

xeCJK 宏包

ctex.org

2013/07/06 3.2.4*

目录

1	简介	1	5.4	字符类别设定	19
2	基本用法	2	5.5	字符类别处理	24
3	用户手册	2	5.6	字符输出规则	27
3.1	宏包选项	2	5.7	全角右标点后的断行	37
3.2	字体设置与选择	5	5.8	段末孤字处理	38
3.2.1	X _Y TeX 的字体名查找	7	5.9	增加 CJK 子分区	42
3.3	CJK 分区字体设置	8	5.10	标点处理	44
3.4	设置 CJK 字符范围	9	5.11	后备字体	54
3.5	标点符号的处理	9	5.12	CJK 字体族声明方式	55
3.5.1	设置特定标点符号的宽度和间距	10	5.13	字体切换	63
3.5.2	定义标点符号处理格式	10	5.14	数学字体设置	70
3.6	其它	12	5.15	抄录环境中的间距调整	71
4	已知问题和兼容性	12	5.16	xeCJK 其它选项	74
5	xeCJK 代码实现	12	5.17	xeCJK 初始化设置	76
5.1	运行环境检查	12	5.18	兼容性修补	78
5.2	内部工具	13	5.19	xeCJKfntef	83
5.3	功能开关	18	5.20	xeCJK-listings	91
			5.21	xeCJK.cfg	97
			版本历史	98	
			代码索引	99	

1 简介

xeCJK 是一个 X_YTeX 宏包,用于排版中日韩(CJK)文字。主要功能:

1. 分别设置 CJK 和英文字体;
2. 自动忽略 CJK 文字间的空格而保留其它空格,允许在非标点汉字和英文字母 (a-z, A-Z) 间断行;
3. 提供多种标点处理方式: 全角式、半角式、开明式、行末半角式和 CCT 式;
4. 自动调整中英文间空白。

xeCJK 使用了 X_YTeX 的一些最新特性,需要 X_YTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 依赖 L^AT_EX3 项目的宏包套件 `l3kernel` 和 `l3packages`。xeCJK 还需要通过 `fontspec` 宏包来调用系统字体。将在 3.1 节介绍的功能选项 `indentfirst` 选项需要 `indentfirst` 宏包的支持。xeCJK 会自动根据需要载入这些宏包。

xeCJK 的原始作者是孙文昌,2009 年 5 月起宏包被收入 `ctex-kit` 项目进行维护,目前主要维护者是刘海洋¹ 和李清²。

* `ctex-kit` rev465.

¹ `leoliu.pku@gmail.com`

² `sobenlee@gmail.com`

2 基本用法

与其他 \LaTeX 宏包一样,引入 `xeCJK` 宏包只要在导言区使用

```
\usepackage{xeCJK}
```

在引入 `xeCJK` 宏包之后,只要设置 `CJK` 文字的字体,就可以在文档中使用中日韩文字了。

可以在各种文档类中使用 `xeCJK` 宏包,最简单的示例是:

```
\documentclass{article}
\usepackage{xeCJK}
\setCJKmainfont{SimSun}

\begin{document}
中文  $\LaTeX$  示例。
\end{document}
```

上述示例设置了中文字体 `SimSun`(宋体)。运行此示例要求系统安装了设置的字体,源文件用 UTF-8 编码保存,使用 \XeLaTeX 编译。

`xeCJK` 只提供了字体和标点控制等基本 `CJK` 语言支持。对于中文文档,可以使用更为高层的 `ctex` 宏包或文档类,它将自动调用 `xeCJK` 并设置好中文字体,同时提供了进一步的本地化支持。详细内容参看 `ctex` 宏包套件的说明。

`xeCJK` 提供了大量选项,可以在宏包调用时作为宏包选项或用 `\xeCJKsetup` 命令进行设置,详见 3.1 节。除了 `\setCJKmainfont` 命令,`xeCJK` 还提供了许多其他命令设置和选择中文字体,详见 3.2 节。其他更详细的功能也都将在下面详细说明。在本文档所在的文件夹的 `example` 目录下也有一些例子可以参考。

3 用户手册

3.1 宏包选项

`xeCJK` 以 $\langle key \rangle = \langle var \rangle$ 的形式提供宏包选项,你可以在调用宏包的时候直接设置这些选项,也可以在调用宏包之后使用 `\xeCJKsetup` 来设置这些选项。`xeCJK` 内部调用 `fontspec` 宏包,可以在调用 `xeCJK` 的时候,使用它的宏包选项。`xeCJK` 会将 `fontspec` 的选项传递给它。

```
\xeCJKsetup { $\langle key_1 \rangle = \langle var_1 \rangle$ ,  $\langle key_2 \rangle = \langle var_2 \rangle$ , ...}
```

其中 $\langle key_1 \rangle$, $\langle key_2 \rangle$ 是设置选项,而 $\langle val_1 \rangle$, $\langle val_2 \rangle$ 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。例如

```
\usepackage[PunctStyle=kaiming]{xeCJK}
```

等价于

```
\usepackage{xeCJK}
.....
\xeCJKsetup{PunctStyle=kaiming}
```

有些选项或命令后面带有 \star 号,这表示这个选项或命令只能在导言区中使用,而 \star 号则表示这个选项或命令只能在导言区使用,并且只影响随后定义的 `CJK` 字体。其余不带特殊标记的选项或命令,如果没有特别说明,可以在导言区或正文中使用。使用粗体来表示 `xeCJK` 的默认设置。

```
LocalConfig  $\star$  LocalConfig = {\true|false|name}}
```

New: 2012-11-22

是否使用本地配置文件 `xeCJK- $\langle name \rangle$.cfg`。 $\langle name \rangle$ 可以是不包含空格的任意使文件名合法的字符串。如果设置为 `true`,则使用的是 `xeCJK.cfg`; 设置为 `false` 则不载入配置文件。可以把将要在下文介绍到的对 `xeCJK` 的一些设置(例如设置常用 `CJK` 字体、修改字符范围和定义新的标点输出格式等)保存到文件 `xeCJK- $\langle name \rangle$.cfg`。然后把这个文件放在本地的 TDS 目录下的适当位置。使用 \TeX Live 的用户,可以新建下列目录,然后再把 `xeCJK- $\langle name \rangle$.cfg` 放在里面:

```
texlive/texmf-local/tex/xelatex/xecjk
```

最后还需要在命令行下执行 `mktextlsr`,刷新文件名数据库以便 \TeX 系统能够找到它。

indentfirst ★	<code>indentfirst = {true false}</code>
Updated: 2012-11-22	是否使用 <code>indentfirst</code> 宏包,使得跟在章节标题后面的第一段首行也缩进。
<p>请注意,<code>xeCJK</code> 宏包中只有上述 <code>LocalConfig</code> 和 <code>indentfirst</code> 这两个选项需要在调用 <code>xeCJK</code> 时设置,而不能通过 <code>\xeCJKsetup</code> 来设置。</p>	
xeCJKactive	<code>xeCJKactive = {true false}</code>
	打开/关闭对中文的特殊处理。事实上,这个选项会打开/关闭 \XeTeX 的整个字符类机制,依赖这个机制的宏包都会受到影响。
CJKspace	<code>CJKspace = {true false}</code>
	缺省状态下, <code>xeCJK</code> 会忽略 CJK 文字之间的空格,使用这一选项来保留它们之间的空格。
CJKmath ★	<code>CJKmath = {true false}</code>
	是否支持在数学环境中直接输入 CJK 字符。使用这个选项后,可以直接在数学环境中输出 CJK 字符。
CJKglue	<code>CJKglue = {\hskip 0pt plus 0.08\baselineskip}</code>
	设置 CJK 文字之间插入的 <code>glue</code> ,上边是 <code>xeCJK</code> 的默认值。一般来说,除非有特殊需要(例如,改变文字间距等),否则不需要设置这个选项,使用默认值即可。如果要设置这个选项,为了行末的对齐,设置的 <code>glue</code> 最好有一定的弹性。
CJKecglue	<code>CJKecglue = {glue}</code>
	设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距,默认值是一个空格。使用这个选项设置的 <code>glue</code> 最好也要用一定的弹性。请注意,这里设置的 <code>glue</code> 只影响 <code>xeCJK</code> 根据需要自动添加的空白,源文件中直接输入的 CJK 文字与西文之间的空格不受影响(直接输出)。有时候 <code>xeCJK</code> 可能不能正确地调整间距,需要手动加空格。
xCJKecglue	<code>xCJKecglue = {true false glue}</code>
	缺省状态下, <code>xeCJK</code> 不对源文件中直接输入的 CJK 文字与西文之间的空格进行调整,如果需要调整,请使用这个选项。如果使用这个选项,将使用 <code>CJKecglue</code> 替换源文件中直接输入的 CJK 文字与西文之间的空格。
CheckSingle	<code>CheckSingle = {true false}</code>
Updated: 2013-06-26	是否避免单个 CJK 文字单独占一个段落的最后一行。需要说明的是,这个选项只有在段末的最后一个字是 CJK 文字或者标点符号,并且倒数第二和第三个字都是文字才能正确处理孤字的问题。如果这倒数三个字有作为控制序列的参数情况,那么一般来说也不能正确处理。
PlainEquation	<code>PlainEquation = {true false}</code>
New: 2012-12-06	如果使用了 <code>\$\$...\$\$</code> 的形式来输入行间数学公式,就需要启用本选项,以便 <code>CheckSingle</code> 选项能够正确识别。推荐使用 <code>\[...\]</code> 的形式来输入行间数学公式。
NewLineCS NewLineCS+ NewLineCS-	<code>NewLineCS = { \par \[}</code>
New: 2012-12-04	设置造成断行的控制序列,以便 <code>CheckSingle</code> 选项能够正确识别。以上是 <code>xeCJK</code> 的初始设置。
EnvCS EnvCS+ EnvCS-	<code>EnvCS = { \begin \end }</code>
New: 2012-12-04	设置 \LaTeX 环境开始和结束的控制序列,以便 <code>CheckSingle</code> 选项能够正确识别。以上是 <code>xeCJK</code> 的初始设置。

