\numberline{\thechapter}#1}%
2512               {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2513             \else
2514               \CTEX@makeheadinganchor
2515               \addcontentsline{toc}{chapter}{#1}%
2516             \fi
2517   <book>   \else
2518   <book>     \addcontentsline{toc}{chapter}{#1}%
2519   <book>     \fi
2520   \else
2521     \addcontentsline{toc}{chapter}{#1}%
2522   \fi
2523   \chaptermark{#1}%
2524   \addtocontents{lof}{\protect\advspace{10\p@}}%
2525   \addtocontents{lot}{\protect\advspace{10\p@}}%
2526   \if@twocolumn
2527     \topnewpage[\@makechapterhead{#2}]%
2528   \else
2529     \makechapterhead{#2}%
2530   \@afterheading
2531   \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2532 \def\@makechapterhead#1{%
2533   % \vspace*{50\p@}%
2534   \vspace*{\glueexpr\CTEX@chapter@beforeskip\relax}%
2535   % {\normalfont \parindent \z@ \raggedright
2536   {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2537     \CTEX@chapter@format
2538     \ifnum \c@secnumdepth >\m@ne
2539     <book>   \if@mainmatter
2540               \ifodd \CTEX@chapter@numbering
2541               % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2542               \CTEX@chaptername \CTEX@chapter@aftername
2543             \fi
2544     <book>   \fi
2545     \fi
2546     \interlinepenalty\@M
2547     % \Huge \bfseries #1\par\nobreak
2548     \CTEX@chapter@titleformat{#1}%
2549     \CTEX@chapter@aftertitle
2550     \nobreak
2551     % \vskip 40\p@
2552     \vskip \glueexpr \CTEX@chapter@afterskip \relax
2553   }}

```

(End definition for \@makechapterhead.)

\@makeschapterhead

```

2554 \def\@makeschapterhead#1{%
2555   % \vspace*{50\p@}%

```

```

2556 \vspace*{\glueexpr\CTEX@chapter@beforeskip\relax}%
2557 % {\normalfont \parindent \z@ \raggedright
2558 {\normalfont \parindent \dimexpr \CTEX@chapter@indent \relax
2559 \CTEX@chapter@format
2560 \interlinepenalty\@M
2561 % \Huge \bfseries #1\par\nobreak
2562 \CTEX@chapter@titleformat{#1}
2563 \CTEX@chapter@aftertitle
2564 \nobreak
2565 % \vskip 40\p@
2566 \vskip \glueexpr \CTEX@chapter@afterskip \relax
2567 }}

```

(End definition for \@makeschapterhead.)

2568 </book|report>

15.8.2.3 section 类的标题

\@startsection

```

2569 \def\@startsection#1#2#3#4#5#6{%
2570 \if@noskipsec \leavevmode \fi
2571 \par
2572 % \@tempskipa #4\relax
2573 % \@afterindenttrue
2574 % \ifdim \@tempskipa <\z@
2575 % \@tempskipa -\@tempskipa \@afterindentfalse
2576 % \fi
2577 \CTEX@update@sectionformat@n{#1}%
2578 \ifodd \CTEX@afterindent
2579 \@afterindenttrue
2580 \else
2581 \@afterindentfalse
2582 \fi
2583 \if@nobreak
2584 \everypar{}%
2585 \else
2586 % \addpenalty\@secpenalty\addvspace\@tempskipa
2587 \addpenalty\@secpenalty\addvspace{\glueexpr #4\relax}%
2588 \fi
2589 \@ifstar
2590 {\@ssect{#3}{#4}{#5}{#6}}%
2591 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

(End definition for \@startsection.)

\@secntformat

```

2592 \def\@secntformat#1{%
2593 % \csname the#1\endcsname\quad}%
2594 \csname CTEX@#1name\endcsname \csname CTEX@#1aftername\endcsname}

```

(End definition for \@secntformat.)

\@sect

```

2595 \def\@sect#1#2#3#4#5#6[#7]#8{%
2596 \ifnum #2>\c@secnumdepth
2597 \let\@svsec\@empty
2598 \else
2599 \ifodd \csname CTEX@#1numbering\endcsname
2600 \refstepcounter{#1}%
2601 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2602 \else
2603 \CTEX@makeheadinganchor

```

```

2604     \let\@svsec\@empty
2605     \fi
2606     \fi
2607     % \@tempskipa #5\relax
2608     % \ifdim \@tempskipa>\z@
2609     \unless \ifodd \CTEX@runin
2610     \beginngroup
2611     #6{%
2612         \@hangfrom{\hskip\glueexpr #3\relax\@svsec}%
2613     %     \interlinepenalty \@M #8\@par}%
2614         \interlinepenalty \@M
2615         \csname CTEX@#1@titleformat\endcsname{#8}%
2616         \csname CTEX@#1@aftertitle\endcsname}%
2617     \endgroup
2618     \csname #1mark\endcsname{#7}%
2619     \addcontentsline{toc}{#1}{%
2620         \ifnum #2>\c@secnumdepth \else
2621             \ifodd \csname CTEX@#1@numbering\endcsname
2622 %             \protect\numberline{\csname the#1\endcsname}%
2623             \protect\numberline{\csname CTEXthe#1\endcsname}%
2624         \fi
2625         \fi
2626     #7}%
2627 \else
2628     \def\@svsechd{%
2629         #6{\hskip\glueexpr #3\relax
2630 %         \@svsec #8}%
2631         \@svsec
2632         \csname CTEX@#1@titleformat\endcsname{#8}%
2633         \csname CTEX@#1@aftertitle\endcsname}%
2634     \csname #1mark\endcsname{#7}%
2635     \addcontentsline{toc}{#1}{%
2636         \ifnum #2>\c@secnumdepth \else
2637             \ifodd \csname CTEX@#1@numbering\endcsname
2638 %             \protect\numberline{\csname the#1\endcsname}%
2639             \protect\numberline{\csname CTEXthe#1\endcsname}%
2640         \fi
2641         \fi
2642     #7}}%
2643 \fi
2644 \@xsect{#5}}

```

(End definition for \@sect.)

\@ssect

```

2645 \def\@ssect#1#2#3#4#5{%
2646 % \@tempskipa #3\relax
2647 % \ifdim \@tempskipa>\z@
2648 \unless \ifodd \CTEX@runin
2649 \beginngroup
2650 #4{%
2651     \@hangfrom{\hskip\glueexpr #1\relax}%
2652 %     \interlinepenalty \@M #5\@par}%
2653         \interlinepenalty \@M
2654         \CTEX@titleformat@n{#5}%
2655         \CTEX@aftertitle}%
2656 \endgroup
2657 \else
2658 % \def\@svsechd{#4{\hskip\glueexpr #1\relax #5}}%
2659 \def\@svsechd{#4{\hskip\glueexpr #1\relax
2660     \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2661 \fi
2662 \@xsect{#3}}

```

(End definition for \@ssect.)

\@xsect

```

2663 \def\@xsect#1{%
2664   % \@tempskipa #1\relax
2665   % \ifdim \@tempskipa>\z@
2666   \unless \ifodd \CTEX@runin
2667     \par \nobreak
2668   %   \vskip \@tempskipa
2669   \vskip\glueexpr #1\relax
2670   \@afterheading
2671 \else
2672   \@nobreakfalse
2673   \global\@noskipsectrue
2674   \everypar{%
2675     \if@noskipsec
2676       \global\@noskipsecfalse
2677       {\setbox\z@\lastbox}%
2678       \clubpenalty\@M
2679       \begingroup \@svsechd \endgroup
2680       \unskip
2681     %   \@tempskipa #1\relax
2682     %   \hskip -\@tempskipa
2683     \hskip\glueexpr #1\relax
2684   \else
2685     \clubpenalty \@clubpenalty
2686     \everypar{}%
2687   \fi}%
2688 \fi
2689 \ignorespaces}

```

(End definition for \@xsect.)

2690 <@@=ctex>

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

```

2691 \cs_new_protected_nopar:Npn \CTEX@update@sectionformat@n #1
2692 {
2693   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2694   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2695   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2696   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2697 }
2698 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2699 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2700 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2701 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

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

```

2702 \int_zero:N \l__ctex_tmp_int
2703 \seq_map_inline:Nn \c__ctex_section_headings_seq
2704 {
2705   \int_incr:N \l__ctex_tmp_int
2706   \cs_gset_protected_nopar:cpx {#1}
2707   {
2708     \exp_not:N \@startsection {#1}
2709     { \int_use:N \l__ctex_tmp_int }
2710     { \exp_not:c { CTEX@#1@indent } }
2711     { \exp_not:c { CTEX@#1@beforeskip } }
2712     { \exp_not:c { CTEX@#1@afterskip } }
2713     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2714   }
2715 }

```

15.8.2.4 附录标题

```

appendix/name
appendix/number
appendix/numbering
2716 \keys_define:nn { ctex }
2717 {
2718   appendix .meta:nn = { ctex / appendix } {#1} ,
2719   appendix / name .code:n =
2720     { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2721   appendix / number .tl_set:N = \CTEX@appendix@number ,
2722   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
2723   appendix / numbering .initial:n = true
2724 }
2725 \tl_new:N \CTEX@preappendix
2726 \tl_new:N \CTEX@postappendix

```

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

\appendix

```

2727 \cs_new_eq:NN \CTEX@save@appendix \appendix
2728 \cs_gset_protected_nopar:Npn \appendix
2729 {
2730   \CTEX@save@appendix
2731   <*article>
2732   \gdef \CTEX@presection { \CTEX@preappendix }
2733   \gdef \CTEX@thesection { \CTEX@appendix@number }
2734   \gdef \CTEX@postsection { \CTEX@postappendix }
2735   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
2736 </article>
2737 <*book|report>
2738   \gdef \CTEX@prechapter { \CTEX@preappendix }
2739   \gdef \CTEX@thechapter { \CTEX@appendix@number }
2740   \gdef \CTEX@postchapter { \CTEX@postappendix }
2741   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
2742 </book|report>
2743 }

```

(End definition for \appendix.)

15.8.2.5 兼容 titlesec 宏包

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

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

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

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

我们这里的 <shape> 为 hang 或者 runin。 \titlespacing 的设置保存在 \ttl@{<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 CT_EX 的 runin 和 afterindent 选项调整 \ttl@{<shape>} 和 <afterindent>。注意, 由 \ttl@extract 得的 <before> 和 <after> 的值总是非负的, 而 CT_EX 的 before skip 和 after skip 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 \ttl@{<section>}。


```

2744 \cs_new_protected_nopar:Npn \ctex_titlesec_hook:
2745 {
2746   \@ifpackagewith { titlesec } { explicit }
2747   {
2748     \cs_set_eq:NN \__ctex_titlesec_format:Nn
2749     \__ctex_titlesec_format_explicit:Nn
2750   }
2751   { }
2752   \clist_map_inline:nn
2753   { indentafter , noindentafter , indentfirst , nonindentfirst }
2754   {
2755     \@ifpackagewith { titlesec } { ##1 }
2756     {
2757       \clist_map_break:n
2758       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
2759     }
2760     { }
2761   }
2762   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
2763 }
2764 \cs_new_protected_nopar:Npn \__ctex_titlesec_hook:n #1
2765 {
2766   \__ctex_titlesec_format:n {#1}
2767   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
2768 }
2769 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:n #1
2770 {
2771   \cs_if_free:cF { ttlf@#1 }
2772   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
2773 }
2774 \cs_new_protected_nopar:Npn \__ctex_titlesec_format:Nn #1#2
2775 {
2776   \tl_set:Nx #1
2777   {
2778     \bool_if:cTF { CTEX@#2@runin }
2779     { \exp_not:N \ttlh@runin }
2780     { \exp_not:N \ttlh@hang }
2781     \tl_tail:N #1
2782   }
2783 }
2784 \cs_new_protected_nopar:Npn \__ctex_titlesec_format_explicit:Nn #1#2
2785 {
2786   \cs_set_nopar:Npx #1 ##1
2787   {
2788     \bool_if:cTF { CTEX@#2@runin }
2789     { \exp_not:N \ttlh@runin }
2790     { \exp_not:N \ttlh@hang }
2791     \exp_args:No \tl_tail:n { #1 { } }
2792   }
2793 }
2794 \cs_new_protected_nopar:Npn \__ctex_titlesec_spacing:Nn #1#2
2795 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
2796 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
2797 {
2798   \exp_not:n { {#1} {#2} {#3} {#4} }
2799   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
2800 }

(End definition for \ctex_titlesec_hook:.)

2801 \@ifpackageloaded { titlesec }
2802 { }
2803 {
2804   \ctex_at_end_package:nn { titlesec }
2805   {
2806     \@ifpackagewith { titlesec } { loadonly }
2807     { }
2808     { \ctex_titlesec_hook: }

```

```

2809     }
2810 }

```

15.8.2.6 兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题²⁰。

\ctex_titleps_hook: 我们修改 titleps 包的内部命令 \ttl@settopmark 和 \ttl@setsubmark, 将 \CTEXtheXXX 等加入更新队列中。

```

2811 \group_begin:
2812 \char_set_catcode_other:N \#
2813 \cs_new_protected_nopar:Npn \ctex_titleps_hook:
2814 {
2815   \ctex_patch_cmd:Nnn \ttl@settopmark
2816   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
2817   {
2818     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
2819     \CTEX@titlepslabel@set {#1}
2820   }
2821   \ctex_patch_cmd:Nnn \ttl@setsubmark
2822   { \protect \@namedef { the#1 } { } }
2823   {
2824     \protect \@namedef { the#1 } { }
2825     \CTEX@titlepslabel@clear {#1}
2826   }
2827   \ctex_patch_cmd:Nnn \ttl@setsubmark
2828   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
2829   {
2830     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
2831     \CTEX@titlepslabel@set {#2}
2832   }
2833 }
2834 \group_end:

```

(End definition for \ctex_titleps_hook:.)

\CTEX@titlepslabel@set
\CTEX@titlepslabel@clear 这两个函数要在随后被 \xdef 展开来获得 \CTEXtheXXX 的内容, 不应该用 \protected 来定义。

```

2835 \cs_new_nopar:Npn \CTEX@titlepslabel@set #1
2836 {
2837   \cs_if_free:cF { CTEXthe#1 }
2838   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
2839 }
2840 \cs_new_nopar:Npn \CTEX@titlepslabel@clear #1
2841 {
2842   \cs_if_free:cF { CTEXthe#1 }
2843   { \protect \@namedef { CTEXthe#1 } { } }
2844 }

```

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

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

```

2845 \ctex_at_end_package:nn { titlesec }
2846 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
2847 \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 }

```

²⁰<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。

15.8.3 目录标签的宽度

`\numberline`

```

2848 \cs_new_protected:Npn \CTEX@toc@width@n #1
2849 {
2850   \hbox_set:Nn \l__ctex_tmp_box {#1}
2851   \dim_set:Nn \@tempdima
2852   {
2853     \dim_max:nn { \@tempdima }
2854     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2855   }
2856 }
2857 \group_begin:
2858 \char_set_catcode_other:N \#
2859 \use:n
2860 {
2861   \group_end:
2862   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
2863   { \CTEX@toc@width@n {#1} }
2864   { }
2865   { \ctex_patch_failure:N \numberline }
2866   \ifpackageloaded { tocloft }
2867   {
2868     {
2869       \ctex_at_end_package:nn { tocloft }
2870       {
2871         \ctex_preto_cmd:NnnTF \numberline
2872         { \char_set_catcode_letter:n { 64 } }
2873         { \CTEX@toc@width@n {#1} }
2874         { }
2875         { \ctex_patch_failure:N \numberline }
2876       }
2877     }
2878   }

```

(End definition for `\numberline`.)

15.8.4 页眉信息的修改

`\ps@headings`

```

2879 <*article>
2880 \ctex_patch_cmd:Nnn \ps@headings
2881 { \thesection \quad }
2882 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2883 \if@twoside
2884 \ctex_patch_cmd:Nnn \ps@headings
2885 { \thesubsection \quad }
2886 { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2887 \fi:
2888 </article>
2889 <*book|report>
2890 \ctex_patch_cmd:Nnn \ps@headings
2891 { \@chapapp\ \thechapter.~\ }
2892 { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2893 \if@twoside
2894 \ctex_patch_cmd:Nnn \ps@headings
2895 { \thesection.~\ }
2896 { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2897 \fi:
2898 </book|report>

```

(End definition for \ps@headings.)

\ps@fancy 这里对 fancyhdr 宏包打补丁。原来 fancyhdr 宏包中使用 \thesection 等宏表示页眉中的章节编号,这里改用 ctex 包所用的 \CTEXthesection 系列宏。

```

2899 \ctex_at_end_package:nn { fancyhdr }
2900 {
2901   <*article>
2902     \ctex_patch_cmd:Nnn \ps@fancy
2903       { \thesection \hskip 1em \relax }
2904       { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2905     \ctex_patch_cmd:Nnn \ps@fancy
2906       { \thesubsection \hskip 1em \relax }
2907       { \ifodd \CTEX@subsection@numbering \CTEXthesubsection \quad \fi }
2908   </article>
2909   <*book|report>
2910     \ctex_patch_cmd:Nnn \ps@fancy
2911       { \@chapapp\ \thechapter.\~\ }
2912     <book> { \ifmainmatter \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi \fi }
2913     <report> { \ifodd \CTEX@chapter@numbering \CTEXthechapter \quad \fi }
2914     \ctex_patch_cmd:Nnn \ps@fancy
2915       { \thesection.\~\ }
2916       { \ifodd \CTEX@section@numbering \CTEXthesection \quad \fi }
2917   </book|report>
2918 }

```

(End definition for \ps@fancy.)

2919 </article|book|report>

15.8.5 beamer 标题页模板的修改

```

2920 <*beamer>
2921 \ExplSyntaxOff

    对应 \partpage。

2922 \defbeamertemplate*{part page}{CTEX}[1][]{
2923   \begingroup
2924   % \centering
2925   % {\usebeamerfont{part name}\usebeamercolor[fg]{part name}\partname~\insertromanpartnumber}%
2926   % \vskip1em\par
2927   \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
2928   \parindent \dimexpr \CTEX@part@indent \relax
2929   \CTEX@part@format
2930   \ifodd \CTEX@part@numbering
2931     \CTEX@partname \CTEX@part@aftername
2932   \fi
2933   \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
2934   % \usebeamerfont{part title}\insertpart\par
2935   \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
2936   \end{beamercolorbox}
2937   \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
2938   \endgroup
2939 }

```

对应 \sectionpage。

```

2940 \defbeamertemplate*{section page}{CTEX}[1][]{
2941   \begingroup
2942   % \centering
2943   % {\usebeamerfont{section name}\usebeamercolor[fg]{section name}\sectionname~\insertsectionnumber}%
2944   % \vskip1em\par
2945   \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
2946   \parindent \dimexpr \CTEX@section@indent \relax
2947   \CTEX@section@format
2948   \ifodd \CTEX@section@numbering

```

```

2949 \CTEX@sectionname \CTEX@section@aftername
2950 \fi
2951 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
2952 % \usebeamerfont{section title}\insertsection\par
2953 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
2954 \end{beamercolorbox}
2955 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
2956 \endgroup
2957 }

```

对应 \subsectionpage。

```

2958 \defbeamertemplate*{subsection page}{CTEX}[1][]{
2959 \begingroup
2960 % \centering
2961 % {\usebeamerfont{subsection name}\usebeamercolor[fg]{subsection name}\subsectionname~\insert
2962 % \vskip1em\par
2963 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
2964 \parindent \dimexpr \CTEX@subsection@indent \relax
2965 \CTEX@subsection@format
2966 \ifodd \CTEX@subsection@numbering
2967 \CTEX@subsectionname \CTEX@subsection@aftername
2968 \fi
2969 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
2970 % \usebeamerfont{subsection title}\insertsubsection\par
2971 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
2972 \end{beamercolorbox}
2973 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
2974 \endgroup
2975 }

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

2976 \defbeamertemplatealias{part page}{default}{CTEX}
2977 \defbeamertemplatealias{section page}{default}{CTEX}
2978 \defbeamertemplatealias{subsection page}{default}{CTEX}
2979 \ExplSyntaxOn
2980 </beamer>

```

15.8.6 标签引用数字的汉化

\refstepcounter 对标题进行引用时,设置标签为通过 number 选项设置的形式。

```

2981 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
2982 {
2983 \protected@edef \@currentlabel
2984 {
2985 \cs_if_exist:cTF { CTEX@the#1 }
2986 { \exp_args:cc { p@#1 } { CTEX@the#1 } }
2987 { \exp_not:o { \@currentlabel } }
2988 }
2989 }

```

(End definition for \refstepcounter.)

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

```

2990 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
2991 {
2992 \seq_map_inline:Nn \c__ctex_headings_seq
2993 { \ctex_fix_varioref_label:n { ##1 } }
2994 }

```

(End definition for \ctex_varioref_hook:.)

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

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

它的第二个参数必须放在花括号内,否则将会被作为宏的定界符号。

```
2995 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
2996 {
2997   \tl_if_empty:cT { p@#1 }
2998   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
2999 }
```

(End definition for `_ctex_fix_varioref_label:n`.)

如果 **varioref** 已经被载入,则使用它来设置。

```
3000 \@ifpackageloaded { varioref }
3001 { \ctex_varioref_hook: }
3002 {
3003   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3004   \RenewDocumentCommand \refstepcounter { m }
3005   {
3006     \CTEX@save@refstepcounter {#1}
3007     \CTEX@setcurrentlabel@n {#1}
3008   }
3009   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3010 }
```

15.8.7 载入 `<scheme>` 文件

```
3011 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3012 </class|heading>
```

15.8.8 标题格式的 `scheme` 定义

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

```
3013 <*scheme&(article|book|report|beamer)>

3014 \keys_set:nn { ctex / part }
3015 {
3016   aftertitle = \par ,
3017   <*plain>
3018   name       = \partname \space ,
3019   <*article|book|report>
3020   number     = \thepart ,
3021   </article|book|report>
3022   <*beamer>
3023   number     = \insertromanpartnumber ,
3024   </beamer>
3025   </plain>
3026   <*chinese>
3027   number     = \chinese { part } ,
3028   </chinese>
3029   <*article>
3030   beforeskip = 4ex ,
3031   afterskip  = 3ex ,
3032   <*plain>
3033   format     = \raggedright ,
3034   nameformat = \Large \bfseries ,
3035   aftername  = \par \nobreak ,
```

```

3036     titleformat = \huge \bfseries ,
3037     afterindent = false
3038 </plain>
3039 <*chinese>
3040     format      = \Large \bfseries \centering ,
3041     aftername    = \quad ,
3042     afterindent = true
3043 </chinese>
3044 </article>
3045 <*book|report>
3046     aftername    = \par \vskip 20 \p@ ,
3047     beforekip    = 0pt \@plus 1fil ,
3048     afterskip    = 0pt \@plus 1fil ,
3049     pagestyle    = plain ,
3050 <*plain>
3051     format      = \centering ,
3052     nameformat   = \huge \bfseries ,
3053     titleformat  = \Huge \bfseries
3054 </plain>
3055 <*chinese>
3056     format      = \huge \bfseries \centering
3057 </chinese>
3058 </book|report>
3059 <*beamer>
3060     format      = \centering ,
3061     nameformat   = \usebeamerfont { part ~ name }
3062                  \usebeamercolor [fg] { part ~ name } ,
3063     aftername    = \vskip 1em \par ,
3064     titleformat  = \usebeamerfont { part ~ title }
3065 </beamer>
3066 }

3067 <*book|report>
3068 \keys_set:nn { ctex / chapter }
3069 {
3070     pagestyle    = plain ,
3071     aftertitle    = \par ,
3072     beforekip    = 50 \p@ ,
3073     afterskip    = 40 \p@ ,
3074 <*plain>
3075     name         = \chaptername \space ,
3076     number       = \thechapter ,
3077     format       = \raggedright ,
3078     nameformat    = \huge \bfseries ,
3079     aftername     = \par \nobreak \vskip 20 \p@ ,
3080     titleformat   = \Huge \bfseries ,
3081     afterindent   = false
3082 </plain>
3083 <*chinese>
3084     number       = \chinese { chapter } ,
3085     format       = \huge \bfseries \centering ,
3086     aftername     = \quad ,
3087     afterindent   = true
3088 </chinese>
3089 }
3090 </book|report>

3091 <@@= >

3092 \keys_set:nn { ctex / section }
3093 {
3094 <*article|book|report>
3095     number       = \thesection ,
3096     aftername     = \quad ,
3097     aftertitle    = \@@par ,
3098     beforekip     = 3.5ex \@plus 1ex \@minus .2ex ,
3099     afterskip     = 2.3ex \@plus .2ex ,
3100     runin        = false ,

```

