

# ctex 宏包说明

*ctex.org*\*

版本号: v0.99b      修改日期: 2010/07/12

## 摘要

ctex 宏包提供了一个统一的中文 L<sup>A</sup>T<sub>E</sub>X 文档框架, 底层支持 CCT、CJK 和 xeCJK 三种中文 L<sup>A</sup>T<sub>E</sub>X 系统。ctex 宏包提供了编写中文 L<sup>A</sup>T<sub>E</sub>X 文档常用的一些宏定义和命令。

ctex 宏包需要 CCT 系统或者 CJK 宏包或者 xeCJK 宏包的支持。主要文件包括 ctexart.cls、ctexrep.cls、ctexbook.cls 和 ctex.sty、ctexcap.sty。

ctex 宏包由 ctex.org 制作并负责维护。

## 目录

<b>1 简介</b>	<b>2</b>
<b>2 使用帮助</b>	<b>3</b>
2.1 使用 CJK 或 xeCJK	3
2.2 使用 CCT	3
2.3 选项	4
2.3.1 只能用于文档类的选项	4
2.3.2 只能用于文档类和 ctexcap.sty 的选项	4
2.3.3 中文编码选项	4
2.3.4 中文字库选项	5
2.3.5 CCT 引擎选项	5
2.3.6 排版风格选项	5
2.3.7 宏包兼容选项	6
2.3.8 缺省选项	6
2.4 基本命令	6
2.4.1 字体设置	6

---

\*<http://www.ctex.org>

1 简介	2
2.4.2 字号、字距、字宽和缩进	7
2.4.3 中文数字转换	7
2.5 高级设置	8
2.5.1 章节标题设置	9
2.5.2 部分修改标题格式	12
2.5.3 附录标题设置	12
2.5.4 其他标题设置	13
2.5.5 其他设置	13
2.6 配置文件	14
3 版本更新	15
4 开发人员	17

## 1 简介

这个宏包的部分原始代码来自于由王磊编写 `cjkbook.cls` 文档类，还有一小部分原始代码来自于吴凌云编写的 `GB.cap` 文件。原来的这些工作都是零零碎碎编写的，没有认真、系统的设计，也没有用户文档，非常不利于维护和改进。2003 年，吴凌云用 `doc` 和 `docstrip` 工具重新编写了整个文档，并增加了许多新的功能。2007 年，oseen 和王越在 `ctex` 宏包基础上增加了对 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<sup>1</sup>，对 `ctex` 宏包及相关宏包和脚本进行了整合，并加入了对 XeTeX 的支持。该项目由 `ctex.org` 社区的开发者共同维护，新版本号为 v0.9。在开发新版本时，考虑到合作开发和调试的方便，我们不再使用 `doc` 和 `docstrip` 工具，改为直接编写宏包文件。

最初 Knuth 设计开发 TeX 的时候没有考虑到支持多国语言，特别是多字节的中日韩语言。这使得 TeX 以至后来的 L<sup>A</sup>TeX 对中文的支持一直不是很好。即使在 CJK 解决了中文字符处理的问题以后，中文用户使用 L<sup>A</sup>TeX 仍然要面对许多困难。最常见的就是中文化的标题。由于中文习惯和西方语言的不同，使得很难直接使用原有的标题结构来表示中文标题。因此需要对标准 L<sup>A</sup>TeX 宏包做较大的修改。此外，还有诸如中文字号的对应关系等等。`ctex` 宏包正是尝试着解决这些问题。中间很多地方用到了在 `ctex.org` 论坛上的讨论结果，在此对参与讨论的朋友们表示感谢。

`ctex` 宏包由五个主要文件构成：`ctexart.cls`、`ctexrep.cls`、`ctexbook.cls` 和 `ctex.sty`、`ctexcap.sty`。`ctex.sty` 主要是提供整合的中文环境，可以配合大多数文档类使用。而 `ctexcap.sty` 则是在 `ctex.sty` 的基础上对 L<sup>A</sup>TeX 的三个标准文档类的格式进行修改以符合中文习惯，该宏包只能配合这三个标准文档类使用。`ctexart.cls`、`ctexrep.cls`、`ctexbook.cls` 则是 `ctex.sty`、`ctexcap.sty` 分别和三个标准文档类结合

<sup>1</sup><http://code.google.com/p/ctex-kit/>

产生的新文档类，除了包含 `ctex.sty`、`ctexcap.sty` 的所有功能，还加入了一些修改文档类缺省设置的内容（如使用五号字体为缺省字体）。

这份说明文档可以通过用  $XeLaTeX$  编译 `ctex.tex` 文件来得到。编译说明文档需要先安装 `ctex` 宏包。

## 2 使用帮助

`ctex` 宏包的使用十分简单。如果是使用 `ctex` 的文档类，只需用 `ctexart`、`ctexrep` 或者 `ctexbook` 替换原来的文档类就可以了。你也可以继续使用原来的文档类，而用 `ctex.sty` 或者 `ctexcap.sty` 宏包来配合使用，两者的效果是一样的（除了不能修改一些文档设置如缺省字体大小）。