<hr/> InlineEnv	InlineEnv = {⟨env ₁ ⟩, ⟨env ₂ ⟩, ⟨env ₃ ⟩, ...}
InlineEnv+	在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 LaTeX 环境的开始 <code>\begin{...}</code> 和结束 <code>\end{...}</code> 视为断行的地方,如果有某些特殊的 LaTeX 环境没有造成断行,可以使用这个选项来声明它,以便 CheckSingle 能正确识别。
InlineEnv-	
Updated: 2012-12-06	
<hr/> AutoFallBack	AutoFallBack = {⟨true false⟩}
	当文档中有个别生僻字时,可以使用这个选项,自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。
<hr/> AutoFakeBold ☆	AutoFakeBold = {⟨true false 数字⟩}
	全局设定当没有声明对应的粗体时,是否使用 伪粗体 ;当输入的是数字时,将使用伪粗体,并将使用输入的数字作为伪粗体的默认粗细程度。
<hr/> AutoFakeSlant ☆	AutoFakeSlant = {⟨true false 数字⟩}
	全局设定当没有声明对应的斜体时,是否使用 伪斜体 ;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。
<hr/> EmboldenFactor ☆	EmboldenFactor = {⟨数字 4⟩}
	设置伪粗体的默认粗细程度。
<hr/> SlantFactor ☆	SlantFactor = {⟨数字 0.167⟩}
	设置伪斜体的粗细程度,范围是 -0.999 ~ 0.999。
<hr/> PunctStyle	PunctStyle = {⟨quanjiao banjiao kaiming hangmobanjiao CCT plain ...⟩}
Updated: 2012-11-10	设置标点处理格式。xeCJK 中预先定义好的格式为
	<div> <div>quanjiao</div> <div>banjiao</div> <div>kaiming</div> <div>hangmobanjiao</div> <div>CCT</div> <div>plain</div> </div> <div> 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; 半角式: 所有标点占半个汉字宽度; 开明式: 句末点号用全角,其他半角; 行末半角式: 所有标点占一个汉字宽度,行首行末对齐; CCT 格式: 所有标点符号的宽度略小于一个汉字宽度; 原样(不调整标点间距)。 </div>
	可以使用 3.5.2 中介绍的 <code>\xeCJKDeclarePunctStyle</code> 定义新的标点格式。
<hr/> KaiMingPunct ☆	KaiMingPunct = {⟨. . ? !⟩}
KaiMingPunct+	设置开明(kaiming)标点处理格式时的句末点号,KaiMingPunct 后带的 + 与 - 分别表示从已有的开明句末点号中增加或减少标点。
KaiMingPunct-	
<hr/> LongPunct ☆	LongPunct = {⟨—— ……⟩}
LongPunct+	设置长标点,例如破折号“——”与省略号“……”,允许在长标点前后断行,但是禁止在它们之间断行。
LongPunct-	
<hr/> MiddlePunct ☆	MiddlePunct = {⟨—— · · · ⟩}
MiddlePunct+	设置居中显示的标点,例如间隔号“·”。对于在 CJK 文字之间的居中标点,xeCJK 会根据不同的标点处理格式,调整居中标点与前后文字之间的空白,保证其确实居中。对于行末出现的居中标点,允许在其后面断行,但禁止在它前面断行。
MiddlePunct-	
<hr/> PunctWidth ☆	PunctWidth = {⟨length⟩}
	缺省状态下,xeCJK 会根据所选择的标点处理格式自动计算标点所占的宽度,如果对缺省设置不满意,可以通过这一选项来改变它。为了使得标点所占的宽度能够适应字体大小的变化,这里设置的 length 的单位最好用 em 等相对距离单位,而不建议使用诸如 pt 之类的绝对距离单位。这里的设置可用于除了 plain 以外的所有标点处理格式。同时,这里的设置对所有的 CJK 标点都生效,如果只要设置部分标点,请使用 3.5.1 节的 <code>\xeCJKsetwidth</code> 。

AllowBreakBetweenPuncts	<code>AllowBreakBetweenPuncts = <true false></code>
缺省状态下,xeCJK 禁止在相邻 CJK 右标点和 CJK 左标点之间换行,可以使用这一选项改变这一设置。	
CheckFullRight	<code>CheckFullRight = <true false></code>
New: 2012-12-02	某些控制序列要求不能在它的前面断行。但是在缺省状态下,单个全角右标点的后面总是可以断行的。因此当这些控制序列出现在全角右标点后面时,可能会出现意料之外的断行。此时可以使用这个选项来避免这个情况。
NoBreakCS NoBreakCS+ NoBreakCS-	<code>NoBreakCS = { \footnote \footnotemark \nobreak }</code>
New: 2012-12-02	设置不能在全角右标点后断行的控制序列。以上是 xeCJK 的默认设置。如果这些控制序列在文档中只出现少量几次,也可以不必使用 <code>CheckFullRight</code> 选项,而是手工在这些控制序列前面加上 3.6 节介绍的 <code>\xeCJKnobreak</code> 。
Verb	<code>Verb = <true false env env+></code>
Updated: 2013-05-29	<code>true</code> 表示在 <code>\verb</code> 命令或 <code>verbatim</code> 环境里不自动调整中英文之间的间距。 <code>env</code> 选项在 <code>verbatim</code> 环境里自动计算中西文间距和中文之间的间距,以便于保持代码的对齐; <code>env</code> 选项不调整 <code>\verb</code> 里的间距, <code>env+</code> 选项还将正文里设置的间距应用到 <code>\verb</code> 里。这个选项对使用到 <code>\verbatim@font</code> 命令的情形均有效,更一般的情况可以使用 3.6 节介绍的 <code>\xeCJKVerbAddon</code> 。

3.2 字体设置与选择

\setCJKmainfont ★	<code>\setCJKmainfont [] {}</code>
设置正文罗马族的 CJK 字体,影响 <code>\rmfamily</code> 和 <code>\textrm</code> 的字体。后面两个参数继承自 <code>fontspec</code> 宏包, <code></code> 表示字体属性选项, <code></code> 是字体名。字体名可以是字体族名,也可以是字体的文件名,查找字体名见 3.2.1 节;可用的字体属性选项参见 <code>fontspec</code> 宏包的文档。需要说明的是 xeCJK 修改了 <code>AutoFakeBold</code> 和 <code>AutoFakeSlant</code> 选项,以便配合全局伪粗体和伪斜体的设定。	
AutoFakeBold AutoFakeSlant	<code>AutoFakeBold = {<true false 数字>}</code> <code>AutoFakeSlant = {<true false 数字>}</code>
局部设置当前字体族的伪粗和伪斜属性。如果没有在局部给出这些选项,将使用全局设定。	
Mapping	<code>Mapping = {<fullwidth-stop full-stop han-trad han-simp ...>}</code>
New: 2013-06-07	xeCJK 提供了以上四个 TECKit 映射文件,可以在设置字体的时候通过 <code>Mapping</code> 选项来使用它们。其中 <code>fullwidth-stop</code> 用于将正常句号“。”转换成全角实心句号“。”, <code>full-stop</code> 的作用相反。 <code>han-trad</code> 用于将简体中文转换成繁体中文, <code>han-simp</code> 的作用相反。需要注意的是,简繁互换都是简单机械的字字对译,不能做到完全准确,使用时要小心。例如简体的“发挥”和“头发”被转换成繁体的“發揮”和“頭髮”,显然后者应作“頭髮”。也可以根据实际需要,制作新的映射文件,请参考 TECKit 的文档。
\setCJKsansfont ★	<code>\setCJKsansfont [] {}</code>
设置正文无衬线族的 CJK 字体,影响 <code>\sffamily</code> 和 <code>\textsf</code> 的字体。	
\setCJKmonofont ★	<code>\setCJKmonofont [] {}</code>
设置正文等宽族的 CJK 字体,影响 <code>\ttfamily</code> 和 <code>\texttt</code> 的字体。	
\setCJKfamilyfont ★	<code>\setCJKfamilyfont {<family>} [] {}</code>
声明新的 CJK 字体族 <code><family></code> 并指定字体。	

`\CJKfamily`
Updated: 2012-10-27

`\CJKfamily` $\{\langle family \rangle\}$
`\CJKfamily+` $\{\langle family \rangle\}$
`\CJKfamily-` $\{\langle family \rangle\}$

用于在文档中切换 CJK 字体族, $\langle family \rangle$ 必须预先声明。`\CJKfamily` 仅对 CJK 字符类有效, `\CJKfamily+` 对所有字符类均有效, `\CJKfamily-` 对非 CJK 字符类有效。当 `\CJKfamily+` 和 `\CJKfamily-` 的参数为空时, 则使用当前的 CJK 字体族。