```

3101 <*plain>
3102     format      = \Large \bfseries ,
3103     afterindent = false
3104 </plain>
3105 <*chinese>
3106     format      = \Large \bfseries \centering ,
3107     afterindent = true
3108 </chinese>
3109 </article|book|report>
3110 <*beamer>
3111 <*plain>
3112     name        = \sectionname \space ,
3113 </plain>
3114     format      = \centering ,
3115     number      = \insertsectionnumber ,
3116     nameformat   = \usebeamerfont { section ~ name }
3117                  \usebeamercolor [fg] { section ~ name } ,
3118     aftername    = \vskip 1em \par ,
3119     titleformat  = \usebeamerfont { section ~ title } ,
3120     aftertitle   = \par
3121 </beamer>
3122 }

3123 \keys_set:nn { ctex / subsection }
3124 {
3125 <*article|book|report>
3126     number      = \thesubsection ,
3127     format      = \large \bfseries ,
3128     aftername    = \quad ,
3129     aftertitle   = \@@par ,
3130     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3131     afterskip    = 1.5ex \@plus .2ex ,
3132     runin       = false ,
3133 <*plain>
3134     afterindent = false
3135 </plain>
3136 <*chinese>
3137     afterindent = true
3138 </chinese>
3139 </article|book|report>
3140 <*beamer>
3141 <*plain>
3142     name        = \subsectionname \space ,
3143     number      = \insertsubsectionnumber ,
3144 </plain>
3145 <*chinese>
3146     number      = \arabic { section } . \arabic { subsection } ,
3147 </chinese>
3148     format      = \centering ,
3149     nameformat   = \usebeamerfont { subsection ~ name }
3150                  \usebeamercolor [fg] { subsection ~ name } ,
3151     aftername    = \vskip 1em \par ,
3152     titleformat  = \usebeamerfont { subsection ~ title } ,
3153     aftertitle   = \par
3154 </beamer>
3155 }

3156 <*article|book|report>

3157 \keys_set:nn { ctex / subsubsection }
3158 {
3159     number      = \thesubsubsection ,
3160     format      = \normalsize \bfseries ,
3161     aftername    = \quad ,
3162     aftertitle   = \@@par ,
3163     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3164     afterskip    = 1.5ex \@plus .2ex ,
3165     runin       = false ,

```



```

3166 <*plain>
3167     afterindent = false
3168 </plain>
3169 <*chinese>
3170     afterindent = true
3171 </chinese>
3172 }

3173 \keys_set:nn { ctex / paragraph }
3174 {
3175     number      = \theparagraph ,
3176     format      = \normalsize \bfseries ,
3177     aftername    = \quad ,
3178     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3179 <*plain>
3180     afterindent = false
3181 </plain>
3182 <*chinese>
3183     afterindent = true
3184 </chinese>
3185 }

3186 \keys_set:nn { ctex / subparagraph }
3187 {
3188     number      = \thesubparagraph ,
3189     format      = \normalsize \bfseries ,
3190     aftername    = \quad ,
3191     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3192 <*plain>
3193     afterindent = false
3194 </plain>
3195 <*chinese>
3196     afterindent = true
3197 </chinese>
3198 }

```

处理 sub3section 与 sub4section 的格式。

```

3199 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3200 {
3201     \keys_set:nn { ctex / paragraph }
3202     {
3203         aftertitle = \@@par ,
3204         afterskip   = 1ex \@plus .2ex ,
3205         runin       = false
3206     }
3207 }
3208 {
3209     \keys_set:nn { ctex / paragraph }
3210     {
3211         afterskip   = 1em ,
3212         runin       = true
3213     }
3214 }
3215 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
3216 {
3217     \keys_set:nn { ctex / subparagraph }
3218     {
3219         aftertitle = \@@par ,
3220         afterskip   = 1ex \@plus .2ex ,
3221         runin       = false
3222     }
3223 }
3224 {
3225     \keys_set:nn { ctex / subparagraph }
3226     {
3227         afterskip   = 1em ,
3228         runin       = true

```

```

3229     }
3230   }
3231   \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
3232   { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
3233   { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }
3234 <@@=ctex>

```

处理附录的格式。

```

3235 \keys_set:nn { ctex / appendix }
3236 <*article>
3237   { number      = \@Alph \c@section }
3238 </article>
3239 <*book|report>
3240   {
3241     name        = \appendixname \space ,
3242     number      = \@Alph \c@chapter
3243   }
3244 </book|report>
3245 </article|book|report>
3246 </scheme&(article|book|report|beamer)>

```

15.8.9 ctex.sty 的 heading 选项

```

3247 <*ctex|ctexheading>

```

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

```

3248 \clist_map_inline:nn { article , book , report , beamer }
3249 {
3250   \@ifclassloaded {#1}
3251   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3252   { }
3253 }

```

(End definition for `\c__ctex_std_class_tl`.)

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

```

3254 \msg_new:nnn { ctex } { not-standard-class }
3255 {
3256   None~of~the~standard~document~classes~was~loaded.\\
3257   Heading~`#1'~is~selected.\\
3258   ctex~may~not~work~as~expected.
3259 }
3260 <ctex>\bool_if:NTF \l__ctex_heading_bool
3261 <ctexheading>\use:n
3262 {
3263   \tl_if_exist:NTF \c__ctex_std_class_tl
3264   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3265   {
3266     \cs_if_exist:NTF \chapter
3267     {
3268       \cs_if_exist:NF \if@mainmatter
3269       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3270       \tl_const:Nn \c__ctex_class_tl { book }
3271     }
3272     { \tl_const:Nn \c__ctex_class_tl { article } }
3273     \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3274   }
3275   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
3276 }
3277 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3278 </ctex|ctexheading>

```

15.8.10 标题配置文件

```

3279 <*name>
3280 \keys_set_known:nn { ctex }
3281 {
3282     contentsname = 目录 ,
3283     listfigurename = 插图 ,
3284     listtablename = 表格 ,
3285     figurename = 图 ,
3286     tablename = 表 ,
3287     abstractname = 摘要 ,
3288     indexname = 索引 ,
3289     bibname = 参考文献 ,
3290     appendixname = 附录 ,
3291     proofname = 证明 ,
3292     algorithmname = 算法 ,
3293     refname = 参考文献 ,
3294     continuation = (续) ,
3295     part / name = { 第 , 部分 } ,
3296     chapter / name = { 第 , 章 }
3297 }
3298 </name>

```

15.9 chinese 方案的其他设置

```

3299 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

3300 <article|book|report>\pagestyle { headings }

```

汉化标题名字。

```

3301 <*generic>
3302 \tl_if_exist:NTF \c__ctex_std_class_tl
3303 {
3304     \str_if_eq:onTF { \c__ctex_std_class_tl } { beamer }
3305     {
3306         \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3307         {
3308             \uselanguage { ChineseGBK }
3309             \languagealias { chinese } { ChineseGBK }
3310             \ctex_file_input:n { ctex-name-gbk.cfg }
3311         }
3312         {
3313             \uselanguage { ChineseUTF8 }
3314             \languagealias { chinese } { ChineseUTF8 }
3315             \ctex_file_input:n { ctex-name-utf8.cfg }
3316         }
3317         \clist_put_left:Nn \trans@languagepath { chinese }
3318     }
3319     {
3320         \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3321         { \ctex_file_input:n { ctex-name-gbk.cfg } }
3322         { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

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。

```

3323     \RequirePackage { indentfirst }
3324 }
3325 }
3326 {
3327     \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3328     { \ctex_file_input:n { ctex-name-gbk.cfg } }
3329     { \ctex_file_input:n { ctex-name-utf8.cfg } }
3330 }
3331 </generic>

```

```

3332 <!*generic>
3333 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
3334 <*article|book|report>
3335 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3336 { \ctex_file_input:n { ctex-name-utf8.cfg } }
3337 </article|book|report>
3338 <*beamer>
3339 {
3340   \uselanguage { ChineseGBK }
3341   \languagealias { chinese } { ChineseGBK }
3342   \ctex_file_input:n { ctex-name-gbk.cfg }
3343 }
3344 {
3345   \uselanguage { ChineseUTF8 }
3346   \languagealias { chinese } { ChineseUTF8 }
3347   \ctex_file_input:n { ctex-name-utf8.cfg }
3348 }
3349 \clist_put_left:Nn \trans@languagepath { chinese }
3350 </beamer>
3351 </!generic>

```

日期格式。

```

3352 \keys_set:nn { ctex } { today = small }

```

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

```

3353 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
3354 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }

```

若用户未设置宏包选项 `linespread`, 则设置行距初始值为 $1.3 \times 1.2 = 1.56$ 倍字体大小。

```

3355 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3356 { \keys_set:nn { ctex / option } { linespread = 1.3 } }

```

若用户未设置宏包选项 `autoindent`, 则自动调整首行缩进。

```

3357 \ctex_if_autoindent_touched:F
3358 { \keys_set:nn { ctex } { autoindent = true } }
3359 </scheme&chinese>

```

15.10 中文字号

```

3360 <*class|ctex|ctexsize>

```

`\zihao`

```

3361 \NewDocumentCommand \zihao { m }
3362 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

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

`\ctex_zihao:n`

```

3363 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
3364 {
3365   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3366   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3367   { \msg_error:nnn { ctex } { fontsize } {#1} }
3368 }
3369 \msg_new:nnnn { ctex } { fontsize }
3370 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3371 {
3372   The~old~font~size~is~used~if~you~continue.\\
3373   The~available~font~sizes~are~listed~as~follow.\\
3374   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3375 }

```

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

15.10.1 定义中文字号

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

```

3376 \prop_new:N \c__ctex_font_size_prop
3377 \seq_new:N \c__ctex_font_size_seq
3378 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
3379 {
3380   \use:x
3381   {
3382     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3383     {
3384       { \dim_to_decimal:n {#2} }
3385       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
3386     }
3387   }
3388   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3389 }
3390 \clist_map_inline:nn
3391 {
3392   { 8 } { 5 bp } ,
3393   { 7 } { 5.5 bp } ,
3394   { -6 } { 6.5 bp } ,
3395   { 6 } { 7.5 bp } ,
3396   { -5 } { 9 bp } ,
3397   { 5 } { 10.5 bp } ,
3398   { -4 } { 12 bp } ,
3399   { 4 } { 14 bp } ,
3400   { -3 } { 15 bp } ,
3401   { 3 } { 16 bp } ,
3402   { -2 } { 18 bp } ,
3403   { 2 } { 22 bp } ,
3404   { -1 } { 24 bp } ,
3405   { 1 } { 26 bp } ,
3406   { -0 } { 36 bp } ,
3407   { 0 } { 42 bp }
3408 }
3409 { \__ctex_save_font_size:nn #1 }

```

(End definition for `\c__ctex_font_size_prop`.)

`\ctex_declare_math_sizes:nnnn`

```

3410 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3411 {
3412   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3413   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3414 }

```

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

`__ctex_get_font_sizes:Nn`

```

3415 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
3416 {
3417   \tl_clear:N #1
3418   \tl_map_inline:nn {#2}
3419   {
3420     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3421     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3422     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3423   }
3424 }

```

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

(End definition for `_ctex_get_font_sizes:Nn`.)

```

3425 \clist_map_inline:nn
3426 {
3427   { 8 }{ 8 }{ 5pt }{ 5pt } ,
3428   { 7 }{ 7 }{ 5pt }{ 5pt } ,
3429   { -6 }{ -6 }{ 5pt }{ 5pt } ,
3430   { 6 }{ 6 }{ 5pt }{ 5pt } ,
3431   { -5 }{ -5 }{ 6pt }{ 5pt } ,
3432   { 5 }{ 5 }{ 7pt }{ 5pt } ,
3433   { -4 }{ -4 }{ 8pt }{ 6pt } ,
3434   { 4 }{ 4 }{ 5 }{ 6 } ,
3435   { -3 }{ -3 }{ -4 }{ -5 } ,
3436   { 3 }{ 3 }{ 4 }{ 5 } ,
3437   { -2 }{ -2 }{ -3 }{ -4 } ,
3438   { 2 }{ 2 }{ 3 }{ 4 } ,
3439   { -1 }{ -1 }{ -2 }{ -3 } ,
3440   { 1 }{ 1 }{ 2 }{ 3 } ,
3441   { -0 }{ -0 }{ -1 }{ -2 } ,
3442   { 0 }{ 0 }{ 1 }{ 2 }
3443 }
3444 { \ctex_declare_math_sizes:nmmm #1 }

```

15.10.2 修改默认字号大小

`\ctex_set_font_size:Nnn`

```

3445 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3446 {
3447   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3448   { \exp_after:wN \_ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3449   { \msg_error:nmm { ctex } { fontsize } {#2} }
3450 }
3451 \cs_new_protected:Npn \_ctex_set_font_size:nnNn #1#2#3#4
3452 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

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

```

3453 \if_case:w \g__ctex_font_size_flag
3454 \ctex_file_input:n { ctex-c5size.clo }
3455 \or:
3456 \ctex_file_input:n { ctex-cs4size.clo }
3457 \fi:
3458 </class>ctex<ctexsize>
3459 <*c5size>
3460 \ctex_set_font_size:Nnn \normalsize { 5 }
3461 {
3462   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3463   \abovedisplayshortskip \z@ \@plus3\p@
3464   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3465   \belowdisplayskip \abovedisplayskip
3466   \let\@listi\@listI
3467 }
3468 \ctex_set_font_size:Nnn \small { -5 }
3469 {
3470   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3471   \abovedisplayshortskip \z@ \@plus2\p@
3472   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3473   \def\@listi{\leftmargin\leftmarginI
3474     \topsep 4\p@ \@plus2\p@ \@minus2\p@
3475     \parsep 2\p@ \@plus\p@ \@minus\p@
3476     \itemsep \parsep}
3477   \belowdisplayskip \abovedisplayskip
3478 }
3479 \ctex_set_font_size:Nnn \footnotesize { 6 }

```

```

3480 {
3481   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3482   \abovedisplayshortskip \z@ \@plus\p@
3483   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3484   \def\@listi{\leftmargin\leftmargini
3485             \topsep 3\p@ \@plus\p@ \@minus\p@
3486             \parsep 2\p@ \@plus\p@ \@minus\p@
3487             \itemsep \parsep}
3488   \belowdisplayskip \abovedisplayskip
3489 }
3490 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3491 \ctex_set_font_size:Nnn \tiny { 7 } { }
3492 \ctex_set_font_size:Nnn \large { -4 } { }
3493 \ctex_set_font_size:Nnn \Large { -3 } { }
3494 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3495 \ctex_set_font_size:Nnn \huge { 2 } { }
3496 \ctex_set_font_size:Nnn \Huge { 1 } { }
3497 </cs5size>
3498 <*cs4size>
3499 \ctex_set_font_size:Nnn \normalsize { -4 }
3500 {
3501   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3502   \abovedisplayshortskip \z@ \@plus3\p@
3503   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3504   \belowdisplayskip \abovedisplayskip
3505   \let\@listi\@listI
3506 }
3507 \ctex_set_font_size:Nnn \small { 5 }
3508 {
3509   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3510   \abovedisplayshortskip \z@ \@plus3\p@
3511   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3512   \def\@listi{\leftmargin\leftmargini
3513             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3514             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3515             \itemsep \parsep}
3516   \belowdisplayskip \abovedisplayskip
3517 }
3518 \ctex_set_font_size:Nnn \footnotesize { -5 }
3519 {
3520   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3521   \abovedisplayshortskip \z@ \@plus3\p@
3522   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3523   \def\@listi{\leftmargin\leftmargini
3524             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3525             \parsep 3\p@ \@plus2\p@ \@minus\p@
3526             \itemsep \parsep}
3527   \belowdisplayskip \abovedisplayskip
3528 }
3529 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3530 \ctex_set_font_size:Nnn \tiny { -6 } { }
3531 \ctex_set_font_size:Nnn \large { -3 } { }
3532 \ctex_set_font_size:Nnn \Large { -2 } { }
3533 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3534 \ctex_set_font_size:Nnn \huge { -1 } { }
3535 \ctex_set_font_size:Nnn \Huge { 1 } { }
3536 </cs4size>
3537 <ctexsize>\normalsize
3538 <*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。

```

3539 \fp_compare:nNf { \l__ctex_line_spread_fp } ? { \c_zero_fp }
3540 {
3541     \exp_args:Nx \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

`\footnotesep` 我们调整了行距, 可能导致脚注的间距与行距不协调, 需要调整 `\footnotesep`。标准文档类对 `\footnotesep` 的设置是, 字体大小为 `\footnotesize` 时 `\strutbox` 的高度 (默认值是 `.7\baselineskip`)。我们沿用这个设置方法, 只需要更新具体的大小。

```

3542     \group_begin: \footnotesize \exp_args:NNNo \group_end:
3543     \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3544 }

```

(End definition for `\footnotesep`.)

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

```

3545 \normalsize

```

15.12 其它功能

CTeX `ctex-faq.sty` 中的定义是

```

\DeclareRobustCommand\CTeX{${\mathbb{C}}$\kern-.05em\TeX}

```

然而 `\mathbb` 未必有定义, 这里就不采用它了, 只定义最简单的形式。同 `hologo` 宏包的设置类似, `CTeX` 可以用在 `\csname` 和 PDF 书签中。

```

3546 \NewDocumentCommand \CTeX { }
3547 { \ifincsname CTeX \else: C \TeX \fi: }
3548 \ctex_at_end_package:n { hyperref }
3549 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

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

`captiondelimitter` 过时选项。

```

3550 \keys_define:nn { ctex }
3551 {
3552     captiondelimitter .code:n =
3553     {
3554         \msg_warning:nnn { ctex } { deprecated-option }
3555         { You~can~load~the~package~caption'~to~get~its~functionality. }
3556     }
3557 }

```

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

```

3558 </class|ctex>

```

15.12.1 列表环境的缩进

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

```

3559 <*scheme&chinese&(article|book|report)>
3560 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3561 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3562 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3563 </scheme&chinese&(article|book|report)>

```

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

```

3564 <*class|ctex>

```


15.13 载入中文字体

`\ctex_fontset_error:n` 字库不可用时给出紧急错误信息, 停止读取定义文件。

```

3565 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
3566 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3567 \msg_new:nnn { ctex } { fontset-unavailable }
3568 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }

```

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

`\ctex_load_fontset:` 如果用户没有指定字体, 则探测操作系统, 载入相应的字体配置。

```

3569 \cs_new_protected_nopar:Npn \ctex_load_fontset:
3570 {
3571   \tl_if_empty:NTF \g__ctex_fontset_tl
3572   {
3573     \ctex_detected_platform:
3574     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3575   }
3576   {
3577     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
3578     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
3579     {
3580       \use:x
3581       {
3582         \ctex_detected_platform:
3583         \msg_error:nnxx { ctex } { fontset-not-found }
3584         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
3585       }
3586       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
3587     }
3588   }
3589 }
3590 \@onlypreamble \ctex_load_fontset:
3591 \msg_new:nnnn { ctex } { fontset-not-found }
3592 {
3593   CTeX~fontset~`#1'~could~not~be~found.\\
3594   Fontset~`#2'~will~be~used~instead.
3595 }
3596 { You~may~run~`mktexlsr'~firstly. }

```

(End definition for `\ctex_load_fontset:`.)

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

3597 \keys_define:nn { ctex }
3598 {
3599   fontset .code:n =
3600   {
3601     \ctex_if_preamble:TF
3602     {
3603       \str_if_eq:x:nnTF {#1} { none }
3604       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3605       {
3606         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3607         {
3608           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3609           \ctex_load_fontset:
3610         }
3611         {
3612           \msg_error:nnxx { ctex } { fontset-loaded }
3613           { \g__ctex_fontset_tl } {#1}
3614         }
3615       }
3616     }
3617     { \msg_error:nn { ctex } { fontset-only-preamble } }

```

```

3618     }
3619   }
3620   \msg_new:nnnn { ctex } { fontset-loaded }
3621   {
3622     CTeX~fontset~`#1'~has~been~loaded.
3623     \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
3624   }
3625   { Only~one~fontset~can~be~loaded~in~the~preamble. }
3626   \msg_new:nnn { ctex } { fontset-only-preamble }
3627   {
3628     The~`fontset'~option~can~be~used~only~in~preamble.
3629   }

```

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

载入中文字库。

```

3630 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3631 { \ctex_load_fontset: }

```

15.14 宏包配置文件

15.14.1 ctex.cfg

```

3632 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3633 </class|ctex>
3634 <*config>
3635 %%
3636 </config>

```

15.14.2 ctexopts.cfg

这里仅为配置文件示例：使用 Windows Vista 或以后版本的字体设置。

```

3637 <*ctexopts>
3638 %%
3639 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3640 </ctexopts>

```

15.15 字体定义文件

15.15.1 传统定义方式

```

3641 <*c19|c70>
3642 %%
3643 %% Chinese characters
3644 %%
3645 <c19>%% character set: GBK (extension of GB 2312)
3646 <c70>%% character set: Unicode
3647 %% font encoding: Unicode
3648 %%
3649 </c19|c70>

```

CJK 宏包使用的字体族。

```

3650 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3651 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3652 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3653 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3654 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3655 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

upTeX 使用的字体族。upTeX 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。

```

3656 <*uptex>
3657 \DeclareKanjiFamily{JY2}{rm}{-}{-}
3658 \DeclareKanjiFamily{JY2}{sf}{-}{-}
3659 \DeclareKanjiFamily{JY2}{tt}{-}{-}
3660 \DeclareKanjiFamily{JT2}{rm}{-}{-}
3661 \DeclareKanjiFamily{JT2}{sf}{-}{-}
3662 \DeclareKanjiFamily{JT2}{tt}{-}{-}
3663 </uptex>

3664 <*rm>
3665 <*c19>
3666 \DeclareFontShape{C19}{rm}{m}{n}{-}{-} CJK * gbksong{\CJKnormal}
3667 \DeclareFontShape{C19}{rm}{b}{n}{-}{-} CJK * gbkhei{\CJKnormal}
3668 \DeclareFontShape{C19}{rm}{bx}{n}{-}{-} CJK * gbkhei{\CJKnormal}
3669 \DeclareFontShape{C19}{rm}{m}{sl}{-}{-} CJK * gbksongsl{\CJKnormal}
3670 \DeclareFontShape{C19}{rm}{b}{sl}{-}{-} CJK * gbkheisl{\CJKnormal}
3671 \DeclareFontShape{C19}{rm}{bx}{sl}{-}{-} CJK * gbkheisl{\CJKnormal}
3672 \DeclareFontShape{C19}{rm}{m}{it}{-}{-} CJK * gbkkai{\CJKnormal}
3673 \DeclareFontShape{C19}{rm}{b}{it}{-}{-} CJKb * gbkkai{\CJKbold}
3674 \DeclareFontShape{C19}{rm}{bx}{it}{-}{-} CJKb * gbkkai{\CJKbold}
3675 </c19>
3676 <*c70>
3677 \DeclareFontShape{C70}{rm}{m}{n}{-}{-} CJK * unisong{\CJKnormal}
3678 \DeclareFontShape{C70}{rm}{b}{n}{-}{-} CJK * unihei{\CJKnormal}
3679 \DeclareFontShape{C70}{rm}{bx}{n}{-}{-} CJK * unihei{\CJKnormal}
3680 \DeclareFontShape{C70}{rm}{m}{sl}{-}{-} CJK * unisongsl{\CJKnormal}
3681 \DeclareFontShape{C70}{rm}{b}{sl}{-}{-} CJK * uniheisl{\CJKnormal}
3682 \DeclareFontShape{C70}{rm}{bx}{sl}{-}{-} CJK * uniheisl{\CJKnormal}
3683 \DeclareFontShape{C70}{rm}{m}{it}{-}{-} CJK * unikai{\CJKnormal}
3684 \DeclareFontShape{C70}{rm}{b}{it}{-}{-} CJKb * unikai{\CJKbold}
3685 \DeclareFontShape{C70}{rm}{bx}{it}{-}{-} CJKb * unikai{\CJKbold}
3686 </c70>
3687 </rm>
3688 <*uptex>
3689 \DeclareFontShape{JY2}{rm}{m}{n}{-}{-} upzhserif-h{}
3690 \DeclareFontShape{JY2}{rm}{m}{it}{-}{-} upzhserifit-h{}
3691 \DeclareFontShape{JY2}{rm}{bx}{n}{-}{-} upzhserifb-h{}
3692 \DeclareFontShape{JT2}{rm}{m}{n}{-}{-} upzhserif-v{}
3693 \DeclareFontShape{JT2}{rm}{m}{it}{-}{-} upzhserifit-v{}
3694 \DeclareFontShape{JT2}{rm}{bx}{n}{-}{-} upzhserifb-v{}
3695 </uptex>