### 2.1 使用 CJK 或 xeCJK

这是 `ctex` 宏包的缺省设置。`ctex` 宏包会自动根据使用的  $T_EX$  引擎调用 `CJK` 或者 `xeCJK` 宏包，你无需再自己调用。

此外，`ctex` 宏包会在 `\begin{document}` 和 `\end{document}` 之间自动加入一个 `CJK` 环境，你无需再添加 `CJK` 环境。`CJK` 宏包的命令都可以在 `\begin{document}` 和 `\end{document}` 之间正常使用。

例子 1：使用文档类宏包

```
\documentclass{ctexart}
\begin{document}
中文宏包测试
\end{document}
```

例子 2：使用普通宏包

```
\documentclass{article}
\usepackage{ctex}
\begin{document}
中文宏包测试
\end{document}
```

### 2.2 使用 CCT

注：CCT 方式不再建议使用。

`ctex` 宏包也可以配合新版的 CCT 使用，只需在使用 `ctex` 宏包时加上 CCT 选项即可。缺省 CCT 会使用 `CJK` 字库，因为这种字库方式比传统 CCT 字库更方便，兼容性也更好。如果要使用传统 CCT 字库，则还要加上 `CCTfont` 选项。

例子 3：使用 `CJK` 方式字库

```
\documentclass[CCT]{ctexart}
\begin{document}
中文宏包测试
\end{document}
```

例子 2: 使用 CCT 方式字库

```
\documentclass[CCT,CCTfont]{ctexart}
\begin{document}
中文宏包测试
\end{document}
```

## 2.3 选项

宏包的选项用于改变一些缺省风格的设置。缺省的设置已经针对中文的习惯进行了尽量的修改, 所以一般用户无需使用这些选项。如果你觉得某些设置不合适, 可以向作者反映。我们会考虑在后面的版本中予以改进。我们也欢迎关于增加或者删减选项的建议。

除了 2.3.1 和 2.3.2 的选项, 其余的选项都可以在所有文档类宏包和普通宏包上使用。

### 2.3.1 只能用于文档类的选项

下面的选项可能会是最经常使用的。但是它们只能用于文档类 (`ctexart`、`ctexrep` 和 `ctexbook`)。

**cs4size** 使用小四字号为缺省字体大小。

**c5size** 使用五号字为缺省字体大小。这个是 *ctex* 宏包的缺省模式。

### 2.3.2 只能用于文档类和 `ctexcap.sty` 的选项

下面这些则只可以在文档类宏包和 `ctexcap.sty` 上使用。

**sub3section** 将 `\paragraph` 命令产生的标题改为 section 类格式。此时 `\subparagraph` 命令产生的标题会具有原来 `\paragraph` 的格式。

**sub4section** 将 `\paragraph` 和 `\subparagraph` 命令产生的标题都改为 section 类格式。

### 2.3.3 中文编码选项

下面的选项用于选择 *ctex* 宏包的内部编码。

**GBK** 使用 GBK 编码。这个是 *ctex* 宏包的缺省模式。

**UTF8** 使用 UTF-8 编码。

注意使用 *XeTeX* 引擎的情况下总是内部使用 UTF-8 编码, 所以不必使用这个选项, 但这并不妨碍编写 GBK 编码的文档。

### 2.3.4 中文字库选项

下面的选项用于选择可用的中文字库。设置这些选项是考虑到不同的操作系统平台提供的中文字库是不同的。不同的  $\text{\TeX}$  发行版可以根据目标操作系统平台和提供的中文字库在 `ctexfonts.cfg` 文件中修改这些选项之一为缺省设置。

**nofonts** 没有中文字库，此时没有中文字体命令可用。

**winfont**s 使用 Windows 的字体设置，默认为六种中易字体：宋体、仿宋、黑体、楷体、隶书、幼圆（在使用  $\text{\XeTeX}$  时只有前四种）。该选项的结果将和老版本 `ctex` 宏包完全一致。这是默认设置。

**adobe**font**s** 在 `xeCJK` 模式中使用 Adobe 的四套字体：宋体、仿宋、黑体、楷体。在 `CJK` 模式（即不使用  $\text{\XeTeX}$  时）下，该选项将使用 `winfont`s 选项的设置。