`\newCJKfontfamily` ★

`\newCJKfontfamily` [$\langle family \rangle$] $\backslash\langle font-switch \rangle$ [$\langle font features \rangle$] $\{\langle font name \rangle\}$

声明新的 CJK 字体族 $\langle family \rangle$ 并指定字体, 并定义 $\backslash\langle font-switch \rangle$, 在文档中可以使用它来切换 CJK 字体族。可以不必指定 $\langle family \rangle$, 这时候 $\langle family \rangle$ 将等于 $\langle font-switch \rangle$ 。事实上, `\newCJKfontfamily` 是 `\setCJKfamilyfont` 和 `\CJKfamily` 的合并。例如

`\newCJKfontfamily[song]\songti{SimSun}`

等价于

`\setCJKfamilyfont{song}{SimSun}`
`\newcommand*\songti{\CJKfamily{song}}`

`\CJKfontspec`

`\CJKfontspec` [$\langle font features \rangle$] $\{\langle font name \rangle\}$

在文档中随机定义新的 CJK 字体族, 并马上使用它。

`\defaultCJKfontfeatures` ★

`\defaultCJKfontfeatures` $\{\langle font features \rangle\}$

全局设置 CJK 字体族的默认选项。例如, 使用

`\defaultCJKfontfeatures{Scale=0.962216}`

可以将全部 CJK 字体缩小为 0.962216。xeCJK 宏包的初始化设置是

`\defaultCJKfontfeatures{Script=CJK}`

`\addCJKfontfeatures`

Updated: 2013-06-30

`\addCJKfontfeatures` $\{\langle font features \rangle\}$
`\addCJKfontfeatures *` $\{\langle font features \rangle\}$
`\addCJKfontfeatures` [$\langle block_1 \rangle$, $\langle block_2 \rangle$, ...] $\{\langle font features \rangle\}$
`\addCJKfontfeatures *` [$\langle block_1 \rangle$, $\langle block_2 \rangle$, ...] $\{\langle font features \rangle\}$

临时增加当前使用的 CJK 字体的选项。第一条命令, 仅对当前 CJK 主分区字体有效; 第二条对主分区和其它分区的字体都有效; 第三条仅对可选参数中指定的分区有效; 第四条对主分区和可选参数中指定的分区有效。例如, 使用

`\addCJKfontfeatures{Scale=1.1}`

可以将文档中当前使用的 CJK 主分区字体放大为 1.1。

`\CJKrmdefault`

保存 `\textrm` 和 `\rmfamily` 所使用的 CJK 字体族, 默认值是 `rm`。类似西文字体的 `\rmdefault`。

`\CJKsfdefault`

保存 `\textsf` 和 `\sffamily` 所使用的 CJK 字体族, 默认值是 `sf`。类似西文字体的 `\sfdefault`。

`\CJKttdefault`

保存 `\texttt` 和 `\ttfamily` 所使用的 CJK 字体族, 默认值是 `tt`。类似西文字体的 `\ttdefault`。

`\CJKfamilydefault`

Updated: 2013-01-01

保存 `\textnormal` 和 `\normalfont` 所使用的 CJK 字体族。类似西文字体的 `\familydefault`。初始值是 `\CJKrmdefault`。如果没有在导言区中修改它, xecjk 会在导言区结束的时候根据西文字体的情况自动更新 `\CJKfamilydefault`。因此, 在导言区里使用

`\renewcommand\familydefault{\sfdefault}`

就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。

```
\setCJKmathfont ★ \setCJKmathfont [<font features>] {<font name>}
```

设置数学公式中的 CJK 字体族。如果使用了 CJKmath 选项,但是没有使用 \setCJKmathfont 设置数学公式中的 CJK 字体,那么将使用 \CJKfamilydefault 作为数学公式中的 CJK 字体。

```
\setCJKfallbackfamilyfont ★ \setCJKfallbackfamilyfont {<family>} [<font features>] {<font name>}
```

设置 CJK 字体族 <family> 的备用字体。例如,使用

```
\setCJKmainfont{SimSun}
\setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}
```

可以将 SimSun-ExtB 作为 SimSun 的备用字体。

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

xeCJK 在 里增加了 FallBack 这个选项。用来在声明主字体的时候,同时设置备用字体。例如,上面的例子等价于:

```
\setCJKmainfont[FallBack=SimSun-ExtB]{SimSun}
```

如果 FallBack 的值为空,将设置的是备用字体。例如,

```
\setCJKmainfont[FallBack,AutoFakeBold,Scale=.97]{SimSun-ExtB}
```

等价于

```
\setCJKfallbackfamilyfont{\CJKrmdefault}[AutoFakeBold,Scale=.97]{SimSun-ExtB}
```

```
\setCJKfallbackfamilyfont ★ \setCJKfallbackfamilyfont {<family>} [<common font features>]
{
  { [<font features1>] {<font name1>} },
  { [<font features2>] {<font name2>} },
  .....
}
```

Updated: 2013-06-30

\setCJKfallbackfamilyfont 还可以用于设置多层的备用字体。例如,使用

```
\setCJKmainfont[AutoFakeBold,AutoFakeSlant]{KaiTi_GB2312}
\setCJKfallbackfamilyfont{\CJKrmdefault}[AutoFakeSlant]
{ [BoldFont=SimHei]{SimSun} ,
  [AutoFakeBold] {SimSun-ExtB} }
```

之后,就设置了 SimSun 是 KaiTi_GB2312 的备用字体,而 SimSun-ExtB 是 SimSun 的备用字体。若当前字体族缺字,并没有备用字体,则尝试使用 \CJKfamilydefault 的备用字体。

3.2.1 Xe_{La}TeX 的字体名查找

由于在 fontspec 宏包文档中缺少关于如何查看 Xe_{La}TeX 可用字体名的说明,这里略作说明。

Xe_{La}TeX 通常使用 fontconfig 库查找和调用字体,因此,可以用 fc-list 命令显示可用的字体。在命令行(Windows 的“命令提示符”,Linux 的 Console)下运行以下命令:

```
fc-list > fontlist.txt
```

可以将系统中所有安装的字体列表存入 fontlist.txt 文件中(可能很长)。

fc-list 命令列出的信息很多,而且在安装字体较多的 Windows 系统上的输出将非常庞大,如其中可能包含:

```
Times New Roman:style=cursiva,kurzíva,kursiv,Πλάγια,Italic,
Kursivoitu,Italique,Dőlt,Corsivo,Cursief,kursywa,Itálico,Курсив,
İtalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta cursiva,tučné kurzíva,fed kursiv,
Fett Kursiv,Έντονα Πλάγια,Bold Italic,Negrita Cursiva,
Lihavoitu Kursivoi,Gras Italique,Félkövér dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrìto Itálico,
```

Полужирный Курсив, Tučná kurzíva, Fet Kursiv, Kalın İtalik,
 Крепко poševno, nghiêng đậm, Lodi etzana
 Times New Roman:style=Negreta,tučné,fed,Fett,Έντονα,Bold,Negrita,
 Lihavoitu,Gras,Félkövér,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
 Полужирный,Fet,Kalın,Krepko,đậm,Lodia
 Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
 Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normalne,Navadno,
 thường,Arrunta
 宋体,SimSun:style=Regular
 黑体,SimHei:style=Normal,obyčejné,Standard,Κανονικά,Regular,Normaali,
 Normál,Normale,Standaard,Normalny,Обычный,Normalne,Navadno,Arrunta

在 fontspec 或 xeCJK 中使用的字体族名是上面列表中冒号前的部分。例如可以使用

```
\setmainfont{Times New Roman}
\setCJKmainfont{SimSun} % 或者 \setCJKmainfont{宋体}
```

来设置字体。

为了方便起见,fc-list 命令也可以加上各种选项控制输出格式,例如如果只要列出所有的中文字体的字体族名,可以用命令:

```
fc-list -f "%{family}\n" :lang=zh > zhfont.txt
```

这样就把字体列表保存在文件 zhfont.txt 中³。这样列出的字体列表就比较简明易用,如 Windows 下预装的中文字体:

```
Arial Unicode MS
FangSong, 仿宋
KaiTi, 楷体
Microsoft YaHei, 微软雅黑
MingLiU, 細明體
NSimSun, 新宋体
PMingLiU, 新細明體
SimHei, 黑体
SimSun, 宋体
```

要列出日文和韩文的字体,可以把 :lang=zh 选项中的 zh 改成 ja 或 ko。

fontspec 和 xeCJK 也可以使用字体的文件名访问字体。例如 Windows 下的宋体也可以使用命令:

```
\setCJKmainfont{simSun.ttc}
```

来设置。设置字体文件名的相关选项和语法在 fontspec 宏包手册中叙述甚详,这里不再赘述。有个别字体名不规范的中文字体,xeCJK 宏包可能无法正确地通过字体名访问,那么也可以使用这种方式设置。

3.3 CJK 分区字体设置

众所周知,CJK 文字数量极其庞大,单一的字体不可能涵盖所有的 CJK 文字。xeCJK 可以在同一 CJK 字体族下,自动使用不同的字体输出 CJK 字符范围内不同区块里的文字。首先要声明 CJK 子分区。