3696 <*sf>
3697 <*c19>
3698 \DeclareFontShape{C19}{sf}{m}{n}{-}{-} CJK * gbkyou{\CJKnormal}
3699 \DeclareFontShape{C19}{sf}{b}{n}{-}{-} CJKb * gbkyou{\CJKbold}
3700 \DeclareFontShape{C19}{sf}{bx}{n}{-}{-} CJKb * gbkyou{\CJKbold}
3701 \DeclareFontShape{C19}{sf}{m}{sl}{-}{-} CJK * gbkyousl{\CJKnormal}
3702 \DeclareFontShape{C19}{sf}{b}{sl}{-}{-} CJKb * gbkyousl{\CJKbold}
3703 \DeclareFontShape{C19}{sf}{bx}{sl}{-}{-} CJKb * gbkyousl{\CJKbold}
3704 \DeclareFontShape{C19}{sf}{m}{it}{-}{-} CJK * gbkyou{\CJKnormal}
3705 \DeclareFontShape{C19}{sf}{b}{it}{-}{-} CJKb * gbkyou{\CJKbold}
3706 \DeclareFontShape{C19}{sf}{bx}{it}{-}{-} CJKb * gbkyou{\CJKbold}
3707 </c19>
3708 <*c70>
3709 \DeclareFontShape{C70}{sf}{m}{n}{-}{-} CJK * uniyousl{\CJKnormal}
3710 \DeclareFontShape{C70}{sf}{b}{n}{-}{-} CJKb * uniyousl{\CJKbold}
3711 \DeclareFontShape{C70}{sf}{bx}{n}{-}{-} CJKb * uniyousl{\CJKbold}
3712 \DeclareFontShape{C70}{sf}{m}{sl}{-}{-} CJK * uniyousl{\CJKnormal}
3713 \DeclareFontShape{C70}{sf}{b}{sl}{-}{-} CJKb * uniyousl{\CJKbold}
3714 \DeclareFontShape{C70}{sf}{bx}{sl}{-}{-} CJKb * uniyousl{\CJKbold}
3715 \DeclareFontShape{C70}{sf}{m}{it}{-}{-} CJK * uniyousl{\CJKnormal}
3716 \DeclareFontShape{C70}{sf}{b}{it}{-}{-} CJKb * uniyousl{\CJKbold}
3717 \DeclareFontShape{C70}{sf}{bx}{it}{-}{-} CJKb * uniyousl{\CJKbold}
3718 </c70>
3719 </sf>
3720 <*uptex>
3721 \DeclareFontShape{JY2}{sf}{m}{n}{-}{-} upzhsans-h{}

```

```

3722 \DeclareFontShape{JY2}{sf}{bx}{n}{<-> upzhsansb-h}{}
3723 \DeclareFontShape{JT2}{sf}{m}{n}{<-> upzhsans-v}{}
3724 \DeclareFontShape{JT2}{sf}{bx}{n}{<-> upzhsansb-v}{}
3725 </uptex>

3726 <*tt>
3727 <*c19>
3728 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
3729 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
3730 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
3731 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
3732 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3733 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3734 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
3735 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
3736 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
3737 </c19>
3738 <*c70>
3739 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
3740 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
3741 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
3742 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
3743 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
3744 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
3745 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
3746 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
3747 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
3748 </c70>
3749 </tt>
3750 <*uptex>
3751 \DeclareFontShape{JY2}{tt}{m}{n}{<-> upzhmono-h}{}
3752 \DeclareFontShape{JT2}{tt}{m}{n}{<-> upzhmono-v}{}
3753 </uptex>
3754 <*fontset>

```

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

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

```

3755 <*windows>
3756 \file_if_exist:nTF { C:/boot.ini }
3757 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3758 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3759 </windows>

```

旧的 Windows 字体设置使用黑体作为无衬线体，楷体和仿宋是 GB2312 编码；新的 Windows 字体设置使用微软雅黑作为无衬线体，楷体和仿宋是大字库。Windows 8 以后，微软雅黑由原来的 .ttf 后缀改为 .ttc 后缀，需要加以区分。

```

3760 <*windowsnew>
3761 \tl_new:N \l__ctex_msyh_suffix
3762 \tl_set:Nn \l__ctex_msyh_suffix { .ttc }
3763 \file_if_exist:nF { C:/Windows/Fonts/msyh.ttc }
3764 {
3765   \file_if_exist:nF { msyh.ttc }
3766   { \tl_set:Nn \l__ctex_msyh_suffix { .ttf } }
3767 }
3768 </windowsnew>
3769 <*windowsnew|windowsold>
3770 \sys_if_engine_pdfTeX:TF
3771 {
3772   \ctex_zhmap_case:nnn
3773   {
3774     \ctex_punct_set:n { windows }
3775     \setCJKmainfont

```

```

3776         [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3777 <*windowsold>
3778     \setCJKsansfont { simhei.ttf }
3779     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3780 </windowsold>
3781 <*windowsnew>
3782     \setCJKsansfont [ BoldFont = msyhbd\l__ctex_msyh_suffix ] { msyh\l__ctex_msyh_suffix }
3783     \setCJKfamilyfont { zhyahei }
3784         [ BoldFont = msyhbd\l__ctex_msyh_suffix ] { msyh\l__ctex_msyh_suffix }
3785     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3786     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3787 </windowsnew>
3788     \setCJKmonofont { simfang.ttf }
3789     \setCJKfamilyfont { zhkai } { simkai.ttf }
3790     \setCJKfamilyfont { zhfs } { simfang.ttf }
3791     \setCJKfamilyfont { zhsong } { simsun.ttc }
3792     \setCJKfamilyfont { zhhei } { simhei.ttf }
3793     \setCJKfamilyfont { zhli } { simli.ttf }
3794     \setCJKfamilyfont { zhyou } { simyou.ttf }
3795     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3796     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3797     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3798     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3799 }
3800 {
3801     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3802     \ctex_punct_set:n { windows }
3803     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3804     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3805     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3806 }
3807 {
3808     \tl_set:Nn \CJKrmdefault { rm }
3809     \tl_set:Nn \CJKsfdefault { sf }
3810     \tl_set:Nn \CJKttdefault { tt }
3811 }
3812 }
3813 {
3814     \sys_if_engine_uptex:TF
3815     {
3816         \ctex_set_upfonts:nnnnnn
3817         {simsun.ttc} {simhei.ttf} {simkai.ttf}
3818 <windowsnew>         {msyh\l__ctex_msyh_suffix} {msyhbd\l__ctex_msyh_suffix}
3819 <windowsold>         {simhei.ttf} {simhei.ttf}
3820         {simfang.ttf}
3821         \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3822         \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
3823         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3824         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3825 <windowsnew>         \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
3826         \ctex_set_upfamily:nnn { zhli } { upschrn } {}
3827         \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
3828         \ctex_set_upfamily:nnn { zhyou } { upschg } {}
3829         \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
3830     }
3831     {
3832 <*windowsold>
3833         \setCJKmainfont
3834         [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3835         \setCJKsansfont { SimHei }
3836         \setCJKmonofont { FangSong_GB2312 }
3837         \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3838         \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3839 </windowsold>
3840 <*windowsnew>
3841         \setCJKmainfont
3842         [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }

```

```

3843 \setCJKsansfont
3844 [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3845 \setCJKmonofont { FangSong }
3846 \setCJKfamilyfont { zhkai } { KaiTi }
3847 \setCJKfamilyfont { zhfs } { FangSong }
3848 </windowsnew>
3849 \setCJKfamilyfont { zhsong } { SimSun }
3850 \setCJKfamilyfont { zhhei } { SimHei }
3851 \setCJKfamilyfont { zhli } { LiSu }
3852 \setCJKfamilyfont { zhyou } { YouYuan }
3853 <*windowsnew>
3854 \setCJKfamilyfont { zhyahei }
3855 [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3856 </windowsnew>
3857 }
3858 }
3859 </windowsnew|windowsold>

```

15.15.3 ctex-fontset-adobe.def

```

3860 <*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。

```

3861 \sys_if_engine_pdftex:TF
3862 {
3863   \sys_if_output_pdf:TF
3864   { \ctex_fontset_error:n { adobe } }
3865   {
3866     \ctex_zhmap_case:nnn
3867     {
3868       \setCJKmainfont
3869       [
3870         cmap = UniGB-UTF16-H ,
3871         BoldFont = AdobeHeitiStd-Regular.otf ,
3872         ItalicFont = AdobeKaitiStd-Regular.otf
3873       ] { AdobeSongStd-Light.otf }
3874       \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3875       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3876       \setCJKfamilyfont { zhsong }
3877       [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3878       \setCJKfamilyfont { zhhei }
3879       [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3880       \setCJKfamilyfont { zhkai }
3881       [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3882       \setCJKfamilyfont { zhfs }
3883       [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3884       \ctex_punct_set:n { adobe }
3885       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3886       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3887       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3888       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3889       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3890     }
3891     {
3892       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3893       \ctex_punct_set:n { adobe }
3894       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3895       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3896       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3897     }
3898     { \ctex_fontset_error:n { adobe } }
3899   }
3900 }
3901 {
3902   \sys_if_engine_uptex:TF

```

```

3903 {
3904   \ctex_set_upfonts:nnnnnn
3905   {AdobeSongStd-Light.otf} {AdobeHeitiStd-Regular.otf} {AdobeKaitiStd-Regular.otf}
3906   {AdobeHeitiStd-Regular.otf} {AdobeHeitiStd-Regular.otf}
3907   {AdobeFangsongStd-Regular.otf}
3908   \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
3909   \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
3910   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3911   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3912 }
3913 {
3914   \setCJKmainfont
3915   [
3916     BoldFont = AdobeHeitiStd-Regular ,
3917     ItalicFont = AdobeKaitiStd-Regular
3918   ] { AdobeSongStd-Light }
3919   \setCJKsansfont { AdobeHeitiStd-Regular}
3920   \setCJKmonofont { AdobeFangsongStd-Regular}
3921   \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3922   \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3923   \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3924   \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3925 }
3926 }
3927 </adobe>
3928 <*fandol>

```

15.15.4 ctex-fontset-fandol.def

```

3929 \sys_if_engine_pdftex:TF
3930 {
3931   \sys_if_output_pdf:TF
3932   { \ctex_fontset_error:n { fandol } }
3933   {
3934     \ctex_zhmap_case:nnn
3935     {
3936       \setCJKmainfont
3937       [
3938         cmap = UniGB-UTF16-H ,
3939         BoldFont = FandolSong-Bold.otf ,
3940         ItalicFont = FandolKai-Regular.otf
3941       ] { FandolSong-Regular.otf }
3942       \setCJKsansfont
3943       [
3944         cmap = UniGB-UTF16-H ,
3945         BoldFont = FandolHei-Bold.otf
3946       ] { FandolHei-Regular.otf }
3947       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3948       \setCJKfamilyfont { zhsong }
3949       [
3950         cmap = UniGB-UTF16-H ,
3951         BoldFont = FandolSong-Bold.otf
3952       ] { FandolSong-Regular.otf }
3953       \setCJKfamilyfont { zhhei }
3954       [
3955         cmap = UniGB-UTF16-H ,
3956         BoldFont = FandolHei-Bold.otf
3957       ] { FandolHei-Regular.otf }
3958       \setCJKfamilyfont { zhfs }
3959       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3960       \setCJKfamilyfont { zhkai }
3961       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
3962       \ctex_punct_set:n { fandol }
3963       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3964       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3965       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3966       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }

```



```

3967         \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3968         \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
3969     }
3970     {
3971         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
3972         \ctex_punct_set:n { fandol }
3973         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3974         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3975         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3976     }
3977     { \ctex_fontset_error:n { fandol } }
3978 }
3979 }
3980 {
3981     \sys_if_engine_uptex:TF
3982     {
3983         \ctex_set_upfonts:nnnnnn
3984         {FandolSong-Regular.otf} {FandolSong-Bold.otf} {FandolKai-Regular.otf}
3985         {FandolHei-Regular.otf} {FandolHei-Bold.otf}
3986         {FandolFang-Regular.otf}
3987         \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
3988         \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
3989         \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
3990         \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
3991     }
3992     {
3993         \setCJKmainfont
3994         [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
3995         { FandolSong-Regular }
3996         \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3997         \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
3998         \setCJKfamilyfont { zhsong }
3999         [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
4000         \setCJKfamilyfont { zhhei }
4001         [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
4002         \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
4003         \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
4004     }
4005 }
4006 </fandol>
4007 <*mac>

```

15.15.5 ctex-fontset-mac.def

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

```

/Library/Fonts/Baoli.ttc: 报隶-简,Baoli SC:style=Regular
/Library/Fonts/Hannotate.ttc: 手札体-简,Hannotate SC:style=Bold
/Library/Fonts/Hannotate.ttc: 手札体-简,Hannotate SC:style=Regular
/Library/Fonts/HanziPen.ttc: 翩翩体-简,HanziPen SC:style=Bold
/Library/Fonts/HanziPen.ttc: 翩翩体-简,HanziPen SC:style=Regular
/Library/Fonts/Hei.ttf: Hei
/Library/Fonts/Hiragino Sans GB W3.otf: 冬青黑体简体中文 W3,Hiragino Sans GB W3
/Library/Fonts/Hiragino Sans GB W6.otf: 冬青黑体简体中文 W6,Hiragino Sans GB W6
/Library/Fonts/Kai.ttf: Kai
/Library/Fonts/Kaiti.ttc: STKaiti
/Library/Fonts/Kaiti.ttc: 楷体-简,Kaiti SC:style=Black
/Library/Fonts/Kaiti.ttc: 楷体-简,Kaiti SC:style=Bold
/Library/Fonts/Kaiti.ttc: 楷体-简,Kaiti SC:style=Regular
/Library/Fonts/Lantinghei.ttc: 兰亭黑-简,Lantinghei SC:style=Demibold
/Library/Fonts/Lantinghei.ttc: 兰亭黑-简,Lantinghei SC:style=Heavy
/Library/Fonts/Lantinghei.ttc: 兰亭黑-简,Lantinghei SC:style=Extralight
/Library/Fonts/Libian.ttc: 隶变-简,Libian SC
/Library/Fonts/Songti.ttc: STSong
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Black

```

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


```

/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Bold
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Light
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Regular
/Library/Fonts/WawaSC-Regular.otf: 娃娃体-简,Wawati SC
/Library/Fonts/WeibeiSC-Bold.otf: 魏碑-简,Weibei SC
/Library/Fonts/Xingkai.ttc: 行楷-简,Xingkai SC:style=Bold
/Library/Fonts/Xingkai.ttc: 行楷-简,Xingkai SC:style=Light
/Library/Fonts/Yuanti.ttc: 圆体-简,Yuanti SC:style=Bold
/Library/Fonts/Yuanti.ttc: 圆体-简,Yuanti SC:style=Light
/Library/Fonts/Yuanti.ttc: 圆体-简,Yuanti SC:style=Regular
/Library/Fonts/YuppySC-Regular.otf: 雅痞-简,Yuppy SC
/Library/Fonts/华文仿宋.ttf: STFangsong
/Library/Fonts/华文细黑.ttf: STHeiti:style=Light
/Library/Fonts/华文黑体.ttf: STHeiti:style=Regular
/System/Library/Fonts/STHeiti Light.ttc: 黑体-简,Heiti SC:style=Light
/System/Library/Fonts/STHeiti Medium.ttc: 黑体-简,Heiti SC:style=Medium

```

在 DVIPDFMx 下,可以通过下述方式使用华文宋体和华文楷体:

```

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

```

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

```

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

```

DVIPDFMx 不能生成 PDF,报下述错误:

```

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

```

如果将 CMap 改为 UniGB-UTF16-H,错误信息是

```

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

```

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

```

4008 \sys_if_engine_pdftex:TF
4009 { \ctex_fontset_error:n { mac } }
4010 {
4011   \sys_if_engine_uptex:TF
4012   { \ctex_fontset_error:n { mac } }
4013   {
4014     \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
4015     \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
4016     \setCJKmonofont { STFangsong }
4017     \setCJKfamilyfont { zhsong } { STSong }
4018     \setCJKfamilyfont { zhhei } { STHeiti }
4019     \setCJKfamilyfont { zhfs } { STFangsong }
4020     \setCJKfamilyfont { zhkai } { STKaiti }
4021   }
4022 }

4023 </mac>
4024 <*founder>

```

15.15.6 ctex-fontset-founder.def

```

4025 \sys_if_engine_pdftex:TF
4026 {
4027   \ctex_zhmap_case:nnn

```

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

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

```

4028 {
4029   \setCJKmainfont
4030     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
4031   \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
4032   \setCJKmonofont { FZFSK.TTF }
4033   \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
4034   \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4035   \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4036   \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4037   \setCJKfamilyfont { zhli } { FZLSK.TTF }
4038   \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
4039   \ctex_punct_set:n { founder }
4040   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4041   \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4042   \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4043   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4044   \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
4045   \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4046   \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4047 }
4048 {
4049   \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4050   \ctex_punct_set:n { founder }
4051   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4052   \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4053   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4054 }
4055 { \ctex_fontset_error:n { founder } }
4056 }
4057 {
4058   \sys_if_engine_uptex:TF
4059   {
4060     \ctex_set_upfonts:nnnnnn
4061       { FZSSK.TTF } { FZXBSK.TTF } { FZKTK.TTF }
4062       { FZXH1K.TTF } { FZHTK.TTF }
4063       { FZFSK.TTF }
4064     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4065     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4066     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4067     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4068     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4069     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4070     \ctex_set_upfamily:nnn { zhyou } { upschg } {}
4071     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4072   }
4073   {
4074     \setCJKmainfont
4075       [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }

```

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

```

4076   \setCJKsansfont [ BoldFont = FZHei-B01 ]
4077     { \sys_if_engine_xetex:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
4078   \setCJKmonofont { FZFangSong-Z02 }
4079   \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
4080   \setCJKfamilyfont { zhhei } { FZHei-B01 }
4081   \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4082   \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4083   \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4084   \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
4085 }

```

```

4086 }
4087 </founder>
4088 <*ubuntu>

```

15.15.7 ctex-fontset-ubuntu.def

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

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

需要注意的是 uming.ttc 和 ukai.ttc 看起来像有四种字形的样子,但其实只有“令”和“骨”这区区两个字有新字形,其余都取旧字形²⁵。

```

4089 \sys_if_engine_pdftex:TF
4090 {
4091   \ctex_zhmap_case:nnn
4092   {
4093     \setCJKmainfont
4094       [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
4095     \setCJKsansfont { wqy-zenhei.ttc }
4096     \setCJKmonofont { uming.ttc }
4097     \setCJKfamilyfont { zhsong } { uming.ttc }
4098     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
4099     \setCJKfamilyfont { zhkai } { ukai.ttc }
4100     \ctex_punct_set:n { ubuntu }
4101     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4102     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4103     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4104     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4105     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4106   }
4107   {
4108     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4109     \ctex_punct_set:n { ubuntu }
4110     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4111     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4112     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4113   }
4114   { \ctex_fontset_error:n { ubuntu } }
4115 }
4116 {
4117   \sys_if_engine_uptex:TF
4118   {
4119     \ctex_set_upfonts:nnnnnn
4120       { uming.ttc } { wqy-zenhei.ttc } { ukai.ttc }
4121       { wqy-zenhei.ttc } { wqy-zenhei.ttc }
4122       { uming.ttc }
4123     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4124     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4125     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4126   }
4127   {
4128     \setCJKmainfont
4129       [ BoldFont = WenQuanYi~Zen~Hei , ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
4130     \setCJKsansfont { WenQuanYi~Zen~Hei }
4131     \setCJKmonofont { AR~PL~UMing~CN }
4132     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
4133     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
4134     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
4135   }
4136 }
4137 </ubuntu>

```

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

15.15.8 中文字体命令

```

4138 <!*windows>
4139 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4140 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
4141 <!ubuntu>\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4142 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4143 <*windowsnew|windowsold|founder>
4144 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4145 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4146 </windowsnew|windowsold|founder>
4147 <windowsnew>\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4148 </!windows>
4149 </fontset>
4150 <*zhmap>

```

15.15.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4151 \begingroup\catcode61\catcode48\catcode32=10\relax%
4152 \catcode 35=6 % #
4153 \catcode123=1 % {
4154 \catcode125=2 % }
4155 \toks0{\endlinechar=\the\endlinechar\relax}%
4156 \toks2{\endlinechar=13 }%
4157 \def\x#1 #2 {%
4158   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4159   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4160 \x 13 5 % carriage return
4161 \x 32 10 % space
4162 \x 35 6 % #
4163 \x 40 12 % (
4164 \x 41 12 % )
4165 \x 45 12 % -
4166 \x 46 12 % .
4167 \x 47 12 % /
4168 \x 58 12 % :
4169 \x 60 12 % <
4170 \x 61 12 % =
4171 \x 64 11 % @
4172 \x 91 12 % [
4173 \x 93 12 % ]
4174 \x 123 1 % {
4175 \x 125 2 % }
4176 \edef\x#1{\endgroup%
4177   \edef\noexpand#1{%
4178     \the\toks0 %
4179     \let\noexpand\noexpand\noexpand#1%
4180     \noexpand\noexpand\noexpand\noexpand\undefined%
4181     \noexpand\noexpand\noexpand\endinput}%
4182     \the\toks2}%
4183 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4184 \begingroup\expandafter\endgroup
4185 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4186 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4187   \expandafter\ifx\csname pdfoutput\endcsname\relax
4188     iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4189   \else ifctexpdf\fi
4190 \endcsname

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4191 \begingroup
4192 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4193   \long\def\x#1\ProvidesFile#2[#3]{%

```

```

4194     #1%
4195     \immediate\write-1{File: #2 #3}%
4196     \expandafter\xdef\csname ver@#2\endcsname{#3}}
4197     \expandafter\x%
4198 \fi
4199 \endgroup

```

15.15.9.1 zhwindowsfonts.tex

```

4200 <*windows>
4201 \ProvidesFile{zhwindowsfonts.tex}%
4202 [2016/04/23 v2.4 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
4203
4204 \ifzhmappdf
4205   \pdfmapline{=gbk@UGBK@ <simsun.ttc}
4206   \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
4207   \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4208   \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4209   \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4210   \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4211   \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4212
4213   \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
4214   \pdfmapline{=unisong@Unicode@ <simsun.ttc}
4215   \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4216   \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4217   \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4218   \pdfmapline{=unili@Unicode@ <simli.ttf}
4219   \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4220
4221   \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
4222   \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4223   \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4224   \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4225   \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4226   \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4227
4228   \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
4229   \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4230   \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4231   \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4232   \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4233   \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4234
4235 \else
4236   \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
4237   \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
4238   \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4239   \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4240   \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4241   \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4242   \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4243
4244   \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
4245   \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
4246   \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4247   \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4248   \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4249   \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4250   \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4251
4252   \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
4253   \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
4254   \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
4255   \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
4256   \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
4257   \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}

```

```

4258
4259 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
4260 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
4261 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
4262 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
4263 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
4264 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
4265
4266 </windows>

```

15.15.9.2 zhadobefonts.tex

```

4267 <*adobe>
4268 \ProvidesFile{zhadobefonts.tex}%
4269 [2016/04/23 v2.4 Adobe font map loader for DVIPDFMx (CTEX)]
4270
4271 \ifzhmappdf
4272 %% pdfTeX does not support OTF fonts
4273
4274 \else
4275 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4276 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4277 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4278 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4279 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4280
4281 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4282 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4283 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4284 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4285 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4286
4287 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4288 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4289 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4290 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4291
4292 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4293 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4294 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4295 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4296
4297 </adobe>

```

15.15.9.3 zhfandolfonts.tex

```

4298 <*fandol>
4299 \ProvidesFile{zhfandolfonts.tex}%
4300 [2016/04/23 v2.4 Fandol font map loader for DVIPDFMx (CTEX)]
4301
4302 \ifzhmappdf
4303 %% pdfTeX does not support OTF fonts
4304
4305 \else
4306 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4307 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4308 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4309 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4310 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4311
4312 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4313 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4314 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4315 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4316 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4317
4318 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4319 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}

```

```

4320 \special{pdf:mapline gbkhaisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4321 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4322
4323 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4324 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4325 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4326 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4327
4328 </fandol>

```

15.15.9.4 zhfounderfonts.tex

```

4329 <*founder>
4330 \ProvidesFile{zhfounderfonts.tex}%
4331 [2016/04/23 v2.4 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
4332
4333 \ifzhmappdf
4334 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4335 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4336 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4337 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4338 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4339 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4340 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4341
4342 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4343 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4344 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4345 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4346 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4347 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4348 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4349
4350 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4351 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4352 \pdfmapline{=gbkhaisl@UGBK@ <FZHTK.TTF}
4353 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4354 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4355 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4356
4357 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4358 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4359 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4360 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4361 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4362 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4363
4364 \else
4365 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4366 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4367 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4368 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4369 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4370 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4371 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4372
4373 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4374 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4375 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4376 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4377 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4378 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4379 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF}
4380
4381 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4382 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4383 \special{pdf:mapline gbkhaisl@UGBK@ unicode FZHTK.TTF -s .167}
4384 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}

```



```

4385 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4386 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4387
4388 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4389 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4390 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4391 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4392 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4393 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4394
4395 </founder>

```

15.15.9.5 zhubuntufonts.tex

```

4396 <*ubuntu>
4397 \ProvidesFile{zhubuntufonts.tex}%
4398 [2016/04/23 v2.4 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
4399
4400 \ifzhmappdf
4401 \pdfmapline{=gbk@UGBK@ <uming.ttc}
4402 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
4403 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
4404 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
4405 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
4406 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
4407
4408 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
4409 \pdfmapline{=unisong@Unicode@ <uming.ttc}
4410 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
4411 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
4412 \pdfmapline{=unifs@Unicode@ <uming.ttc}
4413 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
4414
4415 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
4416 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
4417 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
4418 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
4419 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
4420
4421 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
4422 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
4423 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
4424 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
4425 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
4426
4427 \else
4428 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
4429 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
4430 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
4431 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
4432 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
4433
4434 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
4435 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
4436 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
4437 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
4438 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
4439
4440 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
4441 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
4442 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
4443 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
4444
4445 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
4446 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
4447 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
4448 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
4449

```



```

4450 </ubuntu>
4451 \fi
4452
4453 \ctex@zhmap@endinput
4454 </zhmap>

```

15.15.10 制作 spa 文件

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

```

4455 <*spa>
4456 <*macro>
4457 \input expl3-generic %
4458 \ExplSyntaxOn
4459 \sys_if_engine_xetex:F
4460 {
4461   \msg_new:nn { ctex } { xetex }
4462     { XeTeX~is~required~to~compile~this~document! }
4463   \msg_fatal:nn { ctex } { xetex }
4464 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

4465 \seq_new:N \c__ctex_punct_seq
4466 \seq_gset_from_clist:Nn \c__ctex_punct_seq
4467 {
4468   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4469   "3008 , "300A , "3016 , "3010 ,
4470   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4471   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4472   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4473 }

```

\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。

```

4474 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
4475 {
4476   \iow_open:Nn \g__ctex_spa_iow {#1}
4477   \clist_map_inline:nn {#2}
4478     { \__ctex_write_family:nn ##1 }
4479   \iow_close:N \g__ctex_spa_iow
4480 }
4481 \iow_new:N \g__ctex_spa_iow
4482 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

(End definition for \ctex_make_spa:nn.)