**zhmap** 仅在 `winfont`s 模式下有效。使用 `zhmetrics` 宏包提供的字体文件映射，将中文字库映射到相应的 `ttf` 字库文件。这个是 `ctex` 宏包的缺省模式。

**nozhmap** 仅在 `winfont`s 模式下有效。使用系统提供的字体文件映射方式。如果需要使用自定义的字体映射或者使用 `Type1` 字库，请使用该选项。

### 2.3.5 CCT 引擎选项

下面的选项用于选择底层的中文系统。缺省情况下，宏包会根据编译方式自动选择 `CJK` 或者 `xeCJK` 引擎。

**CCT** 使用 `CCT` 代替 `CJK` 做为底层的中文支持系统。

**CCTfont** 使用传统的 `CCT` 字库方式，该选项会自动激活 `CCT` 选项。

### 2.3.6 排版风格选项

**cap** 使用中文的标题样式，缺省格式由 `ctexcap.cfg` 配置文件内的定义给出。对于 `ctex.sty`，该选项只影响交叉引用中的数字和日期格式。这个是 `ctex` 宏包的缺省模式。

**nocap** 保留使用英文的标题样式。

**punct** 对中文标点的位置（宽度）进行调整。这个是 `ctex` 宏包的缺省模式。

**nopunct** 不对中文标点的位置进行调整（每个标点占有相同的宽度）。

**space** 使用 `CJK` 的保留空格模式，保留中文字符间的空格（类似英文的习惯）。你需要自己处理中文字符间的空格以及换行产生的空格（在行尾加上 `%` 符号可以避免），否则排版结果可能不符合中文习惯。这种模式可以通过 `\CTEXnospace` 转换到 `nospace` 模式。

**nospace** 使用 CJK 的忽略空格模式，也就是 CJK\* 环境的模式。CJK 会自动忽略中文字符间的空格，比较符合中文习惯。在这种模式下，可以使用 ~ 来分隔中英文字符，产生的间距稍小于普通空格，排版效果比较美观。这种模式可以通过 `\CTEXspace` 命令转换到 `space` 模式。这个是 `ctex` 宏包的缺省模式。

**indent** 使用中文的段首缩进模式，即缩进两个汉字宽度，同时每个段落都缩进。这个是 `ctex` 宏包的缺省模式。

**noindent** 使用原来的段首缩进模式，章节标题后的第一段不缩进。

### 2.3.7 宏包兼容选项

**fancyhdr** 保持和 `fancyhdr` 宏包的兼容性。该选项将使得 `fancyhdr` 宏包被自动调用。

**hyperref** 自动判断调用 `fancyhdr` 宏包的正确参数以避免产生乱码。该选项将使得 `hyperref` 宏包被自动调用。

**fntef** 为 `CJKfntef` 宏包和 `CCTfntef` 宏包提供统一接口。该选项将使得 `CJKfntef` 宏包或者 `CCTfntef` 宏包被自动调用。

### 2.3.8 缺省选项

总结一下：`ctex` 宏包的缺省选项是 `GBK zhmap nospace cap punct indent`，`ctex` 文档类的缺省选项是 `GBK zhmap nospace cap punct indent c5size`。

## 2.4 基本命令

`ctex` 宏包给用户提供一个通用的文档框架，使得用户可以自由地在不同的底层中文系统间切换。为此，我们为 CJK 定制了一些模拟 CCT 的命令，也对部分 CCT 命令进行了修改，使得两者保持一致。此外，我们还定义了用于设置文档参数的高级设置命令。

### 2.4.1 字体设置

中文字体很多，但是常用的就那么几个。我们为 CJK 常用的六种中文字体定义了简单易用的命令。它们是：

<code>\songti</code>	宋体： <code>\songti</code> ，CJK 等价命令 <code>\CJKfamily{song}</code>
<code>\heiti</code>	黑体： <code>\heiti</code> ，CJK 等价命令 <code>\CJKfamily{hei}</code>
<code>\fangsong</code>	仿宋： <code>\fangsong</code> ，CJK 等价命令 <code>\CJKfamily{fs}</code>
<code>\kaishu</code>	楷书： <code>\kaishu</code> ，CJK 等价命令 <code>\CJKfamily{kai}</code>
<code>\lishu</code>	隶书： <code>\lishu</code> ，CJK 等价命令 <code>\CJKfamily{li}</code>
<code>\youyuan</code>	幼圆： <code>\youyuan</code> ，CJK 等价命令 <code>\CJKfamily{you}</code>

TeX 系统中必须已经定义好这六种中文字体，并且使用和 CTeX 套装中一致的字体名称。（参见上面 CJK 等价命令的参数）

可用的字体命令还取决于使用的中文字库选项，参见 2.3.4 一节的介绍。

上面的字体命令和 CCT 中的一致，但传统的 CCT 字库中没有隶书和仿宋两种字体，需要用户自行安装定义。如果使用 CCT 时选择 CJK 字库方式，则可以使用这两种中文字体。

上面的字体在不同的字体选项下有不同的设置，不一定都有定义。

2.4.2 字号、字距、字宽和缩进

`\zihao` 中文字号的设置命令是 `\zihao{< 字号>}`，例如 `\zihao{3}`。可以使用的参数有 16 个，小号字体在前面加负号表示，从大到小依次为