³由于汉字编码原因,Windows 下总需要把字体列表输出的文件中防止乱码。

<code>\xeCJKDeclareSubCJKBlock</code> ★	<pre>\xeCJKDeclareSubCJKBlock {⟨block⟩} {⟨block range⟩} \xeCJKDeclareSubCJKBlock* {⟨block⟩} {⟨block range⟩}</pre> <p>其中 $\langle block\ range\rangle$ 是逗号列表, 可以是 CJK 字符的 Unicode 范围, 也可以是单个字符的 Unicode。例如</p> <pre>{ `中 -> `文 , "3400 -> "4DBF , "5000 -> "7000 , `汉 , `字 , "3500 }</pre> <p>的形式。需要注意的是, 这里设置的 $\langle block\ range\rangle$ 除非确实需要(例如某些特殊字体使用了 Unicode 中的私人使用区的情况), 否则不要超出源代码中预设的 CJK 文字范围。使用</p> <pre>\xeCJKDeclareSubCJKBlock{SPUA}{ "E400 -> "E4DA , "E500 -> "E5E8 , "E600 -> "E6CE } \xeCJKDeclareSubCJKBlock{Ext-B}{ "20000 -> "2A6DF }</pre> <p>就声明了 SPUA 和 Ext-B 这两个子分区。同时在 3.2 节介绍的 CJK 字体设置命令的 $\langle font\ features\rangle$ 里新建了 SPUA 和 Ext-B 这两个选项。新建的这两个选项的使用方法跟 3.2 介绍的 FallBack 类似。可以通过它们来设置字体。</p> <p>例如, 可以使用</p> <pre>\setCJKmainfont[SPUA=SunmanPUA,Ext-B=SimSun-ExtB]{SimSun}</pre> <p>设置文档的主字体是 SimSun, SPUA 分区的字体是 SunmanPUA, 而 Ext-B 分区的字体是 SimSun-ExtB。</p> <p><code>\xeCJKDeclareSubCJKBlock</code> 应该在声明所有的 CJK 字体族之前使用。如果有某个 CJK 字体族没有设置 $\langle block\rangle$ 选项, 将使用 <code>\CJKfamilydefault</code> 的 $\langle block\rangle$ 选项作为该 CJK 字体族的 $\langle block\rangle$ 选项。如果希望在使用某 CJK 字体族时, 不在 CJK 主分区与 $\langle block\rangle$ 之间切换字体, 可以使用 $\langle block\rangle=*$ 选项。带星号的命令除了设置 CJK 子分区以外, 还重置标点符号所属的字符类。</p>
---	--

<code>\xeCJKCancelSubCJKBlock</code>	<pre>\xeCJKCancelSubCJKBlock {⟨block₁⟩, ⟨block₂⟩, ...} \xeCJKCancelSubCJKBlock* {⟨block₁⟩, ⟨block₂⟩, ...}</pre> <p>在文档中取消对 CJK 分区的声明。带星号的命令还重置标点符号所属的字符类。</p>
<code>\xeCJKRestoreSubCJKBlock</code>	<pre>\xeCJKRestoreSubCJKBlock {⟨block₁⟩, ⟨block₂⟩, ...} \xeCJKRestoreSubCJKBlock* {⟨block₁⟩, ⟨block₂⟩, ...}</pre> <p>在文档中恢复对 CJK 分区的声明。带星号的命令还重置标点符号所属的字符类。</p>

3.4 设置 CJK 字符范围

<code>\xeCJKDeclareCharClass</code> ★	<pre>\xeCJKDeclareCharClass {⟨class⟩} {⟨class range⟩} \xeCJKDeclareCharClass* {⟨class⟩} {⟨class range⟩}</pre> <p>$\langle class\ range\rangle$ 的格式和 3.3 节的 $\langle block\ range\rangle$ 相同。$\langle class\rangle$ 的有效值见源代码(第 5.4 节)。<code>\xeCJK</code> 已经支持 Unicode 中所有 CJK 文字和标点。一般来说, 不要轻易改变字符类别。带星号的命令除了设置字符类别以外, 为了确保标点处理的正确性, 还重置标点符号所属的字符类。</p>
<code>\xeCJKResetCharClass</code> ★	用于恢复 <code>\xeCJK</code> 对各个字符类别的初始化设置。
<code>\xeCJKResetPunctClass</code> ★	用于重置标点符号所属的字符类。
<code>\normalspacedchars</code>	<pre>\normalspacedchars {⟨char list⟩}</pre> <p>在 $\langle char\ list\rangle$ 中出现的字符两端不自动添加空格, 初始设置是 /。</p>

3.5 标点符号的处理

`\xeCJK` 对标点符号的输出宽度的调整是通过调整其左边或右边的空白宽度来实现的。按照目前的处理方式, 对于位于左边的标点符号(如左引号), `\xeCJK` 只能调整它左边的空白; 对于位于右边的标点符号(如右引号), `\xeCJK` 只能调整它右边的空白; 对于居中的标点符号, 则调整其左右空白, 以保证其居中。

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 `plain` 以外的所有标点处理格式。

<code>\xeCJKsetwidth</code> ★	<code>\xeCJKsetwidth {⟨标点列表⟩} {⟨length⟩}</code>
-------------------------------	---

⟨标点列表⟩ 可以是单个标点,也可以是多个标点。例如,

```
\xeCJKsetwidth{。?}{0.7em}
```

将设置句号和问号所占的宽度为 0.7 em。

<code>\xeCJKsetkern</code> ★	<code>\xeCJKsetkern {⟨前标点⟩} {⟨后标点⟩} {⟨length⟩}</code>
------------------------------	---

`\xeCJK` 会根据选定的标点处理格式自动调整相邻的前后两个 CJK 标点符号的空白宽度。如果需要个别情况进行特殊调整,可以使用这个命令。例如,

```
\xeCJKsetkern{:}{“}{0.3em}
```

将设置冒号与左双引号之间的空白宽度为 0.3 em。

3.5.2 定义标点符号处理格式

<code>\xeCJKDeclarePunctStyle</code> ★	<code>\xeCJKDeclarePunctStyle {⟨style⟩} {⟨options⟩}</code>
--	--

New: 2012-11-10

定义新的标点符号处理格式,已经存在的同名格式将被覆盖。可以设置的选项将在下面介绍。

<code>\xeCJKEditPunctStyle</code> ★	<code>\xeCJKEditPunctStyle {⟨style⟩} {⟨options⟩}</code>
-------------------------------------	---

New: 2012-11-10

修改已有的标点符号处理格式。

下面是可以设置的标点符号格式选项。其中左边一栏是选项名称,中间是选项的输入值类型,右边则是相关说明。某些选项之间是互斥的,具有优先级关系。要使下一级的选项有效,则需要先禁用上一级的设置:对于 `⟨boolean⟩` 类型的选项,将其设置为 `false`,对于 `⟨length⟩` 类型的选项,将其设置为 `\maxdimen`,而对于 `⟨real⟩` 类型的选项,将其设置为 0。

`enabled-global-setting` `⟨boolean⟩` 是否使用 `\xeCJKsetup` 的 `PunctWidth` 选项和 `\xeCJKsetwidth`、`\xeCJKsetkern` 的设置。默认值是 `true`。

`fixed-punct-width` `⟨length⟩` 设置单个标点符号的宽度。默认值是 `\maxdimen`。

`fixed-punct-ratio` `⟨real⟩` 设置单个标点符号的输出宽度与实际宽度的比例。默认值是 1.0。

`mixed-punct-width` `⟨length⟩` 设置句末标点符号的宽度。其中句末标点符号通过 `\xeCJKsetup` 的 `KaiMingPunct` 来设置。默认值是与 `fixed-punct-width` 选项的值相同。

`mixed-punct-width` `⟨real⟩` 设置句末标点符号的宽度比例。默认值是与 `fixed-punct-ratio` 选项的值相同。

`middle-punct-width` `⟨length⟩` 设置居中标点符号的宽度。其中居中标点符号通过 `\xeCJKsetup` 的 `MiddlePunct` 来设置。默认值是与 `fixed-punct-width` 选项的值相同。

`middle-punct-width` `⟨real⟩` 设置居中标点符号的宽度比例。默认值是与 `fixed-punct-ratio` 选项的值相同。

以上三个选项设置的是标点的固定宽度或比例,`\xeCJK` 会根据设定的选项计算标点符号左/右的空白宽度。下面的选项设置的是标点符号左/右的空白宽度或比例,因此不同标点符号的宽度可能会不同。为了使下面的选项生效,需要先禁用上面的相应选项。优先级自上而下。

`fixed-margin-width` `⟨length⟩` 设置标点的左/右空白宽度。默认值是 `\maxdimen`。

`fixed-margin-ratio` `⟨real⟩` 设置标点的左/右空白宽度与字体中该标点的相应实际边界宽度的比例。默认值是 1.0。

`mixed-margin-width` `⟨length⟩` 设置句末标点的左/右空白宽度。默认值是与 `fixed-margin-width` 的值相同。

`mixed-margin-ratio` `⟨real⟩` 设置句末标点的左/右空白宽度的比例。默认值是与 `fixed-margin-ratio` 的值相同。

`middle-margin-width` `⟨length⟩` 设置居中标点的两边空白宽度。默认值是与 `fixed-margin-width` 的值相同。

`middle-margin-ratio` `⟨real⟩` 设置居中标点的两边空白宽度之和与两边实际两边边界宽度之和的比例。默认值是与 `fixed-margin-ratio` 的值相同。