4483 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4484 {
4485   \group_begin:
4486   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4487   \l__ctex_punct_font
4488   \clist_clear:N \l__ctex_punct_bounds_clist
4489   \seq_map_inline:Nn \c__ctex_punct_seq
4490     { \exp_args:No \__ctex_save_bounds:n { \int_use:N \xetex_charglyph:D ##1 } }
4491   \iow_now:Nx \g__ctex_spa_iow
4492   {
4493     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4494         { \l__ctex_punct_bounds_clist , , , }
4495     }
4496     \group_end:
4497 }
4498 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
4499 {
4500     \clist_put_right:Nx \l__ctex_punct_bounds_clist
4501     {
4502         \__ctex_calc_bounds:nn { \c_one } {#1} ,
4503         \__ctex_calc_bounds:nn { \c_three } {#1}
4504     }
4505 }
4506 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1em 的比值的一百倍。

```

4507 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
4508 {
4509     \fp_eval:n
4510     {
4511         round
4512         (
4513             \dim_to_decimal_in_unit:nn
4514             { 100 \xetex_glyphbounds:D #1 ~ #2 }
4515             { 1 em }
4516         )
4517     }
4518 }
4519 \ExplSyntaxOff
4520 <macro>

```

下面是 CTeX 定义的一些字体。

```

4521 <*make>
4522 \input ctexspamacro %
4523
4524 \MAKESPA {ctexpunct.spa}
4525 {
4526     {adobezhhsong}      {AdobeSongStd-Light} ,
4527     {adobezhhei}       {AdobeHeitiStd-Regular} ,
4528     {adobezhkai}       {AdobeKaitiStd-Regular} ,
4529     {adobezhfs}        {AdobeFangsongStd-Regular} ,
4530     {fandolzhhsong}    {FandolSong} ,
4531     {fandolzhsongb}    {FandolSong-Bold} ,
4532     {fandolzhhei}     {FandolHei} ,
4533     {fandolzhheib}    {FandolHei-Bold} ,
4534     {fandolzhkai}     {FandolKai} ,
4535     {fandolzhfs}      {FandolFang} ,
4536     {founderzhhsong}   {FZShuSong-Z01} ,
4537     {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
4538     {founderzhhei}     {FZHei-B01} ,
4539     {founderzhheil}    {FZXiHeiI-Z08} ,
4540     {founderzhkai}    {FZKai-Z03} ,
4541     {founderzhfs}     {FZFangSong-Z02} ,
4542     {founderzhli}     {FZLiShu-S01} ,
4543     {founderzhyou}    {FZXiYuan-M01} ,
4544     {founderzhzyoub}   {FZZhunYuan-M02} ,
4545     {ubuntuzhsong}     {AR PL UMinG CN} ,
4546     {ubuntuzhhei}     {WenQuanYi Zen Hei} ,
4547     {ubuntuzhkai}     {AR PL UKai CN} ,
4548     {windowszhhsong}  {SimSun} ,
4549     {windowszhhei}    {SimHei} ,
4550     {windowszhkai}    {KaiTi} ,
4551     {windowszhfs}     {FangSong} ,
4552     {windowszhli}     {LiSu} ,
4553     {windowszhyou}    {YouYuan} ,
4554     {windowszhyahui}  {Microsoft YaHei} ,
4555     {windowszhyahuib} {Microsoft YaHei Bold}

```

```

4556 }
4557
4558 \primitive\end
4559 </make>
4560 </spa>

```

15.16 translator 宏包的中文字典

```

4561 <*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。

```

4562 <*theorem>
4563 \providetranslation{Comments}{评论}
4564 \providetranslation{comments}{评论}
4565 \providetranslation{Comment}{评论}
4566 \providetranslation{comment}{评论}
4567 \providetranslation{Corollaries}{推论}
4568 \providetranslation{corollaries}{推论}
4569 \providetranslation{Corollary}{推论}
4570 \providetranslation{corollary}{推论}
4571 \providetranslation{Definitions}{定义}
4572 \providetranslation{definitions}{定义}
4573 \providetranslation{Definition}{定义}
4574 \providetranslation{definition}{定义}
4575 \providetranslation{Examples}{例}
4576 \providetranslation{examples}{例}
4577 \providetranslation{Example}{例}
4578 \providetranslation{example}{例}
4579 \providetranslation{Exercises}{练习}
4580 \providetranslation{exercises}{练习}
4581 \providetranslation{Exercise}{练习}
4582 \providetranslation{exercise}{练习}
4583 \providetranslation{Facts}{事实}
4584 \providetranslation{facts}{事实}
4585 \providetranslation{Fact}{事实}
4586 \providetranslation{fact}{事实}
4587 \providetranslation{Key Lemmas}{关键引理}
4588 \providetranslation{key lemmas}{关键引理}
4589 \providetranslation{Key Lemma}{关键引理}
4590 \providetranslation{key lemma}{关键引理}
4591 \providetranslation{Key Observations}{关键观察}
4592 \providetranslation{key observations}{关键观察}
4593 \providetranslation{Key Observation}{关键观察}
4594 \providetranslation{key observation}{关键观察}
4595 \providetranslation{Lemmas}{引理}
4596 \providetranslation{lemmas}{引理}
4597 \providetranslation{Lemma}{引理}
4598 \providetranslation{lemma}{引理}
4599 \providetranslation{Main Theorems}{主要定理}
4600 \providetranslation{main theorems}{主要定理}
4601 \providetranslation{Main Theorem}{主要定理}
4602 \providetranslation{main theorem}{主要定理}
4603 \providetranslation{Observations}{观察}
4604 \providetranslation{observations}{观察}
4605 \providetranslation{Observation}{观察}
4606 \providetranslation{observation}{观察}
4607 \providetranslation{Problems}{问题}
4608 \providetranslation{problems}{问题}
4609 \providetranslation{Problem}{问题}
4610 \providetranslation{problem}{问题}
4611 \providetranslation{Proofs}{证明}
4612 \providetranslation{proofs}{证明}
4613 \providetranslation{Proof}{证明}
4614 \providetranslation{proof}{证明}

```

```

4615 \providetranslation{Proof Sketch}{证明提要}
4616 \providetranslation{Proof sketch}{证明提要}
4617 \providetranslation{proof sketch}{证明提要}
4618 \providetranslation{Proof Sketches}{证明提要}
4619 \providetranslation{Proof sketches}{证明提要}
4620 \providetranslation{proof sketches}{证明提要}
4621 \providetranslation{Sketch of Proof}{证明提要}
4622 \providetranslation{Sketch of Proofs}{证明提要}
4623 \providetranslation{Sketch of proof}{证明提要}
4624 \providetranslation{Sketch of proofs}{证明提要}
4625 \providetranslation{sketch of proof}{证明提要}
4626 \providetranslation{sketch of proofs}{证明提要}
4627 \providetranslation{Propositions}{命题}
4628 \providetranslation{propositions}{命题}
4629 \providetranslation{Proposition}{命题}
4630 \providetranslation{proposition}{命题}
4631 \providetranslation{Remarks}{注}
4632 \providetranslation{remarks}{注}
4633 \providetranslation{Remark}{注}
4634 \providetranslation{remark}{注}
4635 \providetranslation{Solutions}{解}
4636 \providetranslation{solutions}{解}
4637 \providetranslation{Solution}{解}
4638 \providetranslation{solution}{解}
4639 \providetranslation{Theorems}{定理}
4640 \providetranslation{theorems}{定理}
4641 \providetranslation{Theorem}{定理}
4642 \providetranslation{theorem}{定理}
4643 </theorem>
4644 </dict>

```

15.17 ctexcap 宏包

```
4645 <*ctexcap>
```

ctexcap 是过时宏包。

```

4646 \clist_new:N \l__ctex_ctexcap_options_clist
4647 \clist_set:Nx \l__ctex_ctexcap_options_clist
4648 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
4649 \msg_new:nnn { ctexcap } { deprecated }
4650 {
4651   Package~`ctexcap'~is~deprecated.\\
4652   Please~use~package~`ctex'~with~option~`#1'~instead: \\\
4653   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
4654 }
4655 \msg_warning:nnx { ctexcap } { deprecated }
4656 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

4657 \PassOptionsToPackage { heading = true } { ctexcap }
4658 \RequirePackageWithOptions { ctex }
4659 </ctexcap>

```

15.18 ctexhook 宏包

```
4660 <*ctexhook>
```

实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
4661 \cs_new_protected:Npn \ctex_at_end_preamble:n #1
4662 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl {#1} }
4663 \cs_new_protected:Npn \ctex_after_end_preamble:n #1
4664 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl {#1} }
4665 \cs_new_protected_nopar:Npn \CTEX@document@left@hook
4666 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }

```

```

4667 \cs_new_protected_nopar:Npn \CTEX@document@right@hook
4668 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
4669 \cs_set_nopar:Npx \document
4670 {
4671   \CTEX@document@left@hook
4672   \exp_not:o { \document }
4673   \CTEX@document@right@hook
4674 }
4675 \tl_new:N \g__ctex_end_preamble_hook_tl
4676 \tl_new:N \g__ctex_after_end_preamble_hook_tl

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

```

`\ctex_at_end_package:nn` 与 `filehook` 的 `\AtEndOfPackageFile*` 类似, 如果原来没有在载入宏包则在宏包末尾执行语句, 否则立即执行。

```

4677 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
4678 {
4679   \@ifpackageloaded {#1}
4680   {#2}
4681   { \ctex_gadd_hook:cn { g__ctex_at_end_ #1 _hook_tl } {#2} }
4682 }

(End definition for \ctex_at_end_package:nn.)

```

`\ctex_gadd_hook:Nn` 给钩子附加内容。

```

\ctex_gadd_hook:cn
4683 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1#2
4684 {
4685   \tl_if_exist:NF #1 { \tl_new:N #1 }
4686   \tl_gput_right:Nn #1 {#2}
4687 }
4688 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

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

```

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

```

\ctex_package_end_hook:o
4689 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
4690 {
4691   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
4692   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
4693 }
4694 \cs_generate_variant:Nn \ctex_package_end_hook:n { o }

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

```

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

```

4695 \tl_put_left:Nn \@popfilename
4696 {
4697   \cs_if_eq:NNT \@currentx \pkgextension
4698   { \ctex_package_end_hook:o { \@currname } }
4699 }

4700 </ctexhook>

```

15.19 ctexpatch 宏包

```

4701 <*ctexpatch>
4702 \cs_if_exist:NF \str_new:N { \RequirePackage { l3str } }

```

`\ctex_patch_cmd_once:NnnnTF` 只进行第一次匹配进行替换。参数 #2 是宏重建时的 `\catcode` 设置。

```

4703 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2

```

```

4704 {
4705   \ctex_patch_boot:NNnnTF \_ctex_patch_cmd:Nnnnw #1
4706   { once } {#2} { \use_i:nn } { \use_ii:nn }
4707 }

```

(End definition for \ctex_patch_cmd_once:NnnTF.)

\ctex_patch_cmd_all:NnnTF 替换所有匹配到的文本。

```

4708 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnTF #1#2
4709 {
4710   \ctex_patch_boot:NNnnTF \_ctex_patch_cmd:Nnnnw #1
4711   { all } {#2} { \use_i:nn } { \use_ii:nn }
4712 }

```

(End definition for \ctex_patch_cmd_all:NnnTF.)

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

```

4713 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
4714 {
4715   \ctex_patch_boot:NNnnTF \_ctex_patch_cmd:Nnnnw #1
4716   { once }
4717   {
4718     \ExplSyntaxOff
4719     \char_set_catcode_letter:n { 64 }
4720   }
4721   { }
4722   { \ctex_patch_failure:N #1 }
4723 }
4724 \cs_new_protected:Npn \ctex_patch_failure:N #1
4725 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
4726 \msg_new:nnn { ctex } { patch-failure }
4727 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

(End definition for \ctex_patch_cmd:Nnn.)

\ctex_preto_cmd:NnnTF 在宏的原本定义前面增加钩子。

```

4728 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
4729 {
4730   \ctex_patch_boot:NNnnTF \_ctex_hookto_cmd:Nnnnw #1
4731   { left } {#2} { \use_i:nn } { \use_ii:nn }
4732 }

```

(End definition for \ctex_preto_cmd:NnnTF.)

\ctex_appto_cmd:NnnTF 在宏的原本定义后面追加钩子。

```

4733 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
4734 {
4735   \ctex_patch_boot:NNnnTF \_ctex_hookto_cmd:Nnnnw #1
4736   { right } {#2} { \use_i:nn } { \use_ii:nn }
4737 }

```

(End definition for \ctex_appto_cmd:NnnTF.)

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

```

4738 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
4739 {
4740   \tl_set:Nn \_ctex_patch_true:w {#5}
4741   \tl_set:Nn \_ctex_patch_false:w {#6}
4742   \group_begin:

```

```

4743     \char_set_catcode_other:n { 35 }
4744     \ctex_parse_name:NN #1 #2 {#3} {#4}
4745   }
4746   \tl_new:N \__ctex_patch_true:w
4747   \tl_new:N \__ctex_patch_false:w

```

(End definition for \ctex_patch_boot:NNnTF.)

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

```

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

```

ctexpatch 的主要原理是先对宏的 \meaning 作字符串替换, 然后再用 \scantokens 来重建它。我们希望对宏的实际定义打补丁, 为此需要先得到对应的名字。letltxmacro、show2e 和 xpatch 宏包中都有类似的工作。

```

4748 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
4749 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
4750 \group_begin:
4751 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
4752 {
4753   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
4754   {
4755     \bool_if:nTF
4756     {
4757       \cs_if_exist_p:c { ##3 ~ } ||
4758       \cs_if_exist_p:c { #1##3 }
4759     }
4760     {
4761       \group_begin:
4762       \use:x
4763       {
4764         \group_end:
4765         \__ctex_parse_name:nNNNnN
4766         { \token_get_replacement_spec:N ##2 }
4767         \exp_not:N ##2
4768         \exp_not:c { ##3 ~ }
4769         \exp_not:c { #1##3 }
4770       } {##3} ##1
4771     }
4772     { ##1##2 }
4773   }
4774 \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
4775 {
4776   \exp_args:Nc ##6
4777   {
4778     \str_case:nnTF {##1}
4779     {
4780       { \protect ##3 } { }
4781       { \x@protect ##2 \protect ##3 } { }
4782     }
4783     {
4784       \str_if_eq_x:nnTF
4785       { \exp_not:n { #1@protected@ ##3 #1##3 } }
4786       {
4787         \exp_last_unbraced:Nf \__ctex_parse_name:w
4788         \token_get_replacement_spec:N ##3 #3 ~ #2 \q_stop

```

```

4789         }
4790         { #1##5 ~ } { ##5 ~ }
4791     }
4792     {
4793         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
4794         {
4795             { #1@protected@ ##2 ##4 } { }
4796             { #1@ ##4 } { }
4797         }
4798         { #1##5 } {##5}
4799     }
4800 }
4801 }
4802 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
4803 }
4804 \use:x
4805 {
4806     \__ctex_tmp:w
4807     { \c_backslash_str }
4808     { \c_left_brace_str }
4809     { \tl_to_str:n { testopt } }
4810 }
4811 \group_end:
4812 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

(End definition for \ctex_parse_name:NN.)

\l__ctex_prefix_str 分别保存宏的 \meaning 中的前缀、参数文本和替换文本。
\l__ctex_parameter_str
\l__ctex_replacement_str

```

4813 \str_new:N \l__ctex_prefix_str
4814 \str_new:N \l__ctex_parameter_str
4815 \str_new:N \l__ctex_replacement_str

```

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

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

```

\__ctex_get_macro_meaning:w
4816 \group_begin:
4817     \cs_set_protected:Npn \__ctex_tmp:w #1
4818     {
4819         \prg_new_protected_conditional:Npnn
4820         \ctex_get_macro_meaning:N ##1 { TF }
4821         {
4822             \exp_after:wN \__ctex_get_macro_meaning:w
4823             \token_to_meaning:N ##1 \q_mark #1 -> \q_no_value \q_mark \q_stop
4824         }
4825         \cs_new_protected:Npn \__ctex_get_macro_meaning:w
4826         ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
4827         {
4828             \tl_set:Nn \l__ctex_replacement_tl {##3}
4829             \quark_if_no_value:NTF \l__ctex_replacement_tl
4830             { \prg_return_false: }
4831             {
4832                 \str_set:Nn \l__ctex_prefix_str {##1}
4833                 \str_set:Nn \l__ctex_parameter_str {##2}
4834                 \str_set:Nn \l__ctex_replacement_str {##3}
4835                 \prg_return_true:
4836             }
4837         }
4838     }
4839     \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
4840 \group_end:

```

(End definition for \ctex_get_macro_meaning:NTF.)

\ctex_if_rescanable:NnTF 检查宏是否可以重建。


```

4841 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
4842 {
4843   \ctex_get_macro_meaning:NnTF #1
4844   {
4845     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
4846     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
4847   }
4848   {#4}
4849 }
4850 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

(End definition for \ctex_if_rescanable:NnTF.)

```

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

```

4851 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
4852 {
4853   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
4854   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
4855   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
4856   \use:x
4857   {
4858     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
4859     \exp_not:o { \l__ctex_parameter_tl }
4860     { \exp_not:o { \l__ctex_replacement_tl } }
4861   }
4862 }
4863 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
4864 {
4865   \str_if_empty:NTF #2
4866   { \tl_clear:N #1 }
4867   { \tl_set_rescan:Nnx #1 {#3} {#2} }
4868 }
4869 \tl_new:N \l__ctex_prefix_tl
4870 \tl_new:N \l__ctex_parameter_tl
4871 \tl_new:N \l__ctex_replacement_tl

(End definition for \__ctex_patch_rebuild:Nn.)

```

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

```

4872 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
4873 {
4874   \group_end:
4875   \ctex_if_rescanable:NnTF #1 {#3}
4876   {
4877     \use:x
4878     {
4879       \__ctex_patch_replace:nnnTF {#2}
4880       { \tl_to_str:n {#4} }
4881       { \tl_to_str:n {#5} }
4882     }
4883     {
4884       \__ctex_patch_rebuild:Nn #1 {#3}
4885       \__ctex_patch_true:w
4886     }
4887     { \__ctex_patch_false:w }
4888   }
4889   { \__ctex_patch_false:w }
4890 }

(End definition for \__ctex_patch_cmd:Nnnnw.)

```

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

```

4891 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4

```

```

4892 {
4893   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
4894   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
4895 }

```

(End definition for __ctex_patch_replace:nnnTF.)

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

4896 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
4897 {
4898   \group_end:
4899   \ctex_get_macro_meaning:NTF #1
4900   {
4901     \str_if_empty:NTF \l__ctex_parameter_str
4902     { \__ctex_hookto_cmd_parameterless:Nnnw #1 {#2} {#4} }
4903     { \__ctex_hookto_cmd_parameter:Nnnnw #1 {#2} {#3} {#4} }
4904   }
4905   { \__ctex_patch_false:w }
4906 }

```

(End definition for __ctex_hookto_cmd:Nnnnw.)

__ctex_hookto_cmd_parameterless:Nnnw 如果宏没有参数,可以直接进行附加操作。注意保持宏的前缀。

```

4907 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnw #1#2#3
4908 {
4909   \str_if_empty:NF \l__ctex_prefix_str
4910   { \tl_rescan:nx { } { \l__ctex_prefix_str } }
4911   \tex_edef:D #1
4912   {
4913     \use:c { __ctex_ #2 _hook_aux:nn }
4914     { \exp_not:o {#1} }
4915     { \exp_not:n {#3} }
4916   }
4917   \__ctex_patch_true:w
4918 }
4919 \cs_generate_variant:Nn \tl_rescan:nn { nx }
4920 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
4921 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

(End definition for __ctex_hookto_cmd_parameterless:Nnnw.)

__ctex_hookto_cmd_parameter:Nnnnw 如果宏有参数,需要在字符串中进行附加,然后再重建。

```

4922 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
4923 {
4924   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
4925   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
4926   {
4927     \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
4928     \__ctex_patch_rebuild:Nn #1 {#3}
4929     \__ctex_patch_true:w
4930   }
4931   { \__ctex_patch_false:w }
4932 }

```

(End definition for __ctex_hookto_cmd_parameter:Nnnnw.)