初号	小初	一号	小一	二号	小二	三号	小三
0	-0	1	-1	2	-2	3	-3
四号	小四	五号	小五	六号	小六	七号	八号
4	-4	5	-5	6	-6	7	8

英文字体大小会始终保持和中文字体一致。

`\ziju` 汉字字距的调整使用命令 `\ziju{< 字宽的倍数>}`。参数可以是任意的数字，例如 `\ziju{5}` 设置汉字字距为当前汉字字宽的 5 倍，`\ziju{0.5}` 设置汉字字距为当前汉字字宽的一半。这里的汉字字宽指的是实际汉字的宽度，不包含当前字距。该命令不影响英文字距。

`\ccwd` 当前汉字的字宽保存在宏 `\ccwd` 中。字宽是相邻两个汉字中心的距离，也就是说字距会被计算在内。

`\CTEXindent` 正常的缩进两个汉字字宽的距离，同时在汉字大小和字距改变的情况都可以自动修改缩进距离。

`\CTEXnoindent` 取消缩进。

`\CTEXsetfont` `\CTEXsetfont` 命令用于更新当前的中文字体信息，包括当前字距和缩进距离。一般来说，用户无需使用这个命令。

2.4.3 中文数字转换

`\CTEXnumber` 使用 CJK 提供的 `\CJKnumber` 命令可以将阿拉伯数字转换为中文数字。由于 L<sup>A</sup>T<sub>E</sub>X 臭名昭著的脆弱命令的原因，当 `\CJKnumber` 被用在章节标题等地方时，要么出现错误无法使用，要么无法达到预期目的，例如在产生 PDF 书签的时候。于是我们定义了一个 `\CTEXnumber` 命令，可以将产生的中文数字保存下来。该命令的格式为

`\CTEXnumber{<result>}{<number>}`

其中 `<result>` 必须是一个 T<sub>E</sub>X 宏的名字，不需要预先定义。例如

`\CTEXnumber{\test}{100002005}`

则 `\test` 中的内容就是“一亿零二千零五”（不包括引号）。

`\CTEXdigits`      `\CTEXdigits` 命令和 `\CTEXnumber` 命令类似，用于代替 CJK 提供的 `\CJKdigits` 命令。它和 `\CTEXnumber` 命令的不同之处在于转换后结果是中文数字串，而不是按照中文习惯的数字。该命令的格式为

```
\CTEXdigits{<result>}{<number>}
```

其中 `<result>` 必须是一个  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  宏的名字，不需要预先定义。例如

```
\CTEXdigits{\test}{100002005}
```

则 `\test` 中的内容就是“一〇〇〇〇二〇〇五”（不包括引号）。

`\chinese`      对于经常需要转换的计数器，我们特别定义了一个 `\chinese` 命令。该命令可以象罗马数字转换命令 `\roman`、`\Roman` 一样使用。具体格式是

```
\chinese{<counter>}
```

其中 `<counter>` 是一个  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  计数器 (counter)，即由 `\newcounter` 命令产生的，例如 `section`、`figure` 等。

`\Chinese`      `ctex` 宏包会在每次使用 `\setcounter`、`\stepcounter` 或 `\addtocounter` 时利用 `\CTEXcounter{<counter>}` 更新 `\chinese` 命令产生的汉字，如果计数器的修改没有用到上述命令（如页码），就需要在 `\chinese` 前手工使用 `\CTEXcounter` 命令更新。为此，`ctex` 宏包提供了大写的 `\Chinese` 命令，作为上述过程的简写。例如 `\Chinese{page}` 产生“八”。该命令不宜用在 `\section` 等命令的参数中。

## 2.5 高级设置

`\CTEXoptions`      `ctex` 宏包中一般的设置通过 `\CTEXoptions` 命令完成。这个命令的基本格式是

```
\CTEXoptions[<key1>=<val1>, <key2>=<val2>, ...]
```

其中 `<key1>`、`<key2>` 是设置选项，`<val1>`、`<val2>` 则是对应选项的设置内容。多个选项可以在一个语句中完成设置。

`\CTEXsetup`      部分设置如章节标题则通过 `\CTEXsetup` 命令完成。这个命令比 `\CTEXoptions` 多一个参数，用于指定设置对象。基本格式是

```
\CTEXsetup[<key1>=<val1>, <key2>=<val2>, ...]{<type>}
```

其中 `<type>` 是设置的对象类型，如 `part`、`chapter`、`section`、`subsection`、`subsubsection`、`paragraph`、`subparagraph` 等。`<key1>`、`<key2>` 是设置选项，如 `name`、`number`、`format`、`nameformat`、`numberformat`、`aftername`、`titleformat` 等。`<val1>`、`<val2>` 则是对应选项的设置内容。同一个目标类型的多个选项可以在一个语句中完成设置。