`add-min-bound-to-margin` `⟨boolean⟩` 是否在以上计算结果的基础上再加上标点的左右实际边界宽度中的最小值。这个选项对居中的标点无效。默认值是 `false`。

optimize-margin *(boolean)* 使用以上设置空白宽度或比例的选项时,最终输出的标点符号左/右的空白宽度可能大于原来的实际边界宽度。若此时本选项被设置为 **true**,则使用原来的实际边界宽度。而使用 **fixed-punct-width** 选项计算得出的左/右宽度可能小于该标点的另一侧宽度,若此时本选项被启用,则使用该标点的另一侧宽度。默认值为 **false**。

margin-minimum *(length)* 指定标点符号左/右的最小空白宽度。当经过以上选项设置的空白宽度小于这个选项的值时,则使用这个选项的值。默认值是 **0 pt**。

下面的选项处理的是前后相邻的两个标点符号之间的空白宽度。这些选项是互斥的,优先级自上而下。

enabled-kerning *(boolean)* 是否调整前后相邻的两个标点之间的空白宽度。如果设置为 **false**,则每个标点都按原来的输出宽度输出。默认值是 **true**。

min-bound-to-kerning *(boolean)* 是否使用当前字体中前面标点实际左右边界的最小值与后面标点实际左右边界的最小值中的最大值作为两个标点之间的空白宽度。默认值是 **false**。

kerning-total-width *(length)* 设置两个标点的总共宽度。此时 **xeCJK** 会自动计算两个标点之间的空白宽度。默认值是 **\maxdimen**。

kerning-total-ratio *(real)* 设置两个标点的总共输出宽度与实际宽度的比例。默认值是 **0.75**。

same-align-margin *(length)* 前后两个标点位于同侧时,它们之间的空白宽度。默认值是 **\maxdimen**。

same-align-ratio *(real)* 前后两个标点位于同侧时,它们之间的空白宽度与实际输出宽度的比例。默认值是 **0**。

different-align-margin *(length)* 前后两个标点位于异侧时,它们之间的空白宽度。默认值是 **\maxdimen**。

different-align-ratio *(real)* 前后两个标点位于异侧时,它们之间的空白宽度与实际输出宽度的比例。默认值是 **0**。

kerning-margin-width *(length)* 设置前后两个标点之间的空白宽度。默认值是 **\maxdimen**。

kerning-margin-ratio *(real)* 设置前后两个标点之间的空白宽度与实际输出空白的比例。默认值是 **1.0**。

optimize-kerning *(boolean)* 使用以上选项计算出两个标点之间的空白宽度可能小于通过 **min-bound-to-kerning** 选项得出的结果。当出现这一情况时,若此选项被设置为 **true**,则使用该选项的空白宽度。默认值为 **false**。

kerning-margin-minimum *(length)* 指定两个标点之间的最小空白宽度。当经过以上选项设置的空白宽度小于这个选项的值时,则使用这个选项的值。默认值是 **0 pt**。

事实上,**xeCJK** 的默认设置就相当于中文全角(**quanjiao**)格式。可以使用上面说明的选项定义新的标点处理格式。例如,使用

```
\xeCJKDeclarePunctStyle { mine }
{
  fixed-punct-ratio      = 0 ,
  fixed-margin-width     = 0 pt ,
  mixed-margin-width     = \maxdimen ,
  mixed-margin-ratio     = 0.5 ,
  middle-margin-width    = \maxdimen ,
  middle-margin-ratio    = 0.5 ,
  add-min-bound-to-margin = true ,
  min-bound-to-kerning   = true ,
  kerning-margin-minimum = 0.1 em
}
```

就定义了一个名为 **mine** 的标点处理格式。可以在通过

```
\xeCJKsetup{PunctStyle=mine}
```

在文档中使用这个格式。它的意义是:使用标点符号的实际左右边界中的最小值作为其左/右空白的宽度,对于句末标点和居中标点,再加上实际边界空白的一半;使用相邻两个标点的实际边界中的较小值作为它们之间的空白宽度,并且最小的空白宽度是 **0.1 em**。再例如,使用

```
\xeJKEditPunctStyle { hangmobanjiao } { enabled-global-setting = false }
```

将使得 **\xeCJKsetkern** 等的设置对 **hangmobanjiao** 这一格式无效。

3.6 其它

`\xeCJKVerbAddon`
`\xeCJKOffVerbAddon`

Updated: 2013-06-29

调整文字间距以便于让 CJK 字符占的宽度等于西文等宽字体中两个空格的宽度。如果这两个空格的宽度小于当前 CJK 正常文字的宽度,将对 CJK 字体进行适当地缩小。这有利于等宽字体的代码对齐等情形。需要注意的是,`\xeCJKVerbAddon` 对 `xeCJK` 的内部进行了比较大的修改,使用它之后,关于标点符号的禁则将会丢失。所以不能单独使用,应该放在分组里限制其作用域,否则是无效的。当然它可以和其它关于代码抄录的宏包配合使用。例如,可以使用于 `fancyvrb` 宏包的 `formatcom` 选项。此时设置的西文字体应该确实是等宽的以保证对齐。若西文等宽字体发生变动(包括字体大小),则需要在其后面使用 `\xeCJKVerbAddon`,重新计算间距的宽度。`\xeCJKOffVerbAddon` 用于在使用 `\xeCJKVerbAddon` 的环境中局部取消它的作用。由于 `listings` 宏包有自己的代码对齐机制,所以 `\xeCJKVerbAddon` 在由 `listings` 定义的代码环境中无效。

`\xeCJKnobreak`

New: 2012-12-03

……汉字。`\xeCJKnobreak\footnote{脚注}`

`\xeCJKnobreak` 用在全角标点符号后面,目的是确保不能在此处断行。如果已经启用了前面介绍的 `CheckFullRight` 选项,则不需要再用此命令。

4 已知问题和兼容性

\XeTeX 在配置文件 `unicode-letters.tex` 中将所有 CJK 表意文字的 `\catcode` 设置为 11。因此汉字可以直接用作控制序列的名字,但是当汉字出现在控制序列后面的时候,要用空格分隔开,否则就会出现“! Undefined control sequence.”的错误。

`xeCJK` 使用并重新定义了 CJK 宏包的部分宏命令,如 `\CJKfamily`、`\CJKsymbol` 和 `\CJKglue` 等。需要指出,`xeCJK` 不需要 CJK 的支持,并且 `xeCJK` 自动禁止在它之后载入 CJK 宏包。可以在 `xeCJK` 之后载入 `CJKnumb` 宏包,实现数字的中文化。

`xeCJK` 包含有一个子宏包 `xeCJKfntef`,可以用它来实现汉字加点和可断行的下划线等。它是 `CJKfntef` 宏包在 \XeTeX 下的替换版本,基本用法完全一致,在 `CJKfntef` 的源文件 `CJKfntef.sty` 的注解部分里有说明。

`xeCJK` 进行了一些处理,使得在使用 \XeTeX 时 `listings` 宏包可以支持 Unicode,因此在 `listings` 定义的代码环境中可以直接使用中文,不再需要通过 `escapechar`。

新版本(3.x)的 `xeCJK` 完全使用 \LaTeX 3 的语法来编写。 \LaTeX 3 放弃了 `\outer` 宏的概念,因此相关工具在遇到 `\outer` 宏时可能会存在问题。按照目前 `xeCJK` 的实现方式,在 CJK 文字后面遇到 `\outer` 宏时会出现类似

```
! Forbidden control sequence found while scanning use of \use_i:nn
```

的错误。目前已知的有 `cprotect` 宏包提供的 `\cprotect`。它的定义是

```
\outer\long\def\cprotect{\icprotect}
```

因此,这时可以暂时用 `\icprotect` 代替 `\cprotect`。事实上,当 `cprotect` 被引入时,`xeCJK` 将使用

```
\let\cprotect\icprotect
```

来取消 `\cprotect` 的外部宏限制。但由于 `\cprotect` 的特殊性,应该只在外部使用它,即不要让它出现在任何宏的参数中。其它 `\outer` 宏的情况,可以在它前面加上 `\relax` 来回避上面的错误。

`xeCJK` 依赖 \XeTeX 的 `\XeTeXinterchartoks` 机制,与使用相同机制的宏包(例如 `polyglossia`)可能会存在大小不一的冲突。`xeCJK` 虽然为此作了一些处理,但与它们共同使用时应该小心。

5 xeCJK 代码实现

```
1 <*package>  
2 <@@=xeCJK>
```

5.1 运行环境检查

`xeCJK` 必须使用 \XeTeX 引擎的支持。

```
3 \msg_new:nnn { xeCJK } { Require-XeTeX }  
4 {
```

```

5 The~xeCJK~package~requires~XeTeX~to~function.\\\
6 You~must~change~your~typesetting~engine~to~"xelatex" \\
7 instead~of~plain~"latex"~or~"pdflatex"~or~"lualatex".\\
8 Loading~xeCJK~will~abort!
9 }
10 \xetex_if_engine:F { \msg_critical:nn { xeCJK } { Require-XeTeX } }
    下面这些 CJK 系列宏包不应该被使用。
11 \msg_new:nnn { xeCJK } { incompatible-package }
12 {
13     The~`#1'~package~and~xeCJK~are~incompatible.\\\
14     Please ~ \str_if_eq:nnTF {#1} { CJKnumb }
15         { load~it~after~xeCJK. } { do~not~use~it. }
16 }
17 \clist_map_inline:nn { CJKnumb , CJKulem , CJKvert , CJKpunct , CJKutf8 , CJK }
18 {
19     \@ifpackageloaded {#1}
20         { \msg_error:nnn { xeCJK } { incompatible-package } {#1} }
21         { }
22 }