4933 </ctexpatch>

版本历史

v2.0

General: c5size, cs4size 是过时选项。	36
captiondelimitter 是过时选项。	108
fancyhdr 成为过时选项, 原选项功能总是打开。	39
fntef 成为过时选项, 原选项功能总是打开。	39
hyperref 成为过时选项, 原选项功能总是打开。	39
indent, noindent 是过时选项。	37
nofonts, adobefonts, winfonts 是过时选项。	37
nopunct 是过时选项。	38
nospace 是过时选项。	38
nozhmap 是过时选项。	38
punct 选项可以设置标点格式。	38
ctex 宏包新增 heading 选项。	39
\CTEXindent, \CTEXnoindent 是过时命令。	80
\CTEXsetup, \CTEXoptions 是过时命令。	14, 41
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; CTExfilltwosides 是过时环境。	74
标题设置新增 pagestyle 选项。	84
调整 \footnotesep 的大小, 以适合行距的变化。	108
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。	40
将标题汉化功能加入 ctex.sty。	82
将中文字号功能提取到可以独立使用的 ctexsize。	104
解决 etoolbox 与 breqn 关于 \end 的冲突。	108
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	74
删除 c19gbsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jā 宏包支持 Lua \TeX 。	48
新增 autoindent 选项。	36
新增 fontset 选项。	37
新增 linespread 选项。	36
新增 linestretch 选项。	78
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。	39
新增 zhmcjk 支持选项。	38
新增 zihao 选项。	36
新增统一设置接口 \ctexset。	41
应用 \LaTeX 3 重新整理代码。	1
中文字号不再采用近似值。	105
自动检测操作系统, 载入对应的字体配置。	71
\CJK@surr: 解决与 \nouppercase 的冲突。	44

v2.0.1

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

v2.0.2

General: 修复加载 ctex 宏包后章节标题后第一段无段首缩进的问题。	103
---	-----

v2.1

General: format+, nameformat+ 等带加号的选项, 加号与前面的文字之间可以有可选的空格。	83
nameformat 可以接受章节名字为参数。	83
不依赖 ifpdf 宏包。	34
不再设置 hyperref 宏包的 colorlinks 选项。	72
给 pdf \TeX 下的非 UTF8 编码 CJK 字体族加上 CMap。	42
将章节标题设置功能提取到可以独立使用的宏包 ctexheading 中。	1
新的标题格式选项 aftertitle。	83
修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。	98
\ctex_if_macosx:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	72
hyperref: 补充定义 \hypersetup。	39

v2.2

General: beforeskip 和 afterskip 选项的符号不再有特殊意义。	83
beforeskip、afterskip 和 indent 选项支持表达式。	85
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/beforeskip 和 part/afterskip 选项有意义。	85
给 enumitem 宏包注册 \chinese 和 \zhnum。	81
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。	128
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。	98
删去 etoolbox 与 breqn 的兼容补丁。	108
新的标题格式选项 afterindent。	83
新的标题格式选项 numbering。	83
新的标题格式选项 runin。	83
新增子宏包 ctexpatch 实现给宏打补丁功能。	129

v2.3

General: .value_required: 和 .value_forbidden: 已过时。	36
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。	34
更新 LuaTeX-jā 支持 (20150922.0)。	48
更新 unicode-math 宏包补丁。	48
兼容 titleps 宏包。	94
修复 nameformat 作用域问题。	83
与 \LaTeX 3 (2015/12/20) 同步。	32

v2.4

General: 初步支持 up \LaTeX 。	69
加强 beamer 宏包支持。	1
提供 translator 宏包的中文定理名称翻译。	127
提供 up \LaTeX 的 NFSS 字体定义。	110
正确更新 CJK 包的 \CJKfamilydefault。	70
正确设置 up \TeX 下字体命令。	69
\ctex_parse_name:NN: 修复宏名解析错误。	131

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

_	2891, 2895, 2911, 2915
\#	489, 2812, 2858
.../afterindent	20
.../aftername	18
.../aftername+	18
.../afterskip	20
.../aftertitle	19
.../aftertitle+	19
.../beforeskip	20
.../format	16
.../format+	16
.../indent	21
.../name	15
.../nameformat	17
.../nameformat+	17
.../number	16
.../numberformat	17
.../numberformat+	17
.../titleformat	18
.../titleformat+	18
\/	1638
\\	15, 33, 34, 100, 143, 145, 147, 203, 213, 675, 1118, 1200, 2217, 3256, 3257, 3372, 3373, 3593, 3623, 4651, 4652, 4653, 4727
\{	1129, 1130, 4653
\}	1129, 1130, 4653
10pt	8, <u>40</u>
11pt	8, <u>40</u>
12pt	8, <u>40</u>
A	
\abovedisplayshortskip	3463, 3471, 3482, 3502, 3510, 3521
\abovedisplayskip	3462, 3465, 3470, 3477, 3481, 3488, 3501, 3504, 3509, 3516, 3520, 3527
\abstractname	2201
abstractname	12
\addCJKfontfeature	1200, 1223
\addCJKfontfeatures	59, 1218, 1223
\addcontentsline	2399, 2400, 2403, 2406, 2429, 2430, 2433, 2436, 2510, 2515, 2518, 2521, 2619, 2635
\AddEnumerateCounter	2162, 2164, 2165
\addpenalty	2586, 2587
\addtocontents	2524, 2525
\addvspace	2366, 2367, 2524, 2525, 2586, 2587, 2927, 2937, 2945, 2955, 2963, 2973
adobefonts	28
\algorithmname	2208, 2224
algorithmname	12
AlternateFont	25, <u>61</u>
\appendix	92, 2727, 2728

appendix/name	22, <u>92</u>
appendix/number	22, <u>92</u>
appendix/numbering	21, <u>92</u>
\appendixname	2203, 3241
appendixname	12
\arabic	3146
\AtBeginDocument	577, 1627
\AtBeginDvi	447
\AtBeginShipoutFirst	449, 1814, 1830
\AtBeginUTFCommand	719
\AtEndOfClass	133
\AtEndOfPackage	134
\AtEndUTFCommand	725
autoindent	10, <u>36</u> , <u>78</u>

B

backslash commands:

\c_backslash_str	4807
\baselinestretch	2073
\begin	2933, 2951, 2969
\begingroup	2610, 2649, 2679, 2923, 2941, 2959, 4151, 4184, 4191
\belowdisplayshortskip	3464, 3472, 3483, 3503, 3511, 3522
\belowdisplayskip	3465, 3477, 3488, 3504, 3516, 3527
\bfdefault	981, 987, 1313, 1320
\bfseries	2412, 2415, 2443, 2446, 2457, 2470, 2541, 2547, 2561, 3034, 3036, 3040, 3052, 3053, 3056, 3078, 3080, 3085, 3102, 3106, 3127, 3160, 3176, 3189
\bibname	2206, 2209, 2217, 2225, 2231, 2232
bibname	12

bool commands:

\bool_gset_false:N	263, 268
\bool_gset_true:N	258
\bool_if:cTF	2778, 2788, 2799
\bool_if:NT	1158
\bool_if:NTF	68, 3260
\bool_if:nTF	4755
\bool_new:N	129, 1195
\bool_set_false:N	1170
\bool_set_true:N	1188

box commands:

\box_ht:N	3543
\box_new:N	59
\box_wd:N	1900, 2854

C

c5size	28
cap	28
captiondelimiter	29, <u>108</u>
\catcode	4151, 4152, 4153, 4154, 4158, 4159
CCT	28
CCTfont	28

- \ccwd 23, 75, 113, 186, 1900, 1904, 1907, 1909, 1946, 1947, 1948, 2024, 2062, 2081, 2082, 2089, 2116, 2117, 2118, 2121, 2123, 2124, 2135, 2146, 2147, 3560, 3561, 3562
- \centering 2440, 2468, 2924, 2942, 2960, 3040, 3051, 3056, 3060, 3085, 3106, 3114, 3148
- \chapter 87, 2492, 3266
- chapter/beforeskip 30
- chapter/numbering 15
- chapter/pagestyle 21, 84
- \chaptermark 2523
- \chaptername 3075
- char commands:
 - \char_generate:nn 745
 - \char_set_catcode:nn 71, 550, 551
 - \char_set_catcode_active:N 1348
 - \char_set_catcode_letter:n 65, 1630, 2872, 4719
 - \char_set_catcode_math_superscript:n 545
 - \char_set_catcode_other:N 489, 914, 1331, 2812, 2858
 - \char_set_catcode_other:n 544, 4743
 - \char_value_catcode:n 71, 550, 551
- CharRange 25, 61
- \Chinese 29, 2158
- \chinese 24, 81, 2157, 2158, 2165, 3027, 3084
- \CJK 1865
- \CJKbold 3673, 3674, 3684, 3685, 3699, 3700, 3702, 3703, 3705, 3706, 3710, 3711, 3713, 3714, 3716, 3717, 3729, 3730, 3732, 3733, 3735, 3736, 3740, 3741, 3743, 3744, 3746, 3747
- \CJKfamily 57, 524, 528, 532, 536, 1055, 1077, 1257, 1261, 1265, 1269, 1417, 1649, 4139, 4140, 4141, 4142, 4144, 4145, 4147
- \CJKfamilydefault 518, 519, 536, 592, 1251, 1252, 1269, 1277, 1279, 1283, 1290, 1303, 1645, 1654, 1719, 1723, 1731, 1744, 1746
- \CJKfilltwosides 1873
- \CJKfontspec 59, 1213
- \CJKglue 1899, 1913, 1930, 1943
- \CJKhook 46, 607
- \CJKnormal 3666, 3667, 3668, 3669, 3670, 3671, 3672, 3677, 3678, 3679, 3680, 3681, 3682, 3683, 3698, 3701, 3704, 3709, 3712, 3715, 3728, 3731, 3734, 3739, 3742, 3745
- \CJKpunctmapfamily 640, 641, 642, 643, 644, 645, 649, 650, 651, 652, 653, 654
- \CJKrmdefault 512, 519, 520, 524, 1125, 1136, 1226, 1252, 1253, 1257, 1279, 1282, 1283, 1642, 1645, 1646, 1727, 3795, 3797, 3798, 3803, 3804, 3805, 3808, 3885, 3888, 3889, 3894, 3895, 3896, 3963, 3966, 3967, 3973, 3974, 3975, 4040, 4043, 4044, 4051, 4052, 4053, 4101, 4104, 4105, 4110, 4111, 4112
- \CJKsfdefault 513, 521, 528, 1126, 1137, 1232, 1254, 1261, 1643, 1647, 1728, 3779, 3785, 3786, 3809, 3886, 3964, 3968, 4041, 4045, 4102
- \CJKtilde 593
- \CJKttdefault 514, 522, 532, 1127, 1138, 1237, 1255, 1265, 1644, 1648, 1729, 3796, 3810, 3887, 3965, 4042, 4103
- clearalternatefont 26, 65
- \cleardoublepage 2378, 2492
- \clearpage 2380, 2492
- clist commands:
 - \clist_clear:N 1002, 4488
 - \clist_concat:NNN 1024
 - \clist_const:Nn 620
 - \clist_gput_right:Nn 380, 402, 404
 - \clist_gput_right:No 392
 - \clist_gset:Nn 1243
 - \clist_if_empty:NT 1336, 1595
 - \clist_if_empty:NTF 1355
 - \clist_map_break:n 2757, 3251
 - \clist_map_function:nN 1546, 1560
 - \clist_map_function:xN 1512, 1514
 - \clist_map_inline:Nn 610, 1448
 - \clist_map_inline:nn 365, 632, 1553, 1857, 1883, 2752, 3248, 3390, 3425, 4477
 - \clist_new:N 136, 1021, 1244, 1390, 4506, 4646
 - \clist_put_left:Nn 3317, 3349
 - \clist_put_left:Nx 1025
 - \clist_put_right:Nn 1187
 - \clist_put_right:Nx 1436, 4500
 - \clist_set:Nn 1373, 1374
 - \clist_set:No 1006
 - \clist_set:Nx 4647
 - \clist_use:Nn 4656
- \clubpenalty 2678, 2685
- colon commands:
 - \c_colon_str 939, 940, 941
- \contentsname 2196
- contentsname 11
- continuation 12
- cs commands:
 - \cs:w 2998
 - \cs_end: 2998
 - \cs_generate_variant:Nn 91, 465, 477, 762, 966, 1089, 1103, 1176, 1196, 1197, 1369, 1406, 1508, 1546, 1554, 4688, 4694, 4812, 4919
 - \cs_gset_eq:cc 1163, 1166
 - \cs_gset_eq:NN 159, 160, 161, 259, 264, 269, 307, 309, 375, 379, 497, 809, 813, 3354
 - \cs_gset_nopar:Npx 810
 - \cs_gset_protected_nopar:cpx 1051, 2706
 - \cs_gset_protected_nopar:Npn 750, 2360, 2728
 - \cs_gset_protected_nopar:Npx 848, 1410, 1504
 - \cs_if_eq:cNF 950
 - \cs_if_eq:NNT 4697
 - \cs_if_eq:NNTF 4846, 4925
 - \cs_if_exist:cF 804, 846, 884
 - \cs_if_exist:cTF 2985
 - \cs_if_exist:NF 356, 474, 816, 1409, 1501, 1964, 3268, 4702
 - \cs_if_exist:NTF 3266
 - \cs_if_exist_p:c 4757, 4758
 - \cs_if_exist_use:c 1073, 1074
 - \cs_if_exist_use:cF 4, 787
 - \cs_if_exist_use:cT 1523, 4691

- \cs_if_exist_use:cTF 1095
- \cs_if_free:cF 459, 612, 627, 1160, 2771, 2837, 2842
- \cs_if_free:cTF 1312
- \cs_if_free:Nf 469, 2051, 2162, 2846
- \cs_if_free:Ntf 2037
- \cs_new:Npn
 - . 108, 115, 122, 905, 911, 915, 998, 1076, 2796, 4802, 4920
- \cs_new_eq:cc 1892
- \cs_new_eq:cN 6, 7, 8, 9, 20, 23, 1878
- \cs_new_eq:Nc 1955
- \cs_new_eq:NN 93, 104, 105, 106, 131, 132,
 - 150, 357, 466, 597, 658, 825, 850, 926, 957, 1223, 1229,
 - 1740, 1879, 1880, 1943, 1954, 2157, 2158, 2159, 2698,
 - 2699, 2700, 2701, 2727, 3003, 3264, 3269, 4482, 4850, 4921
- \cs_new_nopar:Npn 865, 1068, 1121, 1132, 2835, 2840, 4507
- \cs_new_protected:Npn 60,
 - 743, 748, 999, 1018, 1334, 1425, 1439, 1590, 1949,
 - 2336, 2848, 3445, 3451, 4483, 4661, 4663, 4677, 4683,
 - 4703, 4708, 4713, 4724, 4728, 4733, 4738, 4748, 4753,
 - 4774, 4825, 4841, 4851, 4863, 4872, 4891, 4896, 4907, 4922
- \cs_new_protected_nopar:cpx 1862
- \cs_new_protected_nopar:Npn 75, 77, 133, 134,
 - 445, 457, 467, 472, 478, 510, 540, 557, 599, 605, 608,
 - 625, 630, 638, 647, 656, 727, 785, 794, 807, 812, 814,
 - 826, 840, 867, 882, 895, 919, 928, 945, 959, 976, 1022,
 - 1030, 1049, 1069, 1079, 1104, 1142, 1154, 1156, 1178,
 - 1180, 1273, 1298, 1307, 1353, 1364, 1391, 1400, 1407,
 - 1413, 1433, 1446, 1465, 1479, 1486, 1498, 1518, 1534,
 - 1552, 1555, 1557, 1570, 1582, 1616, 1635, 1658, 1671,
 - 1685, 1717, 1749, 1767, 1783, 1794, 1798, 1837, 1844,
 - 1848, 1869, 1875, 1896, 1910, 1945, 2036, 2038, 2063,
 - 2098, 2108, 2113, 2260, 2290, 2358, 2691, 2744, 2764,
 - 2769, 2774, 2784, 2794, 2813, 2981, 2990, 2995, 3363,
 - 3378, 3410, 3415, 3565, 3569, 4474, 4498, 4665, 4667, 4689
- \cs_new_protected_nopar:Npx 482, 1209
- \cs_set:Npn 1332
- \cs_set_eq:cc 614, 628
- \cs_set_eq:cN 1960
- \cs_set_eq:Nc 2693, 2694, 2695, 2696
- \cs_set_eq:NN 107, 187, 192, 197,
 - 537, 555, 575, 602, 606, 765, 768, 819, 820, 927, 958,
 - 996, 1055, 1270, 1409, 1417, 1427, 1428, 1429, 1501,
 - 1502, 1503, 1638, 1722, 2172, 2177, 2180, 2185, 2748, 2758
- \cs_set_nopar:cpx 1893
- \cs_set_nopar:Npn 1693
- \cs_set_nopar:Npx 2786, 4669
- \cs_set_protected:Npn 730, 4751, 4817
- \cs_set_protected_nopar:Npn 601, 769, 1913, 3452
- \cs_set_protected_nopar:Npx 1951
- \cs_to_str:N 54, 1208, 4749
- \cs_undefine:c 1034, 1035, 1039, 1040, 4692
- \cs_undefine:N 1430
- cs4size 28
- \csname 2593, 2594,
 - 2599, 2615, 2616, 2618, 2621, 2622, 2623, 2632, 2633,
 - 2634, 2637, 2638, 2639, 4183, 4185, 4186, 4187, 4192, 4196
- \CTeX 24, 108, 3546, 3549
- ctex commands:
 - \ctex_add_cmap:cn 470
 - \ctex_add_cmap:n 42, 462, 467
 - \ctex_add_cmap:Nn 472, 477
 - \ctex_add_to_selectfont:n 76, 1949, 1970, 2050
 - \ctex_after_end_preamble:n 34, 107, 128, 4663
 - \g__ctex_after_end_preamble_hook_tl 4664, 4668, 4676
 - \ctex_appto_cmd:NnnTF 130, 493, 1629, 4733
 - \ctex_assign_heading_name:nn ... 84, 2296, 2333, 2720
 - __ctex_assign_heading_name:nnn 84, 2335, 2336
 - \ctex_at_end:n 35, 133, 134, 291, 354, 1943, 3632
 - \ctex_at_end_package:nn
 - 129, 129, 448, 496, 763, 774, 1808, 1810, 1828,
 - 2160, 2359, 2804, 2845, 2847, 2869, 2899, 3009, 3548, 4677
 - \ctex_at_end_preamble:n . 128, 359, 582, 589, 1272, 4661
 - \ctex_auto_ignorespaces: 45, 597, 606, 1987
 - \l__ctex_autoindent_tl
 - 35, 130, 186, 191, 196, 2024, 2028, 2031, 2100, 2103
 - __ctex_calc_bounds:nn 4502, 4503, 4507
 - __ctex_ccglue: 1930, 1943
 - \l__ctex_ccglue_skip 1914, 1918,
 - 1921, 1924, 1926, 1934, 1938, 2074, 2085, 2094, 2130, 2138
 - \ctex_CJK_input:n 44, 540, 555
 - \c__ctex_class_tl
 - .. 82, 84, 407, 412, 417, 422, 3264, 3270, 3272, 3273, 3275
 - __ctex_clear_fntef_color:n ... 1837, 1844, 1848, 1861
 - \c__ctex_cmap_encoding_seq 42, 454, 455, 461
 - \l__ctex_ctexcap_options_clist 4646, 4647, 4656
 - \ctex_declare_math_sizes:nnnn 105, 3410, 3444
 - __ctex_def_heading_keys:n 83, 2290, 2353
 - \ctex_default_pt:n 34, 111, 115
 - __ctex_default_pt:w 117, 122
 - \ctex_detected_platform: .. 71, 1749, 1767, 3573, 3582
 - \l__ctex_encoding_tl 35, 126, 127, 219, 220,
 - 503, 506, 584, 592, 1804, 2013, 2155, 3306, 3320, 3327, 3333
 - \g__ctex_end_preamble_hook_tl 4662, 4666, 4675
 - \c__ctex_engine_file_str 32, 48, 2017
 - \ctex_family_cmap:nn 42, 457, 465
 - \ctex_family_cmap:xn 466
 - \l__ctex_family_default_init_tl
 - 71, 1719, 1739, 1741, 1746
 - __ctex_family_default_wrap:n 1722, 1740, 1743
 - \ctex_file_input:n 33, 75, 395, 659, 2017,
 - 3275, 3310, 3315, 3321, 3322, 3328, 3329, 3335, 3336,
 - 3342, 3347, 3454, 3456, 3574, 3578, 3586, 3632, 3757, 3758
 - \ctex_file_wrapper:nnn 33, 60, 76, 79, 542
 - __ctex_fix_varioref_label:n 98
 - \ctex_fix_varioref_label:n 2993, 2995
 - \g__ctex_font_size_flag
 - .. 36, 150, 159, 160, 161, 375, 379, 401, 3353, 3354, 3453
 - \c__ctex_font_size_prop 105, 3365, 3376, 3382, 3420, 3447
 - \c__ctex_font_size_seq 3374, 3377, 3388
 - \l__ctex_font_size_tl 3365, 3366, 3412, 3413, 3447, 3448
 - \ctex_fontset_error:n
 - 109, 3565, 3864, 3898, 3932, 3977, 4009, 4012, 4055, 4114

- \g_ctex_fontset_tl ... 223, 1751, 1779, 1786, 1787, 3571, 3574, 3577, 3578, 3584, 3586, 3606, 3608, 3613, 3630
- \ctex_gadd_hook:cn 129, 4681
- \ctex_gadd_hook:Nn 129, 607, 4683, 4688
- _ctex_get_font_sizes:Nn 105, 3412, 3415
- \ctex_get_macro_meaning:N 4820
- \ctex_get_macro_meaning:NTF 132, 4843, 4899
- _ctex_get_macro_meaning:w 132, 4822, 4825
- \l_ctex_heading_bool 302, 3260
- \c_ctex_headings_seq 83, 2250, 2252, 2253, 2254, 2257, 2350, 2992
- _ctex_hookto_cmd:Nnnnw 134, 4730, 4735, 4896
- _ctex_hookto_cmd_parameter:Nnnnw .. 134, 4903, 4922
- _ctex_hookto_cmd_parameterless:Nnnw 134, 4902, 4907
- \ctex_hypersetup:n 72, 72, 357, 1794, 1798, 1803, 1806, 1821, 1824
- \ctex_if_autoindent_touched:F 35, 131, 187, 192, 197, 3357
- \ctex_if_ccglue_touched: 1927
- \ctex_if_ccglue_touched:TF 75, 2069
- \ctex_if_ccglue_touched_p: 75
- \ctex_if_macosx:TF 72, 1774, 1775, 1777, 1783
- \ctex_if_preamble:TF 34, 106, 107, 3601
- \ctex_if_rescanable:NnTF 132, 4841, 4875
- \ctex_ignorespaces_case:N .. 46, 599, 1984, 1987, 1990
- _ctex_initial_heading:n 83, 2260, 2352
- _ctex_left_hook_aux:nn 4920
- \l_ctex_line_spread_fp ... 107, 179, 3355, 3539, 3541
- \l_ctex_line_stretch_tl 2056, 2061, 2062, 2066
- \ctex_load_fontset: 109, 3569, 3590, 3609, 3631
- \ctex_load_zhmap:nnnn 43, 510, 517, 3801, 3892, 3971, 4049, 4108
- \l_ctex_ltj_add_alternate_bool 1158, 1170, 1188, 1195
- \ctex_ltj_add_font_features:n 59, 1178, 1196
- \ctex_ltj_add_font_features:nn 59, 1180, 1197
- \ctex_ltj_add_font_features:x 1220
- \ctex_ltj_add_font_features:xn 1179
- _ctex_ltj_alternate_cs:n 1035, 1038, 1039, 1040, 1073, 1074, 1076, 1161, 1164, 1165, 1167, 1168, 1359, 1396, 1458, 1483, 1523, 1540
- \l_ctex_ltj_alternate_family_tl 1421, 1424
- \l_ctex_ltj_alternate_options_clist 1374, 1375, 1386, 1390
- \l_ctex_ltj_alternate_seq 1003, 1343, 1344, 1352, 1358, 1366, 1614
- \l_ctex_ltj_base_CJKfamily_tl 1004, 1020, 1071, 1458, 1483
- \l_ctex_ltj_base_family_tl 65, 1036, 1066, 1072, 1402, 1443, 1502, 1521, 1526, 1527, 1539
- _ctex_ltj_change_encoding: ... 55, 976, 1054, 1416
- \l_ctex_ltj_char_range_clist 1002, 1327, 1336, 1355, 1373, 1385, 1397, 1420, 1442, 1593, 1595
- \ctex_ltj_char_range_key:nn 67, 1587, 1590
- _ctex_ltj_char_range_parse_feature:w 67, 1598, 1602
- \g_ctex_ltj_char_range_prop ... 66, 1450, 1561, 1566
- \l_ctex_ltj_char_range_tl 1450, 1453, 1559, 1561, 1563, 1565, 1572
- _ctex_ltj_check_family:n 56, 1030, 1357
- \ctex_ltj_clear_alternate_font:n 1512, 1518
- \l_ctex_ltj_current_font_tl 50, 787, 791, 792, 804, 846, 851
- \l_ctex_ltj_current_shape_tl ... 869, 874, 879, 880
- \ctex_ltj_declare_alternate_shape:nnnnn ... 64, 1419, 1439, 1441
- \ctex_ltj_declare_char_range:n 1552, 1554
- _ctex_ltj_declare_char_range:nn 1553, 1555
- \ctex_ltj_declare_char_range:nn ... 66, 1556, 1557
- \ctex_ltj_declare_char_range:x 1549
- \ctex_ltj_def_char_range_key:n 67, 1562, 1582
- \g_ctex_ltj_default_features_clist 1024, 1243, 1244
- \ctex_ltj_ensure_default_family: ... 60, 1273, 1736
- \ctex_ltj_extract_font: 51, 819, 826
- _ctex_ltj_family_csname:n ... 1034, 1051, 1068, 1095
- \g_ctex_ltj_family_font_name_prop 56, 1009, 1028, 1032, 1106, 1182, 1275, 1287
- \g_ctex_ltj_family_font_options_prop 56, 1010, 1029, 1185
- \ctex_ltj_family_if_exist:nTF 57
- \ctex_ltj_family_if_exist:xN 1090
- \ctex_ltj_family_if_exist:xNF 1277
- \ctex_ltj_family_if_exist:xNT 1303
- \ctex_ltj_family_if_exist:xNTF 1081, 1282, 1300, 1521
- \g_ctex_ltj_family_int . 1148, 1150, 1435, 1436, 1438
- \g_ctex_ltj_family_name_prop 56, 1027, 1036, 1058, 1092, 1402
- \l_ctex_ltj_family_tl 1083, 1088, 1145, 1146, 1161, 1165, 1168, 1179, 1515, 1516
- _ctex_ltj_family_unknown_warning:n 57, 1086, 1104, 1531
- \l_ctex_ltj_font_options_clist ... 1006, 1011, 1012, 1013, 1021, 1057, 1186, 1187, 1190, 1393, 1394, 1398
- \ctex_ltj_fontspec:nn 58, 1142, 1155
- _ctex_ltj_fontspec:nnn 1156, 1176
- _ctex_ltj_fontspec:xnn 1149
- \ctex_ltj_fontspec:xx 1154, 1189, 1215
- \g_ctex_ltj_fontspec_family_tl 1056, 1059, 1060, 1061, 1067, 1072, 1097
- \g_ctex_ltj_fontspec_prop 1144, 1172, 1177
- \ctex_ltj_get_and_define_fonts:nN ... 54, 919, 927
- \ctex_ltj_get_and_define_fonts_al:nN 923, 926
- \ctex_ltj_get_and_define_fonts_ja:nN 922, 928
- _ctex_ltj_gset_family_cs:nn 56, 1014, 1049
- \ctex_ltj_if_alternate_shape_exist:n 860
- \ctex_ltj_if_alternate_shape_exist:nF 842
- \ctex_ltj_if_alternate_shape_exist:nT ... 829, 869
- \ctex_ltj_if_alternate_shape_exist:nTF 52
- \ctex_ltj_if_jfont:nTF 53, 55, 905, 916, 921
- \ctex_ltj_if_jfont_math:NTF 53, 911, 952
- _ctex_ltj_if_jfont_math:w 912, 915
- \l_ctex_ltj_jfm_tl ... 54, 962, 963, 965, 980, 982, 1025
- \c_ctex_ltj_math_fam_int 988, 989