在 v0.7 版本之前，如果以上命令的参数中包含中文字符，则命令必须放在 `\begin{document}` 之后才能正常工作。从 v0.7 版本开始支持在导言区使用中文。



### 2.5.1 章节标题设置

普通章节标题的格式全部通过 `\CTEXsetup` 命令完成。章节类型在 `\CTEXsetup` 命令的第二个参数中指定。

在 v0.7 版本之前, 如果使用了宏包选项 `cap` (缺省情况即是如此), 则所有对章节标题的修改必须在 `\begin{document}` 以后进行。原因是缺省的中文标题设置文件 `ctexcap.cfg` 文件是在 `\begin{document}` 之后才会自动装入, 因而之前的修改都会被覆盖而无效。这一限制对后面的附录标题以及其他标题设置一样有效。从 v0.7 版本开始, `ctexcap.cfg` 文件在宏包文件结束时就已经被装入, 因此可以在导言区使用设置命令。

**name**={*(pname)*,*(postname)*} 该选项用于设置章节的名字, 包括章节编号前后的词语, 两个之间用逗号分开。例如

```
\CTEXsetup[name={第, 节}]{section}
```

会使得 `section` 的标题使用形如“第 1 节”的名字。注意不要使用中文的逗号。

该选项的缺省设置是

	使用宏包选项 <code>cap</code>	使用宏包选项 <code>nocap</code>
<code>part</code>	{ 第, 部分 }	{Part\space,}
<code>chapter</code>	{ 第, 章 }	{Chapter\space,}
<code>section</code>	同右	{,}
<code>subsection</code>	同右	{,}
<code>subsubsection</code>	同右	{,}
<code>paragraph</code>	同右	{,}
<code>subparagraph</code>	同右	{,}

**number**={*(number)*} 该选项用于设置章节编号的数字样式。例如

```
\CTEXsetup[number={\roman{section}}]{section}
```

会使得 `section` 的标题使用小写罗马数字作为编号。常用的数字样式命令有

```
\chinese{<counter>}: 一, 二, 三, ...
```

```
\arabic{<counter>}: 1, 2, 3, ...
```

```
\roman{<counter>}: i, ii, iii, ...
```

```
\Roman{<counter>}: I, II, III, ...
```

```
\alph{<counter>}: a, b, c, ...
```

```
\Alph{<counter>}: A, B, C, ...
```

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part	<code>{\chinese{part}}</code>	<code>{\Roman{part}}</code>
chapter	<code>{\chinese{chapter}}</code>	<code>{\arabic{chapter}}</code>
section	同右	<code>{\thesection}</code>
subsection	同右	<code>{\thesubsection}</code>
subsubsection	同右	<code>{\thesubsubsection}</code>
paragraph	同右	<code>{\theparagraph}</code>
subparagraph	同右	<code>{\thesubparagraph}</code>

**format**=`{\format}` 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。常用于控制章节标题的对齐方式。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	<code>{\centering}</code>	<code>{\raggedright}</code>
part	<code>{\centering}</code>	<code>{\centering}</code>
chapter	<code>{\centering}</code>	<code>{\raggedright}</code>
section	<code>{\Large\bfseries\centering}</code>	<code>{\Large\bfseries}</code>
subsection	<code>{\large\bfseries\centering}</code>	<code>{\large\bfseries}</code>
subsubsection	<code>{\normalsize\bfseries\centering}</code>	<code>{\normalsize\bfseries}</code>
paragraph	<code>{\normalsize\bfseries\centering}</code>	<code>{\normalsize\bfseries}</code>
subparagraph	<code>{\normalsize\bfseries\centering}</code>	<code>{\normalsize\bfseries}</code>

**nameformat**=`{\nameformat}` 用于控制章节名字的格式，作用域为章节名字，包括编号。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	同右	<code>{\Large\bfseries}</code>
part	同右	<code>{\huge\bfseries}</code>
chapter	同右	<code>{\huge\bfseries}</code>
section	同右	<code>{}</code>
subsection	同右	<code>{}</code>
subsubsection	同右	<code>{}</code>
paragraph	同右	<code>{}</code>
subparagraph	同右	<code>{}</code>

**numberformat**=`{\numberformat}` 用于控制章节编号的格式。一般为空，当你需要编号的格式和前后的章节名字不一样时使用。

**aftername**=`{\aftername}` 用于控制章节标题中章节名字和随后的标题内容之间的格式变换。常用于控制标题内容是否另起一行。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	<code>\quad</code>	<code>\par\nobreak</code>
part	同右	<code>\par\vskip 20pt</code>
chapter	<code>\quad</code>	<code>\par\vskip 20pt</code>
section	同右	<code>{}</code>
subsection	同右	<code>{}</code>
subsubsection	同右	<code>{}</code>
paragraph	同右	<code>{}</code>
subparagraph	同右	<code>{}</code>