```

禁止在 xeCJK 宏包后再载入 CJK 宏包,并使得 CJKnumb 和 CJKfntef 宏包可以使用。

```

23 \tl_gset:cn { ver@CJK.sty } { 9999/99/99 }
24 \tl_gset_eq:cc { ver@CJKutf8.sty } { ver@CJK.sty }
25 \tl_gset_eq:cc { ver@CJKvert.sty } { ver@CJK.sty }
26 \tl_gset_eq:cc { ver@CJKulem.sty } { ver@CJK.sty }
27 \tl_gset_eq:cc { ver@CJKpunct.sty } { ver@CJK.sty }
    应该使用较新版本的 expl3 宏包。
28 \msg_new:nnn { xeCJK } { 13-too-old }
29 {
30     Support~package~`#1'~too~old. \\\
31     Please~update~an~up~to~date~version~of~the~bundles\\\
32     `l3kernel'~and~`l3packages'\\\
33     using~your~TeX~package~manager~or~from~CTAN.\\
34     \str_if_eq:nnT {#1} { expl3 } { Loading~xeCJK~will~abort! }
35 }
36 \@ifpackagelater { expl3 } { 2012/11/10 } { }
37 { \msg_critical:nnn { xeCJK } { 13-too-old } { expl3 } }

```

以下日期以前的 xtemplate 宏包关于 \KeyValue 的 Bug 会影响到后面标点符号的处理。

```

38 \RequirePackage { xtemplate }
39 \@ifpackagelater { xtemplate } { 2012/11/10 } { }
40 { \msg_error:nnn { xeCJK } { 13-too-old } { xtemplate } }
41 \RequirePackage { xparse , l3keys2e }

```

5.2 内部工具

分配临时变量。

```

42 \tl_new:N \l__xeCJK_tmp_tl
43 \int_new:N \l__xeCJK_tmp_int
44 \box_new:N \l__xeCJK_tmp_box
45 \dim_new:N \l__xeCJK_tmp_dim
46 \bool_new:N \l__xeCJK_tmp_bool
47 \skip_new:N \l__xeCJK_tmp_skip
48 \clist_new:N \l__xeCJK_tmp_clist

```

__xeCJK_msg_new:nn 各种信息函数的缩略形式。

```

    \__xeCJK_error:n 49 \cs_new_protected_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
    \__xeCJK_error:nx 50 \cs_new_protected_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
    \__xeCJK_warning:nx 51 \cs_new_protected_nopar:Npn \__xeCJK_error:nx { \msg_error:nnx { xeCJK } }
    \__xeCJK_info:nx 52 \cs_new_protected_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
    \__xeCJK_info:nxx 53 \cs_new_protected_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }

```

```

54 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
55 \cs_new_protected_nopar:Npn \__xeCJK_info:nx { \msg_info:nnx { xeCJK } }
56 \cs_new_protected_nopar:Npn \__xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }
(End definition for \__xeCJK_msg_new:nm and others.)

```

```

\__xeCJK_allow_break:
\__xeCJK_no_break:
57 \cs_new_protected_nopar:Npn \xeCJK_allow_break: { \tex_penalty:D \c_zero }
58 \cs_new_protected_nopar:Npn \xeCJK_no_break: { \tex_penalty:D \c_ten_thousand }
(End definition for \xeCJK_allow_break: and \xeCJK_no_break:.)

```

判断宏包是否被引入,可用于文档正文中。

```

xeCJK_if_package_loaded_p:n
xeCJK_if_package_loaded:nTF
59 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
60 {
61   \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
62   { \prg_return_true: } { \prg_return_false: }
63 }
64 \tl_const:Nx \c__xeCJK_package_ext_tl { \@pkgextension }
(End definition for \xeCJK_if_package_loaded:n.)

```

在 \document 前后加上各种钩子。

```

\__xeCJK_at_end_preamble:n
\__xeCJK_after_preamble:n
\__xeCJK_after_end_preamble:n
65 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
66 \tl_new:N \g__xeCJK_after_preamble_hook_tl
67 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl
68 \cs_new_protected:Npn \__xeCJK_at_end_preamble:n #1
69 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
70 \cs_new_protected:Npn \__xeCJK_after_preamble:n #1
71 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
72 \cs_new_protected:Npn \__xeCJK_after_end_preamble:n #1
73 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
74 \xeCJK_if_package_loaded:nTF { etoolbox }
75 {
76   \AtEndPreamble { \g__xeCJK_at_end_preamble_hook_tl }
77   \AfterPreamble { \g__xeCJK_after_preamble_hook_tl }
78   \AfterEndPreamble { \g__xeCJK_after_end_preamble_hook_tl }
79 }
80 {
81   \AtBeginDocument { \g__xeCJK_after_preamble_hook_tl }
82   \cs_new_protected_nopar:Npn \__xeCJK_document_left_hook:
83   { \group_end: \g__xeCJK_at_end_preamble_hook_tl \group_begin: }
84   \cs_new_protected_nopar:Npn \__xeCJK_document_right_hook:
85   { \scan_stop: \g__xeCJK_after_end_preamble_hook_tl \tex_ignorespaces:D }
86   \cs_gset_nopar:Npx \document
87   {
88     \__xeCJK_document_left_hook:
89     \exp_not:o { \document }
90     \__xeCJK_document_right_hook:
91   }
92 }
(End definition for \__xeCJK_at_end_preamble:n, \__xeCJK_after_preamble:n, and \__xeCJK_after_end_preamble:n.)

```

\xeCJK_reverse:nnn #1 为 #2 或 #3,若 #1 和 #2 相等,则返回 #3,否则返回 #2。

```

93 \cs_new_nopar:Npn \xeCJK_reverse:nnn #1#2#3
94 { \str_if_eq_x:nnTF {#1} {#2} {#3} {#2} }
(End definition for \xeCJK_reverse:nnn.)

```

去掉 #1 外层的分组括号。

```

\__xeCJK_tl_remove_outer_braces:N
\__xeCJK_tl_remove_outer_braces:n
95 \cs_new_protected_nopar:Npn \xeCJK_tl_remove_outer_braces:N #1
96 { \tl_set:Nx #1 { \exp_args:NV \xeCJK_tl_remove_outer_braces:n #1 } }
97 \cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
98 {
99   \exp_last_unbraced:Nf
100   \__xeCJK_tl_remove_outer_braces:w { \tl_trim_spaces:n {#1} } \s_stop

```



```

101 }
102 \cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \s__stop
103 {
104   \bool_if:nTF { \tl_if_single_p:n {#1} && ! ( \tl_if_head_is_N_type_p:n {#1} ) }
105   { \xeCJK_tl_remove_outer_braces:n {#1} }
106   { \tl_trim_spaces:n {#1} }
107 }

```

(End definition for `\xeCJK_tl_remove_outer_braces:N` and `\xeCJK_tl_remove_outer_braces:n`.)

`\xeCJK_cs_clear:N` 让控制序列的意义为空。

```

\xeCJK_cs_gclear:N
108 \cs_new_protected:Npn \xeCJK_cs_clear:N #1
109 { \cs_set_eq:NN #1 \prg_do_nothing: }
110 \cs_new_protected:Npn \xeCJK_cs_gclear:N #1
111 { \cs_gset_eq:NN #1 \prg_do_nothing: }

```

(End definition for `\xeCJK_cs_clear:N` and `\xeCJK_cs_gclear:N`.)

`\xeCJK_swap_cs:NN` 交换 #1 和 #2 的意义。

```

112 \cs_new_protected:Npn \xeCJK_swap_cs:NN #1#2
113 {
114   \cs_set_eq:NN \__xeCJK_swap_cs_aux:w #1
115   \cs_set_eq:NN #1 #2
116   \cs_set_eq:NN #2 \__xeCJK_swap_cs_aux:w
117   \cs_undefine:N \__xeCJK_swap_cs_aux:w
118 }

```

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

`\xeCJK_font_gset_to_current:c` #1 是控制序列的名字, 令它等于当前字体命令。

```

119 \cs_new_protected_nopar:Npn \xeCJK_font_gset_to_current:c #1
120 {
121   \exp_after:wN \cs_gset_eq:NN
122   \cs:w #1 \exp_after:wN \cs_end: \tex_the:D \tex_font:D
123 }

```

(End definition for `\xeCJK_font_gset_to_current:c`.)

`\xeCJK_glyph_if_exist_p:N` 判断当前字体中是否含有字符 #1。`fontspec` 中的类似函数在判断为真的时候, 会留有一个 `\scan_stop:`, 造成不必要的边界, 同时也不完全可展。因此, 我们重新定义它。

`\xeCJK_glyph_if_exist:NTF`

```

124 \prg_new_conditional:Npnn \xeCJK_glyph_if_exist:N #1 { p , T , F , TF }
125 {
126   \etex_iffontchar:D \tex_font:D `#1 \exp_stop_f:
127   \prg_return_true: \else: \prg_return_false: \fi:
128 }

```

(End definition for `\xeCJK_glyph_if_exist:N`.)