- \c__ctex_ltj_math_family_tl 1309, 1311, 1313, 1316, 1320
- \ctex_ltj_math_group_hook: 54, 769, 951, 957
- \c__ctex_ltj_math_tl 983, 984, 986, 988, 1241, 1300, 1310, 1315, 1319
- __ctex_ltj_msg_def_family_map:n 1119, 1121
- __ctex_ltj_msg_family_map:n 1048, 1118, 1132
- __ctex_ltj_noembed_wrap:n 996, 998
- __ctex_ltj_patch_external_font:w 52, 832, 865
- \ctex_ltj_pickup_font: 50, 800, 814, 825, 889, 931, 933, 935
- __ctex_ltj_pop_fontname: 803, 812, 891
- __ctex_ltj_push_fontname:n 799, 807, 888
- \ctex_ltj_reset_alternate_font:n 1514, 1534
- \g__ctex_ltj_reset_alternate_prop 1041, 1525, 1537, 1545
- \ctex_ltj_save_alternate_family:Nnnn 63, 1384, 1404, 1407
- \ctex_ltj_save_alternate_seq:cn 1359
- \ctex_ltj_save_alternate_seq:Nn 62, 1364, 1369
- \ctex_ltj_save_alternate_seq:Nnnwn 62
- \ctex_ltj_save_alternate_seq:Nnnwnw 1367, 1370
- __ctex_ltj_save_alternate_shape:cn 1457, 1482
- __ctex_ltj_save_alternate_shape:Nnn 65, 1498, 1508
- \ctex_ltj_save_char_range:n 66, 1560, 1567
- \ctex_ltj_save_char_range:nn 1569, 1570
- __ctex_ltj_save_fontname:w 809, 813
- \ctex_ltj_select_alternate_font: 52, 867, 1973
- \ctex_ltj_select_font: 50, 785, 1972
- __ctex_ltj_select_font_aux: 50, 788, 794, 805
- \ctex_ltj_set_alternate_family:coonn 1395
- __ctex_ltj_set_alternate_family:n 1062, 1069
- \ctex_ltj_set_alternate_family:nn .. 63, 1362, 1391
- \ctex_ltj_set_alternate_family:nnF .. 62, 1007, 1353
- \ctex_ltj_set_alternate_family:nnn 63, 1403, 1411, 1413
- \ctex_ltj_set_alternate_family:Nnnnn 1400, 1406
- \ctex_ltj_set_alternate_seq:n 61, 1325, 1334
- \ctex_ltj_set_alternate_shape:n ... 1467, 1479, 1490
- \ctex_ltj_set_alternate_shape:nnN .. 65, 1452, 1486
- \ctex_ltj_set_alternate_shape:nnn .. 64, 1455, 1462
- \ctex_ltj_set_alternate_shape:nnnn 1464, 1465
- \ctex_ltj_set_alternate_shape:Nnnnnnn 64, 1442, 1446
- \ctex_ltj_set_family:nnn 55, 999, 1019, 1173
- \ctex_ltj_set_family:xxx 1018, 1204, 1211, 1226, 1232, 1237, 1241
- \ctex_ltj_subst_font: 51, 820, 840
- \ctex_ltj_swap_cs:NN 63, 1418, 1425
- \ctex_ltj_switch_family:n 1079, 1089, 1174, 1210
- \ctex_ltj_switch_family:x 1078, 1146
- __ctex_ltj_tmp:w 1332, 1349, 1427, 1429, 1430
- \l__ctex_ltj_tmp_seq 1340, 1341, 1344, 1351
- \l__ctex_ltj_tmp_tl .. 1005, 1006, 1032, 1043, 1046, 1183, 1191, 1207, 1210, 1211, 1300, 1301, 1303, 1304, 1338, 1339, 1340, 1377, 1379, 1380, 1387, 1597, 1605, 1614
- \l__ctex_ltj_tmpa_tl 1277, 1282
- __ctex_ltj_um_char:Nn 738, 743
- __ctex_ltj_um_char_aux:Nn 748, 762
- __ctex_ltj_um_char_aux:Nx 745
- __ctex_ltj_um_define_math_chars: 727, 765
- \g__ctex_ltj_unknown_family_seq ... 1108, 1110, 1115
- __ctex_ltj_update_family_uid:N 64, 1012, 1393, 1433
- \ctex_ltj_update_mathfont: 61, 1295, 1298
- \ctex_ltj_update_mathfont:n 1301, 1304, 1307
- \ctex_ltj_update_xkanjiskip: 68, 1616, 1974
- __ctex_ltj_use_global_options:N 1013, 1022, 1375, 1394
- \ctex_ltj_use_math_group:Nn 54, 768, 945, 958
- \l__ctex_ltj_xkanjiskip_skip 68, 1619, 1621, 1622, 1625, 1626
- \c__ctex_macosx_file_str 1785, 1789
- \ctex_make_spa:nn 125, 4474, 4482
- \ctex_mono_jfm:n 54, 959, 966
- \ctex_mono_jfm:o 967, 2007
- \l__ctex_msyh_suffix 3761, 3762, 3766, 3782, 3784, 3818
- \ctex_package_end_hook:n 129, 4689, 4694
- \ctex_package_end_hook:o 129, 4698
- \l__ctex_parameter_str ... 132, 4814, 4833, 4854, 4901
- \l__ctex_parameter_tl 4854, 4859, 4870
- \ctex_parse_name:NN 131, 4744, 4748
- \ctex_parse_name:NNn 4753, 4812
- __ctex_parse_name:nNNnN 4765, 4774
- \ctex_parse_name:NNx 4749
- __ctex_parse_name:w 4787, 4793, 4802
- \ctex_patch_boot:NNnnTF 130, 4705, 4710, 4715, 4730, 4735, 4738
- \ctex_patch_cmd:Nnn 130, 1655, 1695, 2815, 2821, 2827, 2880, 2884, 2890, 2894, 2902, 2905, 2910, 2914, 3560, 3561, 3562, 4713
- __ctex_patch_cmd:Nnnnnw .. 133, 4705, 4710, 4715, 4872
- \ctex_patch_cmd_all:NnnnTF 130, 4708
- \ctex_patch_cmd_once:NnnnTF 129, 4703
- \ctex_patch_failure:N 499, 526, 530, 534, 538, 1259, 1263, 1267, 1271, 1633, 1962, 2865, 2875, 4722, 4724
- __ctex_patch_false:w 4741, 4747, 4887, 4889, 4905, 4931
- __ctex_patch_rebuild:Nn 133, 4845, 4851, 4884, 4924, 4928
- __ctex_patch_replace:nnnTF 133, 4879, 4891
- __ctex_patch_rescan:NNn 4853, 4854, 4855, 4863
- __ctex_patch_true:w 4740, 4746, 4885, 4917, 4929
- \ctex_plane_to_utfxvibe:Nn 44, 557, 575
- \l__ctex_prefix_str .. 132, 4813, 4832, 4853, 4909, 4910
- \l__ctex_prefix_tl 4853, 4858, 4869
- \ctex_preto_cmd:NnnTF 130, 523, 527, 531, 535, 1256, 1260, 1264, 1268, 1956, 2862, 2871, 4728
- \l__ctex_punct_bounds_clist ... 4488, 4494, 4500, 4506
- \c__ctex_punct_family_clist 610, 620
- \l__ctex_punct_font 4486, 4487
- \ctex_punct_map_bfseries:nn 46, 630, 3786, 3798, 3804, 3889, 3895, 3967, 3968, 3974, 4044, 4045, 4046, 4052, 4105, 4111

- \ctex_punct_map_family:nn ... 46, 625, 3779, 3785,
3795, 3796, 3803, 3885, 3886, 3887, 3894, 3963, 3964,
3965, 3973, 4040, 4041, 4042, 4051, 4101, 4102, 4103, 4110
- \ctex_punct_map_itshape:nn 47, 647,
3797, 3805, 3888, 3896, 3966, 3975, 4043, 4053, 4104, 4112
- \ctex_punct_map_series:nnn 634, 635, 638
- \c__ctex_punct_seq 4465, 4466, 4489
- \ctex_punct_set:n 46, 608,
3774, 3802, 3884, 3893, 3962, 3972, 4039, 4050, 4100, 4109
- \ctex_punct_space:nn 47, 656, 658
- \l__ctex_punct_tl
..... 278, 579, 580, 667, 967, 2004, 2005, 2006, 2007
- __ctex_rebuild_cmd:w ... 4845, 4846, 4850, 4924, 4925
- \l__ctex_replacement_str
..... 132, 4815, 4834, 4855, 4893, 4894, 4927
- \l__ctex_replacement_tl . 4828, 4829, 4855, 4860, 4871
- __ctex_right_hook_aux:nn 4921
- __ctex_save_bounds:n 4490, 4498
- __ctex_save_cmap:Nn 474, 478
- __ctex_save_font_size:nn 105, 3378, 3409
- __ctex_save_selectfont: 1955, 1960
- \ctex_scheme_input:n 33, 77, 91
- \ctex_scheme_input:o 3011, 3277
- \l__ctex_scheme_tl 313, 3011, 3277
- \g__ctex_section_depth_flag
..... 33, 93, 307, 309, 3199, 3215, 3231
- \c__ctex_section_headings_seq
..... 83, 2246, 2247, 2252, 2703, 2762
- \ctex_select_size:.. 2025, 2032, 2035, 2036, 2057, 2111
- \ctex_set_default_ccwd:Nn ... 34, 108, 196, 2031, 2056
- \ctex_set_font_size:Nnn 106, 3445, 3460,
3468, 3479, 3490, 3491, 3492, 3493, 3494, 3495, 3496,
3499, 3507, 3518, 3529, 3530, 3531, 3532, 3533, 3534, 3535
- __ctex_set_font_size:nnNn 3448, 3451
- \ctex_set_ignorespaces: 46, 601, 603, 605, 607
- \ctex_set_upfamily:nnn
..... 69, 1658, 1670, 3821, 3822, 3823, 3824, 3825,
3826, 3828, 3908, 3909, 3910, 3911, 3987, 3988, 3989,
3990, 4064, 4065, 4066, 4067, 4068, 4070, 4123, 4124, 4125
- \ctex_set_upfonts:nnnnnn
..... 70, 1685, 1692, 3816, 3904, 3983, 4060, 4119
- \ctex_set_upmap:nnn 69,
1671, 1684, 1687, 1688, 1689, 1690, 3827, 3829, 4069, 4071
- \ctex_set_zhmap:n 42, 445, 451, 515, 1673
- \g__ctex_spa_iow 4476, 4479, 4481, 4491
- \c__ctex_std_class_tl 102, 3251, 3263, 3264, 3302, 3304
- \g__ctex_std_options_clist
..... 35, 136, 380, 392, 402, 404, 408, 413, 418, 423
- \ctex_titleps_hook: 94, 2813, 2846, 2847
- __ctex_titlesec_format:n 2758, 2766, 2769
- __ctex_titlesec_format:Nn 2748, 2772, 2774
- __ctex_titlesec_format_explicit:Nn ... 2749, 2784
- \ctex_titlesec_hook: 92, 2744, 2808
- __ctex_titlesec_hook:n 2758, 2762, 2764
- __ctex_titlesec_spacing:Nn 2767, 2794
- __ctex_titlesec_spacing:nnnnnn 2795, 2796
- __ctex_tmp:w 4751, 4806, 4817, 4839
- \l__ctex_tmp_box 33, 59, 1899, 1900, 2850, 2854
- \l__ctex_tmp_dim 33, 58,
2066, 2067, 2072, 2075, 2081, 2120, 2122, 2124, 2133, 2135
- \l__ctex_tmp_int 33, 57, 560,
562, 563, 565, 569, 571, 2080, 2082, 2083, 2089, 2090,
2122, 2123, 2124, 2127, 2128, 2133, 2135, 2702, 2705, 2709
- \l__ctex_tmp_tl 33, 56, 364, 371, 386,
387, 480, 481, 559, 561, 2292, 2344, 2355, 2356, 3420, 3421
- \ctex_update_ccglue: 75, 1910, 2076, 2095, 2139
- \ctex_update_ccwd: 75, 1896, 2070, 2147
- \ctex_update_default_family:.. 45, 70, 582, 1272, 1717
- \ctex_update_em_unit: 76, 1945, 2065, 2115
- \ctex_update_parindent: 79, 2043, 2098, 2140
- \ctex_update_size: 78, 2037, 2038, 2051
- \ctex_update_stretch: 79, 2042, 2063
- \ctex_update_ziju: 80, 2045, 2113
- \c__ctex_upper_case_file_str 1773, 1781
- \ctex_varioref_hook: 97, 2990, 3001, 3009
- \c__ctex_version_tl 3, 6, 7, 8, 9, 20, 23
- __ctex_write_family:nn 4478, 4483
- \c__ctex_zero_tl 2040, 2047, 2049
- \ctex_zhmap_case:nnn
..... 35, 132, 259, 264, 269, 3772, 3866, 3934, 4027, 4091
- \g__ctex_zhmCJK_bool ... 35, 129, 258, 263, 268, 502, 583
- \ctex_zihao:n 104, 3362, 3363
- \ctex_ziju:n 2107, 2108
- \l__ctex_ziju_dim 2116, 2117, 2121, 2132, 2134, 2138, 2142
- \l__ctex_ziju_tl 2040, 2048, 2049, 2110, 2116
- \ctexbibname 2218, 2238
- \CTEXcounter 2159
- \CTEXdigits 24, 81, 2170
- \CTEXfilltwosides 29, 1869, 1879
- \CTEXindent 29, 80, 2143, 2145
- \CTEXnoindent 29, 80, 2149, 2151
- \CTEXnumber 24, 81, 2168
- \CTEXoptions 29, 41, 437, 439
- \ctexset 5, 41, 428, 434, 440
- \CTEXsetfont 29, 78, 2035
- \CTEXsetup 29, 41, 431, 433
- \CTEXsout 29
- \ctexspadef 47, 658, 4493
- \CTEXthechapter 16, 2509, 2512, 2892, 2912, 2913
- \CTEXtheparagraph 16
- \CTEXthepart 16, 2400, 2430
- \CTEXthesection 16, 2882, 2896, 2904, 2916
- \CTEXthesubparagraph 16
- \CTEXthesubsection 16, 2886, 2907
- \CTEXthesubsubsection 16
- \CTEXunderdblline 29
- \CTEXunderdot 29
- \CTEXunderline 29
- \CTEXunderwave 29
- \CTEXxout 29
- \CurrentOption 392

D

- declarecharrange 25, 66
- \DeclareFontEncoding 969
- \DeclareFontFamily
 - ... 43, 493, 499, 978, 3650, 3651, 3652, 3653, 3654, 3655
- \DeclareFontShape
 - 979, 981, 1662, 1663, 1666, 1667, 3666, 3667,
 - 3668, 3669, 3670, 3671, 3672, 3673, 3674, 3677, 3678,
 - 3679, 3680, 3681, 3682, 3683, 3684, 3685, 3689, 3690,
 - 3691, 3692, 3693, 3694, 3698, 3699, 3700, 3701, 3702,
 - 3703, 3704, 3705, 3706, 3709, 3710, 3711, 3712, 3713,
 - 3714, 3715, 3716, 3717, 3721, 3722, 3723, 3724, 3728,
 - 3729, 3730, 3731, 3732, 3733, 3734, 3735, 3736, 3739,
 - 3740, 3741, 3742, 3743, 3744, 3745, 3746, 3747, 3751, 3752
- \DeclareFontSubstitution 972
- \DeclareKanjiFamily
 - 1660, 1661, 3657, 3658, 3659, 3660, 3661, 3662
- \DeclareMathSizes 3413
- \DeclareSymbolFont 984, 1310
- \def 2395, 2425, 2452,
 - 2466, 2476, 2503, 2532, 2554, 2569, 2592, 2595, 2628,
 - 2645, 2658, 2659, 2663, 3473, 3484, 3512, 3523, 4157, 4193
- \defaultCJKfontfeatures 59, 1242, 1250
- \defbeamertemplate 2922, 2940, 2958
- \defbeamertemplatealias 2976, 2977, 2978
- dim commands:
 - \dim_abs:n 2134
 - \dim_add:Nn 2117
 - \dim_compare:nNnF 2102, 2125
 - \dim_compare:nNnTF 110, 2067, 2118
 - \dim_eval:n 2087, 2133
 - \dim_max:nn 2853
 - \dim_min:nn 2134
 - \dim_new:N 58, 1909, 2142
 - \dim_set:Nn 1900, 1946,
 - 1947, 1948, 2066, 2072, 2103, 2116, 2120, 2147, 2851, 3543
 - \dim_sub:Nn 2124
 - \dim_to_decimal:n 34, 3384, 3385, 3422
 - \dim_to_decimal_in_unit:nn 4513
 - \dim_use:N 118, 3543
 - \dim_zero:N 2153
- \dimexpr 2410, 2455, 2536, 2558, 2928, 2946, 2964
- \directlua 705, 706
- \document 4669, 4672
- \endCJKfilltwosides 1875
- \endcsname 2593, 2594, 2599,
 - 2615, 2616, 2618, 2621, 2622, 2623, 2632, 2633, 2634,
 - 2637, 2638, 2639, 4183, 4185, 4186, 4187, 4190, 4192, 4196
- \endCTEXfilltwosides 1875, 1880
- \endgroup 2617, 2656, 2679, 2938, 2956, 2974, 4176, 4184, 4199
- \endinput 4181
- \endlinechar 4155, 4156
- etex commands:
 - \etex_dimexpr:D 118, 2081
 - \etex_glueexpr:D 2073
- \everypar 2584, 2674, 2686
- \EverySelectfont 1705, 1964, 1967
- exp commands:
 - \exp_after:wN
 - 117, 816, 832, 834, 909, 912, 2795, 3366, 3413, 3448, 4822
 - \exp_args:cc 2986
 - \exp_args:Nc 2767, 2772, 4776
 - \exp_args:Nnc 98
 - \exp_args:NNf 1605
 - \exp_args:NNNo 3542
 - \exp_args:Nno 2998
 - \exp_args:No 561, 1782, 2791, 4490, 4839
 - \exp_args:Nx 291, 589, 663, 2107, 3362, 3541
 - \exp_last_unbraced:Nf 4787
 - \exp_not:c 851, 1864,
 - 1866, 1893, 2266, 2267, 2268, 2273, 2275, 2277, 2278,
 - 2281, 2282, 2285, 2297, 2298, 2299, 2300, 2301, 2302,
 - 2303, 2304, 2305, 2306, 2307, 2309, 2310, 2313, 2315,
 - 2317, 2319, 2321, 2323, 2710, 2711, 2712, 2713, 4768, 4769
 - \exp_not:N 122, 123, 484,
 - 591, 592, 593, 778, 779, 781, 887, 972, 1058, 1059, 1060,
 - 1061, 1384, 1505, 1727, 1728, 1729, 1743, 1865, 2276,
 - 2708, 2713, 2779, 2780, 2789, 2790, 3382, 3584, 4767, 4858
 - \exp_not:n 67, 1055, 1056, 1722, 2798, 4785, 4915
 - \exp_not:o 386,
 - 780, 782, 1057, 1190, 1191, 1385, 1386, 1387, 1411,
 - 1505, 1744, 1952, 2355, 2987, 4672, 4858, 4859, 4860, 4914
 - \exp_not:v 4648
- \expandafter
 - 4158, 4159, 4183, 4184, 4185, 4186, 4187, 4192, 4196, 4197
- \ExplSyntaxOff 64, 70, 493, 523, 527, 531, 535,
 - 687, 1256, 1260, 1264, 1268, 1956, 2862, 2921, 4519, 4718
- \ExplSyntaxOn 69, 716, 2979, 4458

E

- \edef 4176, 4177
- \else 1656, 1657, 2370, 2379,
 - 2387, 2401, 2405, 2431, 2435, 2492, 2499, 2513, 2517,
 - 2520, 2528, 2580, 2585, 2598, 2602, 2620, 2627, 2636,
 - 2657, 2671, 2684, 4188, 4189, 4235, 4274, 4305, 4364, 4427
- else commands:
 - \else: 505, 863, 909, 1931, 3547
- \em 69, 1655
- \eminnershape 1657
- \end 2936, 2954, 2972, 4558

F

- false commands:
 - \c_false_bool 2701
- \familydefault 1725
- fancyhdr 28, 39
- \fangsong 4141
- \fi 1709, 2364, 2372, 2381, 2389, 2404, 2407,
 - 2414, 2434, 2437, 2445, 2485, 2486, 2489, 2492, 2501,
 - 2516, 2519, 2522, 2531, 2543, 2544, 2545, 2570, 2576,
 - 2582, 2588, 2605, 2606, 2624, 2625, 2640, 2641, 2643,

2661, 2687, 2688, 2882, 2886, 2892, 2896, 2904, 2907,
2912, 2913, 2916, 2932, 2950, 2968, 4188, 4189, 4198, 4451

fi commands:

\fi: 405, 539, 576, 595, 863, 909, 1931, 2887, 2897, 3457, 3547
\figurename 2199
figurename 12

file commands:

\g_file_current_name_tl 679, 1782
\file_if_exist:nF 3763, 3765
\file_if_exist:nTF 46, 1769, 1771, 1773, 1785, 3577, 3756
\file_if_exist_input:nF 84
\file_input:n 76, 85, 87, 515, 548

five commands:

\c_five 3385
fntef 29, 39
\font 3650, 3651, 3652, 3653, 3654, 3655
fontset 6, 37, 109
\fontsize 3366

fontspec commands:

\g_fontspec_encoding_tl 977
\l_fontspec_family_tl 1444
_fontspec_namewrap:n 55, 996
\fontspec_set_family:Nnn 1056, 1421
\footnotesep 108, 3543
\footnotesize 3479, 3518, 3542

four commands:

\c_four 309, 569, 571

fp commands:

\fp_compare:nNnF 3539
\fp_compare:nNnT 3355
\fp_eval:n 2110, 4509
\fp_use:N 2047, 3541