**titleformat**=`{\langle titleformat \rangle}` 用于控制标题内容的格式，作用域为章节标题内容。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	<code>\Large\bfseries</code>	<code>\huge\bfseries</code>
part	<code>\huge\bfseries</code>	<code>\Huge\bfseries</code>
chapter	<code>\huge\bfseries</code>	<code>\Huge\bfseries</code>
section	同右	<code>{}</code>
subsection	同右	<code>{}</code>
subsubsection	同右	<code>{}</code>
paragraph	同右	<code>{}</code>
subparagraph	同右	<code>{}</code>

**beforeskip**=`{\langle beforeskip \rangle}` 用于控制章节标题前的空距。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	同右	<code>{4ex}</code>
part	无效	无效
chapter	同右	<code>{50pt}</code>
section	同右	<code>{-3.5ex plus -1ex minus -.2ex}</code>
subsection	同右	<code>{-3.25ex plus -1ex minus -.2ex}</code>
subsubsection	同右	<code>{-3.25ex plus -1ex minus -.2ex}</code>
paragraph	同右	<code>{3.25ex plus 1ex minus .2ex}</code>
subparagraph	同右	<code>{3.25ex plus 1ex minus .2ex}</code>

在 section 及以下的标题中，使用负的距离表示标题后的段落不缩进（如标准的英文 LaTeX 文档），否则缩进。标题上方真正的空距是该参数的绝对值。

**afterskip**=`{\langle afterskip \rangle}` 用于控制章节标题后的空距。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	同右	{3ex}
part	无效	无效
chapter	同右	{40pt}
section	同右	{2.3ex plus .2ex}
subsection	同右	{1.5ex plus .2ex}
subsubsection	同右	{1.5ex plus .2ex}
paragraph	同右	{-1em}
subparagraph	同右	{-1em}

在 section 及以下的标题中，正的距离表示向下留出的空距（如标准的 section 标题），使用负的距离则表示向右留出的空距的负值（如标准的 paragraph 标题）。

**indent**={*indent*} 用于控制章节标题本身的缩进。

该选项的缺省设置是

	使用宏包选项 cap	使用宏包选项 nocap
part (article)	同右	{0pt}
part	无效	无效
chapter	同右	{0pt}
section	同右	{0pt}
subsection	同右	{0pt}
subsubsection	同右	{0pt}
paragraph	同右	{0pt}
subparagraph	同右	{\parindent}

### 2.5.2 部分修改标题格式

如果只想修改标题格式中的某些参数而不是完全重新设置，可以使用带 + 号的设置选项。例如

```
\CTEXsetup[format+={\zihao{1}}]{section}
```

则 section 的标题使用一号字体，而其他格式设置保持不变。

标题格式相关的选项都支持这一功能，包括 `format`、`nameformat`、`numberformat`、`aftername` 和 `titleformat`，而且对所有文档类型都有效。

### 2.5.3 附录标题设置

附录 (appendix) 的标题也使用 `\CTEXsetup` 命令进行设置，第二个参数设为 `appendix`。但是只能使用 `name` 和 `number` 两个设置选项。在使用了 `\appendix` 命令之后，附录的名字和编号会被自动使用。附录的名字和前面的章节不同，它只有一个部分，放在编号之前。在 article 类文档中，附录是用 section 实现的，而在 report 和 book 类文档

中附录使用的是 `chapter` 的设置。因此在设置附录的编号的时候要注意使用正确的计数器。如果你要设置其他格式的附录标题，可以根据使用的文档类直接用 `section` 或者 `chapter` 的设置命令来控制，但是要记住把设置命令放在 `\appendix`（如果有的话）的后面，否则会被 `\appendix` 命令的设置覆盖。

附录的缺省设置是

	使用宏包选项 <code>cap</code>	使用宏包选项 <code>nocap</code>
<code>name (article)</code>	同右	<code>{}</code>
<code>name</code>	<code>{ 附录~}</code>	<code>{Appendix\space}</code>
<code>number (article)</code>	同右	<code>{\Alph{section}}</code>
<code>number</code>	同右	<code>{\Alph{chapter}}</code>

#### 2.5.4 其他标题设置

除章节标题外其他标题的设置通过 `\CTEXoptions` 设置。包括

`contentsname` 目录名

`listfigurename` 表格目录

`listtablename` 插图目录

`figurename` 图

`tablename` 表

`abstractname` 摘要

`indexname` 索引

`bibname` 参考文献

例如

```
\CTEXoptions[indexname={总索引}]
```

把索引的名字改为“总索引”。

#### 2.5.5 其他设置

**设置 `\today` 的日期格式** 使用 `\CTEXoptions` 可以设置 `\today` 命令产生的日期格式。支持的格式包括

1. 阿拉伯数字加中文年月日

```
\CTEXoptions[today=small]
```

`\today` 生成的日期例子为“2010 年 7 月 12 日”。

## 2. 中文数字加中文年月日

```
\CTEXoptions[today=big]
```

`\today` 生成的日期例子为“二〇一〇年七月十二日”。

## 3. L<sup>A</sup>T<sub>E</sub>X 标准格式

```
\CTEXoptions[today=old]
```

`\today` 生成的日期例子为“July 12, 2010”。

**设置图表标题的分隔符** 使用 `\CTEXoptions` 可以设置 `\caption` 命令产生的图表标题的分隔符。这个分隔符缺省是使用冒号:。可以通过命令

```
\CTEXoptions[captiondelimiter={\langle string \rangle}]
```