`\c_xeCJK_space_skip_tl` 当前字体状态下, 一个字间空格产生的 `glue` 的长度, 包括伸展和收缩部分。

```

129 \tl_const:Nn \c_xeCJK_space_skip_tl
130 {
131   \int_compare:nNnTF \g__xeCJK_spacefactor_int = \c_one_thousand
132   {
133     \skip_if_eq:nNnTF \tex_spaceskip:D \c_zero_skip
134     {
135       \tex_fontdimen:D \c_two \tex_font:D
136       plus \tex_fontdimen:D \c_three \tex_font:D
137       minus \tex_fontdimen:D \c_four \tex_font:D
138     }
139     { \tex_spaceskip:D }
140   }
141   {
142     \skip_if_eq:nNnTF \tex_spaceskip:D \c_zero_skip
143     {
144       \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
145       { \__xeCJK_space_skip_scale:nnn { \tex_fontdimen:D \c_two \tex_font:D } }
146       {

```

```

147         \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
148         {
149             \__xeCJK_space_skip_scale:nnn
150             {
151                 \tex_fontdimen:D \c_two \tex_font:D +
152                 \tex_fontdimen:D \c_seven \tex_font:D
153             }
154         }
155         { \tex_xspaceskip:D \use_none:nn }
156     }
157     { \tex_fontdimen:D \c_three \tex_font:D }
158     { \tex_fontdimen:D \c_four \tex_font:D }
159 }
160 {
161     \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
162     { \__xeCJK_space_skip_scale:nnn { \tex_spaceskip:D } }
163     {
164         \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
165         {
166             \__xeCJK_space_skip_scale:nnn
167             { \tex_spaceskip:D + \tex_fontdimen:D \c_seven \tex_font:D }
168         }
169         { \tex_xspaceskip:D \use_none:nn }
170     }
171     { \etex_gluestretch:D \tex_spaceskip:D }
172     { \etex_glueshrink:D \tex_spaceskip:D }
173 }
174 }
175 }
176 \cs_new_nopar:Npn \__xeCJK_space_skip_scale:nnn #1#2#3
177 {
178     \dim_eval:n {#1}
179     plus \fp_eval:n { \g__xeCJK_spacefactor_int / 1000 } #2
180     minus
181     \int_div_truncate:nn
182     { 1000 * \tex_number:D #3 } { \g__xeCJK_spacefactor_int } sp
183 }
184 \int_new:N \g__xeCJK_spacefactor_int
185 \int_gset_eq:NN \g__xeCJK_spacefactor_int \c_one_thousand
(End definition for \c_xeCJK_space_skip_tl.)

```

`\xeCJK_glue_to_skip:nN` 取得一个 glue 的长度, 包括伸展和收缩部分。

```

186 \cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2
187 {
188     \hbox_set:Nn \l__xeCJK_tmp_box { #1 \scan_stop: \exp_after:wN } \exp_after:wN
189     \skip_set:Nn \exp_after:wN #2 \exp_after:wN { \skip_use:N \tex_lastskip:D }
190 }
(End definition for \xeCJK_glue_to_skip:nN.)

```

`\xeCJK_if_blank_x_p:n` 判断是否为空或者仅含一个空格。

```

\xeCJK_if_blank_x:nTF
191 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
192 {
193     \if_case:w \pdfTeX_strcmp:D { } {#1} \exp_stop_f:
194     \prg_return_true:
195     \else:
196     \if_case:w \pdfTeX_strcmp:D { ~ } {#1} \exp_stop_f:
197     \prg_return_true: \else: \prg_return_false: \fi:
198     \fi:
199 }
(End definition for \xeCJK_if_blank_x:n.)

```

\xeCJK_int_until_do:nn 由于定义较为简单,可以比 \int_until_do:nNnn 稍微快一点点。
 _xeCJK_int_until_do:wn

```

200 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
201 { \_xeCJK_int_until_do:wn \use_none:n { \reverse_if:N \if_int_compare:w #1#2 } }
202 \cs_new_protected:Npn \_xeCJK_int_until_do:wn \use_none:n #1
203 { #1 \exp_after:wN \_xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
204 \int_new:N \l_xeCJK_begin_int
205 \int_new:N \l_xeCJK_end_int
(End definition for \xeCJK_int_until_do:nn and \_xeCJK_int_until_do:wn.)

```

\xeCJK_peek_catcode_ignore_spaces:NTF 我们在里面设置了一个变量 \l_xeCJK_peek_ignore_spaces_bool 用于标识后面的空格是否被省略掉了。

```

206 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
207 {
208   \cs_set_eq:NN \l_peek_search_token #1 \scan_stop:
209   \tl_set:Nn \_xeCJK_peek_catcode_true:w { \group_align_safe_end: #2 }
210   \tl_set:Nn \_xeCJK_peek_catcode_false:w { \group_align_safe_end: #3 }
211   \bool_set_false:N \l_xeCJK_peek_ignore_spaces_bool
212   \group_align_safe_begin:
213   \peek_after:Nw \_xeCJK_peek_catcode_ignore_spaces_branches:w
214 }
215 \cs_new_protected_nopar:Npn \_xeCJK_peek_catcode_ignore_spaces_branches:w
216 {
217   \if_meaning:w \l_peek_token \c_space_token
218     \bool_set_true:N \l_xeCJK_peek_ignore_spaces_bool
219     \exp_after:wN \peek_after:Nw
220     \exp_after:wN \_xeCJK_peek_catcode_ignore_spaces_branches:w
221     \tex_romannumeral:D 0
222   \else:
223     \if_catcode:w
224       \exp_not:N \l_peek_token \exp_not:N \l_peek_search_token
225       \exp_after:wN \exp_after:wN
226       \exp_after:wN \_xeCJK_peek_catcode_true:w
227     \else:
228       \exp_after:wN \exp_after:wN
229       \exp_after:wN \_xeCJK_peek_catcode_false:w
230     \fi:
231   \fi:
232 }
233 \tl_new:N \_xeCJK_peek_catcode_true:w
234 \tl_new:N \_xeCJK_peek_catcode_false:w
235 \bool_new:N \l_xeCJK_peek_ignore_spaces_bool
(End definition for \xeCJK_peek_catcode_ignore_spaces:NTF.)

```

\xeCJK_peek_after_ignore_spaces:nw 与 \@ifnextchar 和 \futurenonspacelike 类似,会省略掉后面的空格。

```

236 \cs_new_protected:Npn \xeCJK_peek_after_ignore_spaces:nw #1
237 {
238   \tl_set:Nn \_xeCJK_peek_after_do:w { \group_align_safe_end: #1 }
239   \group_align_safe_begin:
240   \peek_after:Nw \_xeCJK_peek_ignore_spaces_branches:w
241 }
242 \cs_new_protected_nopar:Npn \_xeCJK_peek_ignore_spaces_branches:w
243 {
244   \if_meaning:w \l_peek_token \c_space_token
245     \exp_after:wN \peek_after:Nw
246     \exp_after:wN \_xeCJK_peek_ignore_spaces_branches:w
247     \tex_romannumeral:D 0
248   \else:
249     \exp_after:wN \_xeCJK_peek_after_do:w
250   \fi:
251 }
(End definition for \xeCJK_peek_after_ignore_spaces:nw.)

```

`\xeCJK_token_value_class:N` 用于取得记号 #1 所在的 XeTeX 字符类。#1 应为 `\catcode` 为 11 或 12 的显性或隐性记号。

```

252 \cs_new_nopar:Npn \xeCJK_token_value_class:N #1
253 { \XeTeXcharclass \xeCJK_token_value_charcode:N #1 }
(End definition for \xeCJK_token_value_class:N)

```

`\xeCJK_token_value_charcode:N` 当记号 #1 的 `charcode` 大于等于 0x10000 时, XeTeX 0.9999.0 版以前的 `\meaning` 的返回结果比较特殊⁴, 需要特别处理。同时使用较新版本中提供的原语设置 `mathcode`。目前, 0.9999.0 版以后的 XeTeX 的 `\meaning` 对于超出 BMP 的字符, 会返回两个字符, 分别对应于其 UTF-16 编码的首尾代理。⁵

```

254 \cs_new_nopar:Npn \xeCJK_token_value_charcode:N #1
255 { \exp_after:wN \__xeCJK_token_value_charcode:w \token_to_meaning:N #1 \q_stop }
256 \fp_compare:nNnTF { \int_use:N \xetex_XeTeXversion } > { 0.9998 }
257 {
258   \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
259   {
260     \int_eval:n
261     {
262       \tl_if_empty:nTF {#4}
263       { `#3 }
264       { ( `#3 - "D800 ) * "400 + ( `#4 - "DC00 ) + "10000 }
265     }
266   }
267   \cs_new_eq:NN \xeCJK_xetex_mathcode:w \Umathcode
268 }
269 {
270   \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
271   { \int_eval:n { \tl_if_empty:nTF {#4} { `#3 } { "20000 } } }
272   \cs_new_eq:NN \xeCJK_xetex_mathcode:w \XeTeXmathcode
273 }
(End definition for \xeCJK_token_value_charcode:N)

```

`\xeCJK_if_CJK_class_p:N` 判断字符 #1 是否为 CJK 字符类, 包括文字和标点符号。

`\xeCJK_if_CJK_class:N`TF

```

274 \prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
275 {
276   \if_cs_exist:w \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 } \cs_end:
277   \prg_return_true: \else: \prg_return_false: \fi:
278 }
279 \cs_new_nopar:Npn \__xeCJK_CJK_class_tl:n #1
280 { c__xeCJK_CJK_class_ \int_eval:n {#1} _tl }
281 \cs_generate_variant:Nn \__xeCJK_CJK_class_tl:n { c }
(End definition for \xeCJK_if_CJK_class:N)

```

`\xeCJK_if_same_class_p:NN` 判断两个字符是否同属于一个字符类。

`\xeCJK_if_same_class:NN`TF

```

282 \prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
283 {
284   \if_int_compare:w
285   \xeCJK_token_value_class:N #1 = \xeCJK_token_value_class:N #2 \exp_stop_f:
286   \prg_return_true: \else: \prg_return_false: \fi:
287 }
(End definition for \xeCJK_if_same_class:NN)

```

5.3 功能开关

xeCJKactive 事实上, 将开启或关闭 XeTeX 的整个字符类机制。

```

288 \keys_define:nn { xeCJK / options }
289 {
290   xeCJKactive .choice: ,
291   xeCJKactive / true .code:n = { \makexeCJKactive } ,

```

⁴参见 <http://tug.org/pipermail/xetex/2013-January/023967.html> 和 <http://tex.stackexchange.com/a/64848>。

⁵参见 <http://tug.org/pipermail/xetex/2013-June/024543.html>。

```

292     xeCJKActive / false .code:n = { \makexeCJKinactive } ,
293     xeCJKActive      .default:n = { true }
294   }

```

(End definition for `xeCJKActive`. This function is documented on page 3.)