G

GBK 6, 37
\gdef 1700, 2732, 2733, 2734, 2735, 2738, 2739, 2740, 2741
\global 2495, 2673, 2676
\globaljfont 834
\glueexpr 2367, 2391, 2421, 2462,
2478, 2534, 2552, 2556, 2566, 2587, 2612, 2629, 2651,
2658, 2659, 2669, 2683, 2927, 2937, 2945, 2955, 2963, 2973

group commands:

\group_begin: . 488, 721, 729, 796, 818, 844, 886, 913,
1001, 1053, 1330, 1415, 1500, 1520, 1536, 1721, 2272,
2280, 2811, 2857, 3542, 4485, 4666, 4742, 4750, 4761, 4816
\group_end:
... 492, 501, 725, 741, 801, 822, 857, 890, 917, 1016,
1063, 1350, 1422, 1506, 1532, 1543, 1733, 2283, 2287,
2834, 2861, 3542, 4496, 4666, 4764, 4811, 4840, 4874, 4898
\group_insert_after:N 854, 855
\gtfamily 1656

H

hbox commands:

\hbox_set:Nn 1899, 2850
heading 9, 39
\heiti 4140

\hskip ... 2612, 2629, 2651, 2658, 2659, 2682, 2683, 2903, 2906
\hspace 2399, 2400, 2429, 2430, 2512
\Huge 2446, 2470, 2547, 2561, 3053, 3080, 3496, 3535
\huge 2415,
2443, 2457, 2541, 3036, 3052, 3056, 3078, 3085, 3495, 3534
hyperref 29, 39
\hypersetup 356, 357, 1795
\hyphenchar 3650, 3651, 3652, 3653, 3654, 3655

I

if commands:

\if_bool:N 502, 583
\if_case:w 401, 3453
\if_cs_exist:N 556
\if_false: 105
\if_meaning:w 1930
\if_true: 104
\ifctexpdf 34, 104, 105
\ifdim 2574, 2608, 2647, 2665
\ifincsname 3547
\IfNoValueF 435, 441
\IfNoValueTF 1208, 1469, 1574, 1607, 1609, 2339
\ifnum 1707,
2396, 2411, 2426, 2442, 2504, 2538, 2596, 2620, 2636, 4188
\ifodd 2368, 2397, 2411, 2427, 2442, 2497, 2506, 2540,
2578, 2599, 2609, 2621, 2637, 2648, 2666, 2882, 2886,
2892, 2896, 2904, 2907, 2912, 2913, 2916, 2930, 2948, 2966
\ifx 4186, 4187, 4192
\ifzhmappdf 4204, 4271, 4302, 4333, 4400
\ignorespaces 2689
\immediate 4195
indent 28, 37
\indexname 2202
indexname 12
\input 4457, 4522
\insertcontinuationtext 2211, 2220, 2227
\insertpart 2934, 2935
\insertromanpartnumber 2925, 3023
\insertsection 2952, 2953
\insertsectionnumber 2943, 3115
\insertsubsection 2970, 2971
\insertsubsectionnumber 2961, 3143
int commands:
\int_compare:nNnF 2127, 3353
\int_compare:nNnTF 562, 2083, 3199, 3215, 3231
\int_const:Nn 988
\int_div_truncate:nn 79, 569
\int_eval:n 1470, 1472, 1473, 1575, 1577, 1578
\int_from_hex:n 561
\int_gincr:N 1148, 1435
\int_incr:N 2705
\int_mod:nn 571
\int_new:N 57, 1438
\int_set:Nn 552, 560, 2080, 2122
\int_set_eq:NN 546
\int_sub:Nn 565, 2128

<code>\int_to_Hex:n</code>	563, 568, 570	<code>listfigurename</code>	12
<code>\int_use:N</code>	485, 552, 901, 1150, 1436, 2709, 4490	<code>\listtablename</code>	2198
<code>\int_zero:N</code>	2702	<code>listtablename</code>	12
<code>\interlinepenalty</code>	2408, 2439, 2453, 2467, 2546, 2560, 2613, 2614, 2652, 2653	<code>\LoadClass</code>	409, 414, 419, 424
iow commands:		<code>\long</code>	4193
<code>\iow_close:N</code>	4479	<code>\lstlistingname</code>	779, 780
<code>\iow_indent:n</code>	4653	<code>\lstlistlistingname</code>	781, 782
<code>\iow_new:N</code>	4481	<code>\ltjdefcharrange</code>	688, 689, 690, 692, 696, 697, 700, 703
<code>\iow_now:Nx</code>	4491	<code>LTJFONTUID</code>	64
<code>\iow_open:Nn</code>	4476	<code>\ltjgetparameter</code>	1619, 1904, 1934
<code>\itemsep</code>	3476, 3487, 3515, 3526	<code>\ltjsetkanjiskip</code>	1921
<code>\itshape</code>	1656, 1657	<code>\ltjsetmathletter</code>	746
J		<code>\ltjsetparameter</code>	704, 710, 1636
<code>\jfam</code>	953, 989	<code>\ltjsetxkanjiskip</code>	1622
K		lua commands:	
<code>\kaishu</code>	4142	<code>\lua_escape_x:n</code>	
<code>\kanjifamily</code>	1646, 1647, 1648, 1651	<code>..</code>	862, 874, 901, 908, 1460, 1475, 1476, 1493, 1494, 1503
<code>\kanjifamilydefault</code>	1654	<code>\lua_now_x:n</code>	722,
<code>\kanjiskip</code>	1924, 1938	<code>756, 836, 862, 866, 871, 876, 898, 907, 975, 1481, 1505, 1753</code>	
kernel commands:		M	
<code>\l_kernel_expl_bool</code>	68	<code>\MAKESPA</code>	4482, 4524
keys commands:		mark commands:	
<code>\l_keys_choice_tl</code>	292	<code>\q_mark</code>	4823, 4826
<code>\keys_define:nn</code>	153, 386, 389, 993,	<code>\markboth</code>	2417, 2438
<code>1323, 1432, 1509, 1547, 1586, 1979, 2000, 2018, 2052,</code>		<code>\mathalpha</code>	734
<code>2173, 2194, 2222, 2232, 2235, 2238, 2355, 2716, 3550, 3597</code>		<code>\mathbin</code>	734
<code>\keys_if_exist:nnF</code>	1584	<code>\mathfence</code>	735
<code>\l_keys_key_tl</code>	139, 141, 143	<code>\mathgroup</code>	953
<code>\keys_set:nn</code>	27, 169, 175,	<code>\mathop</code>	735
<code>231, 240, 249, 275, 285, 292, 299, 329, 335, 428, 435,</code>		<code>\mathord</code>	734
<code>441, 3014, 3068, 3092, 3123, 3157, 3173, 3186, 3201,</code>		<code>\mathpunct</code>	735
<code>3209, 3217, 3225, 3232, 3233, 3235, 3352, 3356, 3358, 3639</code>		<code>\mathrel</code>	734
<code>\keys_set_known:nn</code>	3280	max commands:	
<code>\keys_set_known:nnN</code>	1005	<code>\c_max_dim</code>	2067
L		<code>\mcfamily</code>	1656
<code>\labelformat</code>	2998	<code>\mddefault</code>	973, 979, 985, 1311, 1316
<code>\languagealias</code>	3309, 3314, 3341, 3346	minus commands:	
<code>\LARGE</code>	3494, 3533	<code>\c_minus_one</code>	150, 546, 3353
<code>\Large</code>	2412, 3034, 3040, 3102, 3106, 3493, 3532	mode commands:	
<code>\large</code>	3127, 3492, 3531	<code>\mode_if_math:T</code>	947
<code>\lastbox</code>	2677	<code>\mode_if_math:TF</code>	752
<code>\leavevmode</code>	2364, 2570	msg commands:	
left commands:		<code>\msg_critical:nnn</code>	3566
<code>\c_left_brace_str</code>	4808	<code>\msg_critical:nnx</code>	51, 679
<code>\leftmargin</code>	3473, 3484, 3512, 3523	<code>\msg_error:nn</code>	3617
<code>\leftmargini</code>	3473, 3484, 3512, 3523	<code>\msg_error:nnn</code> ..	19, 22, 38, 41, 670, 683, 686, 3367, 3449
<code>\let</code>	1713, 2597, 2604, 3466, 3505, 4179, 4185	<code>\msg_error:nnx</code>	2189
<code>\linespread</code>	3541	<code>\msg_error:nnxx</code>	3583, 3612
<code>linespread</code>	10, 36	<code>\msg_fatal:nn</code>	4463
<code>linestretch</code>	11, 78	<code>\msg_new:nn</code>	4461
<code>\linewidth</code>	2081, 2089, 2121	<code>\msg_new:nnn</code>	138, 140, 142, 144, 146, 673,
<code>\lishu</code>	4144	<code>1047, 1116, 1198, 1853, 2215, 3254, 3567, 3626, 4649, 4726</code>	
<code>\listfigurename</code>	2197	<code>\msg_new:nnnn</code>	12, 30, 43, 97, 2191, 3369, 3591, 3620

- \msg_warning:nnn 167, 173, 201, 211, 226, 235,
244, 273, 283, 297, 327, 333, 341, 346, 352, 1111, 3554, 3604
- \msg_warning:nnnn 433, 439, 1864, 1871, 2145, 2151
- \msg_warning:nxx 3273, 4655, 4725
- \msg_warning:nxxx 1043
- N**
- nan commands:
 - \c_nan_fp 180
 - \newCJKfontfamily 59, 1205
 - \NewDocumentCommand ... 428, 431, 437, 1077, 1203, 1205,
1213, 1218, 1224, 1230, 1235, 1240, 1242, 1370, 1462,
1567, 1602, 1649, 2035, 2106, 2143, 2149, 2168, 2170,
2333, 3361, 3546, 4139, 4140, 4141, 4142, 4144, 4145, 4147
 - \newfontfeature 990, 991, 992
 - \newpage 2479, 2484
- no commands:
 - \q_no_value 4823
 - \nobreak 2412,
2419, 2460, 2541, 2547, 2550, 2561, 2564, 2667, 3035, 3079
 - \nobreakspace 2412, 2443
 - nocap 28
 - \noexpand 4177, 4179, 4180, 4181
 - nofonts 28
 - noindent 28
 - nopunct 28
 - \normalem 1836
 - \normalfont 535, 537, 538, 1227,
1233, 1238, 1268, 1270, 1271, 1294, 2409, 2410, 2440,
2441, 2454, 2455, 2468, 2469, 2535, 2536, 2557, 2558, 2713
 - \normalsize 3160, 3176, 3189, 3460, 3499, 3537, 3545
 - nospace 28
 - nozhmap 28
 - \null 2390, 2391, 2482
 - \numberline 95,
2511, 2512, 2622, 2623, 2638, 2639, 2862, 2865, 2871, 2875
- O**
- one commands:
 - \c_one 160, 1463, 1568, 2135, 2334, 4502
 - \onecolumn 2385
- or commands:
 - \or: 403, 3455
- P**
- \pagestyle 3300
- \par 2365, 2412, 2443, 2446, 2457,
2470, 2541, 2547, 2561, 2571, 2667, 2926, 2927, 2934,
2937, 2944, 2945, 2952, 2955, 2962, 2963, 2970, 2973,
3016, 3035, 3046, 3063, 3071, 3079, 3118, 3120, 3151, 3153
- paragraph/afterskip 30
- paragraph/beforekip 30
- paragraph/numbering 15
- paragraph/runin 19
- \parindent
2102, 2103, 2125, 2146, 2147, 2152, 2153, 2409, 2410,
2454, 2455, 2535, 2536, 2557, 2558, 2928, 2946, 2964, 3233
- \parsep 3475, 3476, 3486, 3487, 3514, 3515, 3525, 3526
- \part 85, 2363, 2376
- part/beforekip 30
- part/numbering 15
- part/pagestyle 21, 84
- \partname 2412, 2443, 2925, 3018
- \PassOptionsToClass 408, 413, 418, 423
- \PassOptionsToPackage 503, 1799, 1827, 2155, 4657
- \pdfmapline 4205,
4206, 4207, 4208, 4209, 4210, 4211, 4213, 4214, 4215,
4216, 4217, 4218, 4219, 4221, 4222, 4223, 4224, 4225,
4226, 4228, 4229, 4230, 4231, 4232, 4233, 4334, 4335,
4336, 4337, 4338, 4339, 4340, 4342, 4343, 4344, 4345,
4346, 4347, 4348, 4350, 4351, 4352, 4353, 4354, 4355,
4357, 4358, 4359, 4360, 4361, 4362, 4401, 4402, 4403,
4404, 4405, 4406, 4408, 4409, 4410, 4411, 4412, 4413,
4415, 4416, 4417, 4418, 4419, 4421, 4422, 4423, 4424, 4425
- \pdfoutput 4188
- \pdfstringdefDisableCommands 3549
- pdftex commands:
 - \pdftex_pdffontattr:D 484
 - \pdftex_pdflastobj:D 485
 - \pdftex_pdfobj:D 481
 - \phantomsection 2360
- prg commands:
 - \prg_do_nothing:
957, 1409, 1501, 1880, 1954, 1984, 2699, 4850
 - \prg_new_conditional:Npnn 860, 1927
 - \prg_new_protected_conditional:Npnn 1090, 4819
 - \prg_return_false: ... 863, 1100, 1931, 1935, 1939, 4830
 - \prg_return_true: 863, 1093, 1098, 1931, 1935, 1939, 4835
 - \primitive 4558
 - \ProcessKeysOptions 398
 - \proofname 2204
 - proofname 12
- prop commands:
 - \prop_get:NnN 1185
 - \prop_get:NnNT 1402
 - \prop_get:NnNTF 1103, 1144, 1182, 1450, 3365, 3420, 3447
 - \prop_get:NxNTF 1092
 - \prop_gpop:NnNT 1032, 1036, 1537
 - \prop_gput:Nnn 1009, 1172, 3382
 - \prop_gput:Nno 1010, 1058, 1525, 1561
 - \prop_gremove:Nn 1041
 - \prop_if_empty:Nf 1106, 1275
 - \prop_map_break:n 1289
 - \prop_map_inline:Nn 1287
 - \prop_new:N 1027, 1028, 1029, 1177, 1545, 1566, 3376
 - \protect .. 2511, 2512, 2524, 2525, 2622, 2623, 2638, 2639,
2816, 2818, 2822, 2824, 2828, 2830, 2838, 2843, 4780, 4781
 - \ProvidesFile 4193, 4201, 4268, 4299, 4330, 4397
 - \providetranslation
4563, 4564, 4565, 4566, 4567, 4568, 4569,
4570, 4571, 4572, 4573, 4574, 4575, 4576, 4577, 4578,
4579, 4580, 4581, 4582, 4583, 4584, 4585, 4586, 4587,
4588, 4589, 4590, 4591, 4592, 4593, 4594, 4595, 4596,

4597, 4598, 4599, 4600, 4601, 4602, 4603, 4604, 4605,
4606, 4607, 4608, 4609, 4610, 4611, 4612, 4613, 4614,
4615, 4616, 4617, 4618, 4619, 4620, 4621, 4622, 4623,
4624, 4625, 4626, 4627, 4628, 4629, 4630, 4631, 4632,
4633, 4634, 4635, 4636, 4637, 4638, 4639, 4640, 4641, 4642
punct 9, 28, 38, 77
\punctstyle 580, 2005

Q

\quad . 2593, 2881, 2882, 2885, 2886, 2892, 2896, 2904, 2907,
2912, 2913, 2916, 3041, 3086, 3096, 3128, 3161, 3177, 3190
quark commands:
 \quark_if_no_value:N\TF 4829
\quotation 108, 3562

R

\raggedright 2409, 2454, 2535, 2557, 3033, 3077
\refname 2205, 2210, 2217, 2226, 2234, 2235
refname 12
\refstepcounter 97, 2398, 2428, 2507, 2600, 3003, 3004
\relax 2367, 2391, 2410, 2421, 2426, 2442, 2455, 2462, 2478,
2534, 2536, 2552, 2556, 2558, 2566, 2572, 2587, 2601,
2607, 2612, 2629, 2646, 2651, 2658, 2659, 2664, 2669,
2681, 2683, 2903, 2906, 2927, 2928, 2937, 2945, 2946,
2955, 2963, 2964, 2973, 4151, 4155, 4158, 4186, 4187, 4192
\renewcommand 2363, 2376, 2492
\RenewDocumentCommand 3004
\RequirePackage 27,
54, 55, 348, 359, 504, 507, 508, 509, 662, 681, 684,
718, 726, 778, 1808, 1817, 1835, 1841, 2156, 3323, 4702
\RequirePackageWithOptions 4658
resetalternatefont 26, 65
reverse commands:
 \reverse_if:N 583
\rmdefault 1727
\rmfamily 523, 526, 1256, 1259, 1646

S

scan commands:
 \scan_stop:
 ... 65, 118, 834, 953, 1502, 1503, 2073, 2081, 4486, 4668
scheme 9, 39
\scriptsize 3490, 3529
\secdef 2373, 2392, 2502
section/afterskip 30
section/beforeskip 30
section/numbering 15
section/runin 19
\sectionname 2943, 3112
\selectfont
1084, 1528, 1541, 1652, 1695, 1956, 1962, 1976, 2037, 3366
seq commands:
 \seq_clear:N 1003
 \seq_concat:NNN 1343
 \seq_gput_left:Nn 2253, 2254
 \seq_gput_right:Nn 1110, 3388
 \seq_gset_eq:NN 2252

 \seq_gset_from_clist:Nn 2247, 2257, 4466
 \seq_if_empty:N\TF 1358
 \seq_if_in:Nn\TF 1108
 \seq_if_in:Nn\T 461
 \seq_map_function:NN 2762
 \seq_map_inline:Nn 1366, 2350, 2703, 2992, 4489
 \seq_new:N .. 454, 1115, 1351, 1352, 2246, 2250, 3377, 4465
 \seq_put_right:No 1614
 \seq_set_filter:NNn 1341
 \seq_set_from_clist:Nn 455
 \seq_set_split:Nn\T 1340
 \seq_use:Nnnn 3374
\setbox 2677
\setCJKfamilyfont 59, 1129,
1203, 3783, 3789, 3790, 3791, 3792, 3793, 3794, 3837,
3838, 3846, 3847, 3849, 3850, 3851, 3852, 3854, 3876,
3878, 3880, 3882, 3921, 3922, 3923, 3924, 3948, 3953,
3958, 3960, 3998, 4000, 4002, 4003, 4017, 4018, 4019,
4020, 4033, 4034, 4035, 4036, 4037, 4038, 4079, 4080,
4081, 4082, 4083, 4084, 4097, 4098, 4099, 4132, 4133, 4134
\setCJKmainfont .. 59, 1125, 1224, 1229, 1245, 3775, 3833,
3841, 3868, 3914, 3936, 3993, 4014, 4029, 4074, 4093, 4128
\setCJKmathfont 59, 1240, 1248
\setCJKmonofont 59, 1127, 1235, 1247, 3788, 3836,
3845, 3875, 3920, 3947, 3997, 4016, 4032, 4078, 4096, 4131
\setCJKromanfont 1229, 1249
\setCJKsansfont .. 59, 1126, 1230, 1246, 3778, 3782, 3835,
3843, 3874, 3919, 3942, 3996, 4015, 4031, 4076, 4095, 4130
\SetSymbolFont 986, 1315, 1319
\sfdefault 1728
\sffamily 527, 530, 1260, 1263, 1647
six commands:
 \c_six 3385
skip commands:
 \skip_horizontal:N 1914, 1918
 \skip_if_eq:nn\T 1618
 \skip_if_eq:nn\TF 1934, 1938
 \skip_new:N 1625, 1926
 \skip_set:Nn 1621, 1626, 1904, 1907, 2074, 2085, 2130, 2138
 \skip_zero:N 2094
\small 3468, 3507
\songti 4139
\space 2508, 2541, 3018, 3075, 3112, 3142, 3241
space 10, 28, 38, 77
space commands:
 \c_space_tl 485
\special 1675, 1676, 1679,
1680, 1815, 1830, 4236, 4237, 4238, 4239, 4240, 4241,
4242, 4244, 4245, 4246, 4247, 4248, 4249, 4250, 4252,
4253, 4254, 4255, 4256, 4257, 4259, 4260, 4261, 4262,
4263, 4264, 4275, 4276, 4277, 4278, 4279, 4281, 4282,
4283, 4284, 4285, 4287, 4288, 4289, 4290, 4292, 4293,
4294, 4295, 4306, 4307, 4308, 4309, 4310, 4312, 4313,
4314, 4315, 4316, 4318, 4319, 4320, 4321, 4323, 4324,
4325, 4326, 4365, 4366, 4367, 4368, 4369, 4370, 4371,
4373, 4374, 4375, 4376, 4377, 4378, 4379, 4381, 4382,

4383, 4384, 4385, 4386, 4388, 4389, 4390, 4391, 4392,
4393, 4428, 4429, 4430, 4431, 4432, 4434, 4435, 4436,
4437, 4438, 4440, 4441, 4442, 4443, 4445, 4446, 4447, 4448
\SplitArgument 1463, 1568, 2334
stop commands:
 \q_stop 118, 123, 912,
 915, 1367, 1371, 1598, 1603, 4788, 4793, 4802, 4823, 4826
str commands:
 \str_case:nnTF 4778
 \str_case:onF 1725
 \str_case:onTF 4793
 \str_case_x:nnF 1123, 1134
 \str_const:Nn 1789
 \str_const:Nx 48, 1781
 \str_if_empty:Nf 4909
 \str_if_empty:Ntf 4865, 4901
 \str_if_eq:nnF 3623
 \str_if_eq:nnTF 961
 \str_if_eq:onF 584, 3630
 \str_if_eq:onTF
 506, 1804, 1812, 3304, 3306, 3320, 3327, 3333, 3606
 \str_if_eq_x:nnF 579
 \str_if_eq_x:nnTF 1279, 3603, 4784
 \str_lower_case:n 480
 \str_new:N 4702, 4813, 4814, 4815
 \str_set:Nn 4832, 4833, 4834
 \str_upper_case:n 1782
\strutbox 3543
sub3section 9, 39
sub4section 9, 39
subparagraph/afterskip 30
subparagraph/beforeskip 30
subparagraph/numbering 15
subparagraph/runin 19
subsection/afterskip 30
subsection/beforeskip 30
subsection/numbering 15
subsection/runin 19
\subsectionname 2961, 3142
subsubsection/afterskip 30
subsubsection/beforeskip 30
subsubsection/numbering 15
subsubsection/runin 19
sys commands:
 \c_sys_engine_str 46, 49, 51
 \sys_if_engine_pdftex:TF
 128, 3770, 3861, 3929, 4008, 4025, 4089
 \sys_if_engine_uptex:TF
 3814, 3902, 3981, 4011, 4058, 4117
 \sys_if_engine_xetex:F 4459
 \sys_if_engine_xetex:TF 4077
 \sys_if_output_pdf:TF 103, 490, 1807, 3863, 3931

T
\tablename 2200
tablename 12

\TeX 3547
T_EX and L^AT_EX 2_ε commands:
 \@italiccorr 68, 1638
 \@par 2613, 2652, 3097, 3129, 3162, 3203, 3219
 \@afterheading 2422, 2463, 2530, 2670
 \@afterindentfalse 2371, 2496, 2500, 2575, 2581
 \@afterindenttrue 2369, 2498, 2573, 2579
 \@Alph 3237, 3242
 \@chapapp 2508, 2541, 2891, 2911
 \@chapter 88, 2502, 2503
 \@clubpenalty 2685
 \@currentlabel 2983, 2987
 \@currentxt 4, 4648, 4697
 \@currname 4, 4648, 4698
 \@dblarg 2591
 \@defaultunits 34
 \@empty 98, 2597, 2604
 \@endpart 87, 2449, 2473, 2476
 \@EverySelectfont@AtNextHook 1699, 1700
 \@EverySelectfont@EveryHook 1698
 \@EverySelectfont@Init 70, 1693, 1713, 1959
 \@font@info 1708
 \@hangfrom 2612, 2651
 \@ifclassloaded 3250
 \@ifpackagelater 37, 40, 669, 682, 685, 766, 1842
 \@ifpackageloaded
 18, 21, 678, 1703, 1792, 2801, 2866, 3000, 4679
 \@ifpackagewith 2746, 2755, 2806
 \@ifstar 2589
 \@listI 3466, 3505
 \@listi 3466, 3473, 3484, 3505, 3512, 3523
 \@M 2408,
 2439, 2453, 2467, 2546, 2560, 2613, 2614, 2652, 2653, 2678
 \@makechapterhead 88, 2527, 2529, 2532
 \@makeschapterhead 88, 2554
 \@minus 3098, 3130, 3163, 3178, 3191, 3462,
 3464, 3470, 3472, 3474, 3475, 3481, 3483, 3485, 3486,
 3501, 3503, 3509, 3511, 3513, 3514, 3520, 3522, 3524, 3525
 \@namedef 2816, 2818, 2822, 2824, 2828, 2830, 2838, 2843
 \@nameuse 2816, 2818, 2828, 2830, 2838
 \@ne 2799
 \@nil 887
 \@nobreakfalse 2672
 \@noskipsecfalse 2676
 \@noskipsectrue 2673
 \@onlypreamble 34, 451, 517,
 1245, 1246, 1247, 1248, 1249, 1250, 1670, 1684, 1692, 3590
 \@part 86, 2373, 2392, 2395, 2425
 \@pgkextension 6, 7, 8, 9, 20, 23, 680, 4697
 \@plus 3047,
 3048, 3098, 3099, 3130, 3131, 3163, 3164, 3178, 3191,
 3204, 3220, 3462, 3463, 3464, 3470, 3471, 3472, 3474,
 3475, 3481, 3482, 3483, 3485, 3486, 3501, 3502, 3503,
 3509, 3510, 3511, 3513, 3514, 3520, 3521, 3522, 3524, 3525
 \@popfilename 129, 4695
 \@schapter 2502