设置为任意的单个字符或者字符串 `\langle string \rangle`。

## 2.6 配置文件

主要的配置文件有:

- `ctexopts.cfg`
- `ctexcap.cfg`, `ctexcap-gbk.cfg`, `ctexcap-utf8.cfg`
- `ctex.cfg`

`ctexopts.cfg` 用于设置缺省选项。例如可以根据系统中的可用中文字库, 设置 2.3.4 一节中的几个选项之一为缺省选项。该文件会在处理宏包选项之前装入。

`ctexcap.cfg` 是缺省中文标题格式的定义, 当你使用 `cap` 选项时就会使用这里的定义。你可以把它改为你经常使用的格式, 这样就不用每次都在正文中修改了。`ctexcap.cfg` 中的设置都可以通过宏包提供的设置命令在正文中进行修改。与中文编码有关的定义分别在 `ctexcap-gbk.cfg` 和 `ctexcap-utf8.cfg` 文件中。

最后, 宏包还将读入 `ctex.cfg` 文件, 该文件中的设置将覆盖其他配置文件中的设置。用户可以在该文件中加入自己的定义。

在修改这些配置文件的时候, 你可以修改系统目录中的文件, 也可以拷贝一份放到当前目录下, 然后修改。TeX 会优先使用当前目录下的同名文件。这样你可以针对不同的应用设置不同的缺省配置文件。

### 3 版本更新

- v0.99b 2010/07/11** 发现 BUG，把 `\chinese*` 改为 `\Chinese`。
- v0.99a 2010/07/11** 增加 `\chinese*` 命令，作为 `\CTEXcounter` 和 `\chinese` 混合体的简写形式。
- v0.99 2010/07/08** 提早引入 `expl3` 宏包，解决 `\chinese` 命令在 `expl3` 下的不能正确使用的问题。
- v0.98 2010/06/19** 解决 `xeCJK` 默认代替 `indentfirst` 功能的问题。修正 `xeCJK` 模式下 `winfonts` 的中文字体。
- v0.97 2010/01/22** 修正 `pdfLATEX` 与 `XeLATEX` 中生成 PDF 书签的问题。
- v0.96 2009/11/24** 添加 `zhmap` 与 `nozhmap` 选项控制 `zhwinfonts` 的载入。
- v0.95 2009/10/20** 移除对 `CJKnumb` 宏包的依赖。去除无用的 `cjkfonts` 选项，以 `winfonts` 作为默认值。
- v0.94 2009/09/13** 改进 `hyperref` 选项的支持
- v0.93 2009/07/10** 加入选项 `hyperref` 以支持自动用合适的参数调用 `hyperref` 宏包
- v0.92 2009/06/25** Add backward compatibility to old `ctexutf8` packages
- v0.91 2009/05/23** 调整宏包结构，增加对各种系统中文字库的支持选项：`cjkfonts`, `winfonts`, `adobefonts`
- v0.9 2009/05/05** 整合 `ctexutf8` 宏包，增加对 UTF-8 编码的支持；开始支持 `XeTEX` 中文系统（使用 `xeCJK` 宏包）
- v0.8a 2007/05/06** 增加 `bold` 字体的定义
- v0.8 2006/06/09** 将 `ctex.sty` 文件分割为 `ctex.sty` 和 `ctexcap.sty`，后者只支持标准文档类增加对 `\stepcounter` 的重定义，以和 `calc` 宏包兼容
- v0.7f 2006/04/12** 采用修改 `\AtBeginDocument` 和 `\AtEndDocument` 命令的方式来设置 CJK 环境，以减少宏包冲突
- v0.7e 2006/03/22** 改用 `\DeclareRobustCommand` 定义 `\CTEXnumber` 和 `\CTEXcounter`；除去 `\CTEXdigits` 和 `\CTEX@getdigit` 命令带来的多余空格
- v0.7d 2005/12/28** 在 `fntef` 类宏包后使用 `\normalem` 恢复 `\em` 宏的缺省定义
- v0.7c 2005/12/20** 增加对 `\if@mainmatter` 的判断，以兼容 `amsbook` 宏包
- v0.7b 2005/12/09** 调整宏包导入位置，解决 `fntef` 类宏包早于相应中文宏包导入的问题
- v0.7a 2005/11/28** 将 `ctex.cfg` 文件的读取时间前移，使得导言中的设置命令优先

- v0.7 2005/11/25** 支持在导言区中使用中文和章节标题设置命令（感谢 tercelxy 的建议）；增加 CJKfntef 宏包和 CCTfntef 宏包的统一接口（感谢 chenyu.21cn 的建议）
- v0.6b 2005/11/07** 将节以下编号和标题之间的空距定义转移到相应的 aftername 变量中
- v0.6a 2005/09/30** 增加对 \CCT@set@fontsize 的判断
- v0.6 2005/09/24** 针对 cct 0.6180 的修改，\set@fontsize: cct 从 0.6180 开始将宏 \oset@fontsize 改为 \CCT@set@fontsize
- v0.5c 2004/09/29** 避免重复执行设置 CJK 环境结束语句
- v0.5b 2004/09/29** 改变设置 CJK 环境结束语句的 \AtEndDocument 执行的位置，以减少宏包冲突
- v0.5a 2004/09/06** 修改图表标题分隔符设置中的错误
- v0.5 2004/08/23** General: Move Chinese definitions from ctex.cfg to ctex.def
- v0.4d 2004/08/14** \ps@fancy: 增加对 mainmatter 的判断；\refstepcounter: 修改 \ref 命令，不再包含除编号外的内容
- v0.4c 2004/07/26** 增加判断以避免嵌套定义 \setcounter 和 \addtocounter
- v0.4b 2004/07/13** \baselinestretch: 把 \baselinestretch 从 1.2 改为 1.3
- v0.4a 2004/05/15** \CTEXdigits: 增加 \CTEXdigits 命令；\ziju: 修改 CCT 的字距命令使得缩进保持一致
- v0.4 2004/05/13** General: 如果指定了标准的 L<sup>A</sup>T<sub>E</sub>X 字体大小，则不使用中文字号；中文字号定义改为直接使用 pt 为单位；\zihao: 删除 \CTEX@fontsize 命令，改为直接使用 \fontsize 命令
- v0.3b 2004/05/11** General: 增加 fancyhdr 选项
- v0.3a 2004/04/30** General: 修改命令 \CCTpuncttrue 的拼写错误
- v0.3 2004/04/24** General: 对页眉设置进行微调；对中文标题的章节编号格式进行调整，去掉 \S；修改为使用 \chinese 命令以避免产生错误；修正 sub3section 和 sub4section 选项无效的问题；增加对图表标题分隔符的设置；\ps@fancy: 解决与 fancyhdr 的冲突
- v0.2d 2004/04/23** General: Change option c5size to base on 10pt basic class; 补上字号定义中行间距参数中缺少的 \CTEX@bp; 修改缺省的字号大小
- v0.2c 2004/02/13** General: Add CJKpunct as standard configuration; \ifCTEX@punct: 增加判断是否调整中文标点宽度的选项
- v0.2b 2004/02/13** General: 修改缺省的行距；修改缺省的字号大小



- v0.2a 2004/02/11** `\baselinestretch`: 增加对行距的设置; `\CTEX@spaceChar`: 加快处理速度, 改善和 CJKpunct 的兼容性
- v0.2 2004/01/16** General: Add support for CCT; 增加部分修改标题格式设置的选项; 增加修改标题前后空距设置的选项; `\CTEXsetfont`: `\CTEXfontinfo` 命令改为 `\CTEXsetfont`; `\ziju`: 参数的单位由绝对距离改为相对于当前汉字大小的倍数
- v0.1f 2003/12/24** `\refname`: 修正 article 类中参考文献标题没有使用中文的问题
- v0.1e 2003/11/05** `\refstepcounter`: 修正 `\ref` 命令后多出空格的问题
- v0.1d 2003/09/27** `\addtocounter`: 将对 `\setcounter` 和 `\addtocounter` 的修改放到导言的最后以和其他宏包兼容
- v0.1c 2003/08/19** General: 去掉生成的.out 文件里章的标题前的多余空格
- v0.1b 2003/08/17** `\zihao`: 删除多余的 `\newcount` 命令
- v0.1a 2003/08/15** General: 修正 ctex.sty 中无法使用 sub3section 和 sub4section 选项的问题
- v0.1 2003/08/15** General: First beta release
- v0.0 2003/04/26** General: Initial version

## 4 开发人员

- 吴凌云 ([aloft@ctex.org](mailto:aloft@ctex.org))
- 江疆 ([gzzjgod@gmail.com](mailto:gzzjgod@gmail.com))
- 王越 ([yuleopen@gmail.com](mailto:yuleopen@gmail.com))
- 刘海洋 ([LeoLiu.PKU@gmail.com](mailto:LeoLiu.PKU@gmail.com))
- LiYanrui.m2 (LiYanrui)
- 陈之初 (Neals)