\@secntformat	89, 2592, 2601	\CJK@loadEncoding	44
\@secpentalty	2586, 2587	\CJK@makeActive	45, 586
\@sect	89, 2591, 2595	\CJK@plane	44, 76, 469, 470, 480, 1964
\@setfontsize	3452	\CJK@surr	44, 44, 44, 556, 575
\@spart	87, 2373, 2392, 2452, 2466	\CJK@upperReset	45
\@ssect	90, 2590, 2645	\CJKfamily	56
\@startsection	89, 91, 92, 98, 2569, 2708	\CJKfamilydefault	45, 45, 60, 60, 70, 70, 70, 71, 71, 135
\@svsec	2597, 2601, 2604, 2612, 2630, 2631	\CJKglue	79, 79, 108
\@svsechd	2628, 2658, 2659, 2679	\CJKhook	46, 46
\@tempdima	2851, 2853	\CJKrmdefault	43
\@tempskipa	2572, 2574, 2575, 2586, 2607, 2608, 2646, 2647, 2664, 2665, 2668, 2681, 2682	\CJKsymbol	76
\@tempswafalse	2388	\CJKunderdotbasesep	29
\@tempswattrue	2386	\CJKunderline	29
\@topnewpage	2527	\clearpage	45
\@topnum	2495	\contentsname	11, 11
\@xsect	91, 2644, 2662, 2663	\CS	51, 51
\abstractname	12	\csname	108
\addCJKfontfeature	65	\CTEX@afterindent	2578, 2695, 2700
\AfterEndPreamble	128	\CTEX@aftertitle	2655, 2660, 2694, 2699
\algorithmname	12	\CTEX@appendix@number	2721, 2733, 2739
\appendixname	12, 22, 22, 22	\CTEX@appendix@numbering	2722, 2735, 2741
\AtBeginDocument	34, 44	\CTEX@chapter@afterindent	2497
\AtBeginDvi	42	\CTEX@chapter@aftername	2542
\AtBeginShipoutFirst	42	\CTEX@chapter@afterskip	2552, 2566
\AtEndOfClass	35	\CTEX@chapter@aftertitle	2549, 2563
\AtEndOfPackage	35	\CTEX@chapter@beforeskip	2534, 2556
\AtEndOfPackageFile*	129	\CTEX@chapter@format	2537, 2559
\AtEndPreamble	128	\CTEX@chapter@indent	2536, 2558
\baselineskip	10, 11, 68, 79, 79	\CTEX@chapter@numbering	2506, 2540, 2741, 2892, 2912, 2913
\bfseries	46	\CTEX@chapter@pagestyle	2347, 2494
\bibname	12	\CTEX@chapter@titleformat	2548, 2562
\c@chapter	3242	\CTEX@chaptername	2542
\c@secnumdepth	2396, 2411, 2426, 2442, 2504, 2538, 2596, 2620, 2636	\CTEX@document@left@hook	4665, 4671
\c@section	3237	\CTEX@document@right@hook	4667, 4673
\caption	29	\CTEX@Family@CMap	466, 494, 497
\catcode	33, 33, 44, 44, 120, 129	\CTEX@makeheadinganchor	85, 2358, 2360, 2402, 2432, 2514, 2603
\ccwd	10, 11, 11, 23, 23, 29, 29, 34, 75, 76, 76, 78, 78, 79, 80	\CTEX@part@afterindent	2368
\chapter	9, 15, 21, 22, 22, 102	\CTEX@part@aftername	2413, 2444, 2931
\chaptermark	13	\CTEX@part@afterskip	2421, 2462, 2478, 2937
\chaptername	15	\CTEX@part@aftertitle	2418, 2448, 2459, 2472, 2935
\char	48	\CTEX@part@beforeskip	2367, 2391, 2927
\CheckCommand	76	\CTEX@part@format	2410, 2441, 2456, 2469, 2929
\Chinese	29, 29	\CTEX@part@indent	2410, 2455, 2928
\chinese	24, 24, 29, 29, 81, 135	\CTEX@part@numbering	2397, 2411, 2427, 2442, 2930
\CJK@ignorespaces	45, 597	\CTEX@part@pagestyle	2346, 2383
\CJK@encoding	55, 55, 56, 64, 793, 797, 881, 968, 969, 973, 975, 977, 978, 979, 981, 985, 987, 1310, 1313, 1315, 1319, 1460, 1475, 1476, 1493, 1494	\CTEX@part@titleformat	2416, 2447, 2458, 2471, 2935
\CJK@envStart	45, 45, 45, 45, 591	\CTEX@partname	2413, 2444, 2931
\CJK@family	50, 50, 788, 790, 793, 798, 845, 881, 1081, 1527, 1537, 1539, 1976, 1977	\CTEX@postappendix	2726, 2734, 2740
\CJK@ignorespaces	46, 46, 602, 606	\CTEX@postchapter	2740
\CJK@input	44, 45, 555	\CTEX@postsection	2734
\CJK@loadBinding	44, 45, 587	\CTEX@preappendix	2725, 2732, 2738
		\CTEX@prechapter	2738
		\CTEX@presection	2732
		\CTEX@runin	2609, 2648, 2666, 2696, 2701

\CTEX@save@appendix	2727, 2730	\DeclareRobustCommand	131
\CTEX@save@refstepcounter	3003, 3006	\define@newfont	50, 821
\CTEX@section@aftername	2949	\do@subst@correction	50, 51, 820
\CTEX@section@afterskip	2955	\em	69, 69, 69
\CTEX@section@aftertitle	2953	\enc@update	1696, 1701
\CTEX@section@beforeskip	2945	\end	135
\CTEX@section@format	2947	\escapechar	54
\CTEX@section@indent	2946	\EverySelectfont	76
\CTEX@section@numbering	2735, 2882, 2896, 2904, 2916, 2948	\external@font	831, 832, 834
\CTEX@section@titleformat	2953	\extract@font	50, 819
\CTEX@sectionname	2949	\f@baselineskip	2073
\CTEX@selectfont@hook	76, 1951, 1952, 1954, 1957, 1964, 1967	\f@encoding	797, 837, 950
\CTEX@setcurrentlabel@n	2981, 3007	\f@family	50, 798, 837, 845
\CTEX@subsection@aftername	2967	\f@series	793, 881
\CTEX@subsection@afterskip	2973	\f@shape	793, 881
\CTEX@subsection@aftertitle	2971	\f@size	793, 799, 876, 884, 887, 888, 1900, 1946, 2854
\CTEX@subsection@beforeskip	2963	\fangsong	8, 8
\CTEX@subsection@format	2965	\figurename	12
\CTEX@subsection@indent	2964	\font	51, 51
\CTEX@subsection@numbering	2886, 2907, 2966	\font@name	50, 51, 802, 809, 810, 813, 816, 834, 835, 852, 930, 931, 932, 933, 934, 941, 1708
\CTEX@subsection@titleformat	2971	\fontcharwd	76, 76
\CTEX@subsectionname	2967	\footnotesep	107, 108, 108, 135
\CTEX@thechapter	2739	\footnotesize	108
\CTEX@thesection	2733	\get@external@font	828
\CTEX@titleformat@n	91, 2654, 2660, 2693, 2698	\getanddefine@fonts	54, 926, 927
\CTEX@titleslabel@clear	94, 2825, 2840	\globaljfont	51
\CTEX@titleslabel@set	94, 2819, 2831, 2835	\heiti	8
\CTEX@toc@width@n	2848, 2863, 2873	\Hy@driver	1812
\CTEX@todayold	2172, 2177	\Hy@unicodetrue	73
\CTEX@update@sectionformat@n	91, 2577, 2691	\hypersetup	135
\CTEX@verbatim@font@hook	1631, 1635	\HyPsd@ConvertToUnicode	73
\ctex@zhmap@endinput	4453	\HyPsd@LoadUnicode	73
\CTEXdigits	24, 24	\HyPsd@pdfencoding	73, 73
\CTEXindent	135	\HyPsd@ToBigChars	73
\CTEXnoindent	135	\if@mainmatter	2505, 2539, 2912, 3268, 3269
\CTEXnumber	24, 24, 24, 24, 24	\if@nobreak	2583
\CTEXoptions	29, 29, 135	\if@noskipsec	2364, 2570, 2675
\ctexset 5, 5, 5, 5, 5, 5, 7, 11, 14, 14, 25, 26, 29, 29, 109, 135		\if@openright	2377, 2481, 2492
\CTEXsetup	29, 135	\if@tempswa	2487
\CTEXsout	135	\if@twocolumn	2384, 2526
\CTEXthechapter	13, 13, 13, 16	\if@twoside	2480, 2883, 2893
\CTEXthesection	13, 13, 16, 96	\iffalse	53
\CTEXunderdblline	135	\ifin@	53, 909
\CTEXunderdot	135	\iftrue	53
\CTEXunderdotbasesep	29	\indexname	12
\CTEXunderline	29, 135	\insertcontinuationtext	12
\CTEXunderwave	135	\itdefault	51
\CTEXxout	135	\itshape	47
\curr@fontshape	799, 829, 838, 842, 888	\kaishu	8
\DeclareAlternateKanjiFont	64	\labelformat	98, 98
\DeclareFontEncoding	53	\leftmark	13
\DeclareFontFamily	42, 51	\linewidth	79, 79
\DeclareFontShape	51	\lishu	8, 8
\DeclareFontShape@	64, 1419	\listfigurename	12
		\listtablename	12

<code>\lowercase</code>	135	<code>\rightmark</code>	13
<code>\ltj@@does@alt@set</code>	52	<code>\rmfamily</code>	47
<code>\ltj@@getjfontnumber</code>	53, 897	<code>\roman</code>	24
<code>\ltj@set@stackfont</code>	939, 940, 941	<code>\scantokens</code>	131
<code>\ltj@allalchar</code>	723, 757	<code>\scriptfont@name</code>	933, 940
<code>\ltj@curjfnt</code>	51, 76	<code>\section</code>	9, 9, 14, 15, 19, 20, 20, 21, 22, 22, 30, 83
<code>\ltj@pickup@altfont@auxy</code>	53, 882	<code>\section*</code>	15
<code>\ltj@pickup@altfont@copy</code>	53, 895	<code>\sectionmark</code>	13, 13
<code>\ltj@setpar@global</code>	938	<code>\sectionname</code>	15
<code>\ltj@tempcntc</code>	53, 901	<code>\sectionpage</code>	14, 96
<code>\ltjalchar</code>	48	<code>\selectfont</code>	52, 68, 70, 76, 76, 76, 78
<code>\ltjsetparameter</code>	68	<code>\setCJKfamilyfont</code>	25, 25, 25
<code>\m@ne</code>	2396, 2411, 2504, 2538, 3650, 3651, 3652, 3653, 3654, 3655	<code>\setCJKmainfont</code>	25
<code>\markboth</code>	13	<code>\sf@size</code>	932
<code>\markright</code>	13	<code>\sffamily</code>	47
<code>\math@bgroup</code>	949	<code>\size@update</code>	78, 79, 2037, 2051
<code>\math@egroup</code>	954	<code>\sldefault</code>	51
<code>\math@fonts</code>	936	<code>\songti</code>	8
<code>\mathbb</code>	108	<code>\special</code>	6, 72
<code>\maxdimen</code>	11, 78	<code>\split@name</code>	887
<code>\meaning</code>	131, 132, 132, 133	<code>\SplitArgument</code>	84
<code>\newcommand</code>	131	<code>\ssf@size</code>	934
<code>\newrobustcmd</code>	131	<code>\strutbox</code>	108
<code>\newtitlemark</code>	94, 95	<code>\subparagraph</code>	9, 9, 9, 9, 19, 33
<code>\nfss@catcodes</code>	55	<code>\subsectionname</code>	15
<code>\normalsize</code>	8, 8, 104	<code>\subsectionpage</code>	14, 97
<code>\nouppercase</code>	44, 135	<code>\subst@correction</code>	848, 855
<code>\p@</code>	1900, 1946, 2443, 2524, 2525, 2533, 2541, 2551, 2555, 2565, 2854, 3046, 3072, 3073, 3079, 3462, 3463, 3464, 3470, 3471, 3472, 3474, 3475, 3481, 3482, 3483, 3485, 3486, 3501, 3502, 3503, 3509, 3510, 3511, 3513, 3514, 3520, 3521, 3522, 3524, 3525	<code>\tablename</code>	12
<code>\pagestyle</code>	9	<code>\textfont@name</code>	931, 939
<code>\paragraph</code>	9, 9, 9, 9, 9, 19, 33, 33	<code>\tf@size</code>	930
<code>\parindent</code>	10, 29, 78, 80, 108	<code>\thechapter</code>	13, 16, 22
<code>\part</code>	14, 20, 21	<code>\thesection</code>	13, 16, 16, 22, 96
<code>\partname</code>	15	<code>\titleformat</code>	92
<code>\partpage</code>	14, 96	<code>\titlespacing</code>	92
<code>\PassOptionsToClass</code>	41	<code>\today</code>	11, 11
<code>\PassOptionsToPackage</code>	72	<code>\trans@languagepath</code>	3317, 3349
<code>\pdfmapline</code>	114, 117	<code>\ttfamily</code>	47
<code>\pdfstringdef</code>	73	<code>\ttl@extract</code>	92
<code>\pickup@font</code>	50	<code>\ttl@setsubmark</code>	94, 2821, 2827
<code>\pickup@jfont</code>	825	<code>\ttl@settopmark</code>	94, 2815, 2846
<code>\ProcessKeysOptions</code>	27	<code>\ttlh@hang</code>	2780, 2790
<code>\proofname</code>	12	<code>\ttlh@runin</code>	2779, 2789
<code>\protected</code>	49, 94	<code>\tw@</code>	1707
<code>\protected@edef</code>	2169, 2171, 2601, 2983	<code>\uppercase</code>	44, 44
<code>\ProvidesFile</code>	120	<code>\use@mathgroup</code>	54, 768, 958
<code>\ps@fancy</code>	96, 2902, 2905, 2910, 2914	<code>\usepackage</code>	28, 29
<code>\ps@headings</code>	95, 2880, 2884, 2890, 2894	<code>\verbatim@font</code>	1629, 1633
<code>\refname</code>	12, 12	<code>\wrong@fontshape</code>	50
<code>\refstepcounter</code>	97	<code>\x@protect</code>	4781
<code>\relax</code>	44, 78	<code>\xdef</code>	94
<code>\reset@font</code>	537, 1270	<code>\XeTeXglyphbounds</code>	125
		<code>\yahei</code>	8, 8
		<code>\youyuan</code>	8, 8
		<code>\z@</code>	710, 713, 2409, 2454, 2495, 2535, 2557, 2574, 2608, 2647, 2665, 2677, 2799, 3463, 3471, 3482, 3502, 3510, 3521
		<code>\zhdigits</code>	24

- \zhnum 81, 135
- \zhnumber 24, 24, 24, 24
- \zihao 23, 23
- \ziju 23
- \zw 68, 68, 76
- tex commands:**
 - \tex_def:D 4858
 - \tex_divide:D 79, 2082, 2123
 - \tex_edef:D 4911
 - \tex_endlinechar:D 546, 552
 - \tex_font:D 484, 4486
 - \tex_iftrue:D 3269
 - \tex_ignorespaces:D 1078, 1216, 1221, 1990, 2107, 3362, 4668
 - \tex_immediate:D 481
 - \the 4155, 4158, 4159, 4178, 4182
 - \thechapter 2508, 2511, 2541, 2891, 2911, 3076
 - \theparagraph 3175
 - \thepart 2399, 2412, 2429, 2443, 3020
 - \thesection 2881, 2895, 2903, 2915, 3095
 - \thesubparagraph 3188
 - \thesubsection 2885, 2906, 3126
 - \thesubsubsection 3159
 - \thispagestyle 2382, 2383, 2483, 2493, 2494
- three commands:**
 - \c_three 307, 2127, 3215, 4503
- \tiny 3491, 3530
- tl commands:**
 - \tl_clear:c 1838, 1849, 2340
 - \tl_clear:N 191, 364, 387, 1559, 1563, 1977, 2028, 2356, 3417, 4866
 - \tl_const:cn 657, 680
 - \tl_const:cx 2264, 2270
 - \tl_const:Nn 407, 412, 417, 422, 519, 968, 983, 1252, 1253, 1254, 1255, 3251, 3270, 3272
 - \tl_const:Nx 3, 1309, 2047
 - \tl_gput_right:cn 462
 - \tl_gput_right:Nn 4662, 4664, 4686
 - \tl_gset:Nn 1283, 1779, 1786, 1787
 - \tl_gset:Nx 563, 566, 930, 932, 934, 1723, 1751, 3608
 - \tl_gset_eq:NN 1060, 1746
 - \tl_gset_rescan:Nnn 1290
 - \tl_head:N 3421
 - \tl_if_blank:NtF 1376, 1472, 1473, 1577, 1578, 1592, 1610
 - \tl_if_blank_p:n 1342
 - \tl_if_empty:cT 2997
 - \tl_if_empty:Nf 788, 2100
 - \tl_if_empty:nF 1664, 1677
 - \tl_if_empty:NtF 2276, 3571
 - \tl_if_eq:NNT 1719
 - \tl_if_eq:NNTF 2040
 - \tl_if_exist:Nf 518, 520, 521, 522, 1251, 1253, 1254, 1255, 4685
 - \tl_if_exist:NtF 82, 2220, 2231, 2234, 3263, 3302
 - \tl_if_in:nnT 732
 - \tl_if_in:NnTF 4893
 - \tl_map_inline:Nn 1488
 - \tl_map_inline:nn 3418
 - \tl_new:c 2262, 2263
 - \tl_new:N 56, 126, 130, 520, 521, 522, 790, 791, 879, 965, 1020, 1046, 1066, 1067, 1088, 1424, 1565, 1739, 2048, 2061, 2725, 2726, 3761, 4675, 4676, 4685, 4746, 4747, 4869, 4870, 4871
 - \tl_put_left:Nn 1646, 1647, 1648, 1959, 4695
 - \tl_put_right:Nn 371, 1605, 2313, 2315, 2317, 2319, 2321, 2323
 - \tl_put_right:Nx 936, 1572, 2292, 3421, 3422
 - \tl_replace_all:Nnn 1339, 1380
 - \tl_rescan:nn 133, 4919
 - \tl_rescan:nx 4910
 - \tl_set:cn 2338, 2341
 - \tl_set:Nn 112, 113, 186, 219, 220, 512, 513, 514, 779, 781, 792, 880, 962, 963, 1004, 1071, 1083, 1338, 1377, 1379, 1420, 1593, 1597, 1642, 1643, 1644, 1645, 1654, 1976, 2013, 2024, 2062, 2344, 3549, 3762, 3766, 3808, 3809, 3810, 4740, 4741, 4828
 - \tl_set:Nx 127, 480, 559, 831, 1207, 1741, 2004, 2110, 2776, 2795
 - \tl_set_eq:NN 797, 798, 845, 931, 933, 977, 1072, 1097, 1527, 1539, 2049
 - \tl_set_rescan:Nnx 4867
 - \tl_tail:N 2781
 - \tl_tail:n 2791
 - \tl_to_str:n 123, 124, 4809, 4839, 4880, 4881
 - \tl_trim_spaces:n 1556
 - \today 2172, 2177, 2180, 2185
 - today 11, 81
 - token commands:**
 - \token_get_replacement_spec:N 4766, 4788
 - \token_to_meaning:N 4823
 - \token_to_str:N 54, 912, 921, 930, 932, 934, 1125, 1126, 1127, 1129, 1136, 1137, 1138, 1200, 2217, 2218, 3370, 4493, 4653, 4725
 - \toks 4155, 4156, 4158, 4159, 4178, 4182
 - \topsep 3474, 3485, 3513, 3524
 - \tracingfonts 1707
 - \TrimSpaces 431
 - true commands:**
 - \c_true_bool 2700
 - \ttdefault 1729
 - \ttfamily 531, 534, 1264, 1267, 1648
 - two commands:**
 - \c_two 93, 161, 375, 379, 2128, 2854, 3199, 3231
 - \c_two_hundred_fifty_six 562, 565
 - \twocolumn 2488
 - \typeout 2508, 2509
 - U**
 - um commands:**
 - _um_define_math_chars: 765
 - _um_input_math_symbol_table: 740
 - _um_switchto_literal: 770

`_um_sym:nnn` 730
`\undefined` 1713, 4180
`\unless` 2609, 2648, 2666
`\unskip` 2680
`\updefault` ... 973, 979, 981, 985, 987, 1311, 1313, 1316, 1320
`\upshape` 1656
use commands:
`\use:c` 799, 837, 838, 888, 930, 932, 934, 988, 1038, 1540, 4894, 4913, 4927
`\use:n` 131, 1280, 1284, 1740, 1878, 2698, 2859, 3261
`\use:nn` 4921
`\use:x` 62, 120, 386, 776, 887, 970, 1019, 1155, 1382, 1556, 2355, 3380, 3580, 4762, 4804, 4856, 4877
`\use_i:nn` 106, 909, 4706, 4711, 4731, 4736
`\use_i:nnn` 259
`\use_ii:nn` 107, 909, 4706, 4711, 4731, 4736
`\use_ii:nnn` 132, 264
`\use_iii:nnn` 269
`\use_none:n` .. 187, 192, 197, 1055, 1283, 1417, 1879, 2159
`\use_none:nn` 497
`\usebeamercolor` 2925, 2943, 2961, 3062, 3117, 3150
`\usebeamerfont` 2925, 2934, 2943, 2952, 2961, 2970, 3061, 3064, 3116, 3119, 3149, 3152
`\uselanguage` 3308, 3313, 3340, 3345
`\usepackage` 4653
UTF8 6, 37

V

`\verse` 108, 3560, 3561
`\vfil` 2390, 2477
`\vskip` 2391, 2420, 2421, 2443, 2461, 2462, 2478, 2541, 2551, 2552, 2565, 2566, 2668, 2669, 2926, 2944, 2962, 3046, 3063, 3079, 3118, 3151
`\vspace` 2533, 2534, 2555, 2556

W

`winfonts` 28
`\write` 4195

X

`\x` 4157, 4160, 4161, 4162, 4163, 4164, 4165, 4166, 4167, 4168, 4169, 4170, 4171, 4172, 4173, 4174, 4175, 4176, 4183, 4193, 4197
`\xdef` 4196
`\xeCJKsetup` 663, 1845, 1918, 1985, 1988, 1991, 2006
xetex commands:
`\xetex_charglyph:D` 4490
`\xetex_glyphbounds:D` 4514

Y

`\yahei` 4147
`\youyuan` 4145

Z

zero commands:
`\c_zero` 159, 2083, 3354
`\c_zero_dim` 2075, 2087, 2102, 2118, 2125, 2327, 3232
`\c_zero_fp` 2047, 3355, 3539
`\c_zero_skip` 2325, 2326
`\zhdigits` 24, 2171
`zhmap` 6, 28, 38
`\zhnum` 2157, 2164
zhnum commands:
`\zhnum_int:n` 2164, 2165
`\zhnumber` 24, 2169
`\zhnumsetup` 2181, 2186
`\zhtoday` 2180, 2185
`\zihao` 23, 104, 3361, 3370
`zihao` 8, 36
`\ziju` 23, 80, 2106
`\zw` 711, 1621, 1626, 1904, 1947