`\makexeCJKActive`

`\makexeCJKinactive`

```

295 \NewDocumentCommand \makexeCJKActive { } { \XeTeXinterchartokenstate = \c_one }
296 \NewDocumentCommand \makexeCJKinactive { } { \XeTeXinterchartokenstate = \c_zero }

```

(End definition for `\makexeCJKActive` and `\makexeCJKinactive`.)

抑制 BOM。

```

297 \char_set_catcode_ignore:n { "FEFF }

```

5.4 字符类别设定

分别用于记录在 `xeCJK` 中使用的字符类别名称和新建的字符类别的编号。

`\g__xeCJK_class_seq`
`\g__xeCJK_new_class_seq`

```

298 \seq_new:N \g__xeCJK_class_seq
299 \seq_new:N \g__xeCJK_new_class_seq

```

(End definition for `\g__xeCJK_class_seq` and `\g__xeCJK_new_class_seq`.)

`\xeCJK_new_class:n`

新建一个字符类别。#1 是自定义名称。

```

300 \cs_new_protected_nopar:Npn \xeCJK_new_class:n #1
301 {
302   \int_if_exist:cTF { \__xeCJK_class_csname:n {#1} }
303   { \__xeCJK_error:nx { class-already-defined } {#1} }
304   {
305     \exp_args:Nc \newXeTeXintercharclass { \__xeCJK_class_csname:n {#1} }
306     \clist_new:c { g__xeCJK_#1_range_clist }
307     \seq_gput_right:Nn \g__xeCJK_class_seq {#1}
308     \seq_gput_right:Nv \g__xeCJK_new_class_seq { \__xeCJK_class_csname:n {#1} }
309   }
310 }

```

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

`\xeCJK_save_class:nn`

保存 \XeTeX 预定义的字符类别。#1 是自定义名称, #2 是编号。

```

311 \cs_new_protected_nopar:Npn \xeCJK_save_class:nn #1#2
312 {
313   \int_if_exist:cTF { \__xeCJK_class_csname:n {#1} }
314   { \__xeCJK_error:nx { class-already-defined } {#1} }
315   {
316     \int_const:cn { \__xeCJK_class_csname:n {#1} } {#2}
317     \clist_new:c { g__xeCJK_#1_range_clist }
318     \seq_gput_right:Nn \g__xeCJK_class_seq {#1}
319   }
320 }

```

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

`__xeCJK_class_csname:n`

字符类名称对应的控制序列名字。

```

321 \cs_new_nopar:Npn \__xeCJK_class_csname:n #1 { c__xeCJK_#1_class_int }
322 \cs_new_eq:cN { \__xeCJK_class_csname:n { Others } } \l__xeCJK_tmp_int
323 \__xeCJK_msg_new:nn { class-already-defined }
324 {
325   XeTeX-character~class~`#1'~has-been~already-defined.\\
326   Please-take~another~name. \\
327 }

```

(End definition for `__xeCJK_class_csname:n`.)

`xeCJK` 需要以下字符类别用于字符输出。其中 `Default`、`CJK`、`FullLeft`、`FullRight`、`Boundary` 为 \XeTeX 中预定义的类别, `xeCJK` 新增加了 `HalfLeft`、`HalfRight`、`NormalSpace` 和 `IVS`。其中异体字选择符 (Ideographic Variation Selectors)⁶ 需要 \XeTeX 0.9999.0 以上的版本⁷和相关字体的支持。

⁶<http://www.unicode.org/reports/tr37/>

⁷<http://tug.org/pipermail/xetex/2013-March/024118.html>

类别	说明	例子
Default	西文一般符号	abc123
CJK	CJK 表意符号	汉字あい
FullLeft	全角左标点	(《: “
FullRight	全角右标点	,。)》”
HalfLeft	半角左标点	([{
HalfRight	半角右标点	,.?)]}
NormalSpace	前后原始间距的符号	/
Boundary	边界	空格
IVS	异体字选择符	“回字有四样写法”

Default 这五类是 X_YTeX 预定义类别。

```

      CJK      328 \xeCJK_save_class:nn { Default } { \c_zero }
FullLeft      329 \xeCJK_save_class:nn { CJK } { \c_one }
FullRight     330 \xeCJK_save_class:nn { FullLeft } { \c_two }
Boundary      331 \xeCJK_save_class:nn { FullRight } { \c_three }
              332 \xeCJK_save_class:nn { Boundary } { \c_two_hundred_fifty_five }
(End definition for Default and others.)

```

HalfLeft 新增西文半角左/右标点、前后原始间距的符号和异体字选择符类。

```

HalfRight     333 \xeCJK_new_class:n { HalfLeft }
NormalSpace   334 \xeCJK_new_class:n { HalfRight }
IVS           335 \xeCJK_new_class:n { NormalSpace }
              336 \xeCJK_new_class:n { IVS }
(End definition for HalfLeft and others.)

```

\c__xeCJK_HalfLeft_chars_clist 西文半角左/右标点和前后原始间距的字符类。

```

\c__xeCJK_HalfRight_chars_clist 337 \clist_const:Nn \c__xeCJK_HalfLeft_chars_clist
\c__xeCJK_NormalSpace_chars_clist 338 { "28 , "2D , "5B , "60 , "7B }
                                339 \clist_const:Nn \c__xeCJK_HalfRight_chars_clist
                                340 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
                                341 \clist_const:Nn \c__xeCJK_NormalSpace_chars_clist { "2F }
(End definition for \c__xeCJK_HalfLeft_chars_clist, \c__xeCJK_HalfRight_chars_clist, and \c__xeCJK_NormalSpace_chars_clist.)

```

以下对全角标点符号的归类来源于 X_YTeX 的脚本 [unicode-char-prep.pl](#) 和 Unicode 数据库⁸。

\c__xeCJK_OP_chars_clist

Open Punctuation (OP)

U+2018	‘	U+201C	“	U+2329	⟨	U+3008	⎵	U+300A	《	U+300C	⌈	U+300E	『
U+3010	【	U+3014	〔	U+3016	⎶	U+3018	⎷	U+301A	⎸	U+301D	⌋	U+FE17	⎹
U+FE35	⎵	U+FE37	⎶	U+FE39	⎷	U+FE3B	⎸	U+FE3D	⎹	U+FE3F	⎺	U+FE41	⎻
U+FE43	⎻	U+FE47	⎼	U+FE59	⎽	U+FE5B	⎾	U+FE5D	⎿	U+FF08	(U+FF3B	[
U+FF5B	{	U+FF5F	⎵	U+FF62	⎶								

以下代码的第一行是中西文共用的左引号。

```

342 \clist_const:Nn \c__xeCJK_OP_chars_clist
343 {
344     "2018 , "201C ,
345     "2329 , "3008 , "300A , "300C , "300E , "3010 , "3014 , "3016 , "3018 , "301A ,
346     "301D , "FE17 , "FE35 , "FE37 , "FE39 , "FE3B , "FE3D , "FE3F , "FE41 , "FE43 ,
347     "FE47 , "FE59 , "FE5B , "FE5D , "FF08 , "FF3B , "FF5B , "FF5F , "FF62
348 }
(End definition for \c__xeCJK_OP_chars_clist.)

```

\c__xeCJK_PR_chars_clist

Prefix Numeric (PR)

```

      U+20A9 | ￥ | U+FE69 | $ | U+FF04 | $ | U+FFE1 | £ | U+FFE5 | ¥ | U+FFE6 | ₩ |
349 \clist_const:Nn \c__xeCJK_PR_chars_clist
350 { "20A9 , "FE69 , "FF04 , "FFE1 , "FFE5 , "FFE6 }
(End definition for \c__xeCJK_PR_chars_clist.)

```

⁸<http://www.unicode.org/reports/tr14/>

`\c__xeCJK_FullLeft_chars_clist` 以上两类标点符号出现在文字的左边,不应出现在行尾位置。

```
351 \clist_const:Nx \c__xeCJK_FullLeft_chars_clist
352 {
353     \c__xeCJK_OP_chars_clist ,
354     \c__xeCJK_PR_chars_clist
355 }
(End definition for \c__xeCJK_FullLeft_chars_clist.)
```

`\c__xeCJK_CL_chars_clist` Close Punctuation (CL)

U+00B7	·	U+2019	’	U+201D	”	U+2014	—	U+2015	–	U+2025	..	U+2026	...
U+2027	·	U+2500	—	U+232A	}	U+3001	、	U+3002	。	U+3009	}	U+300B	》
U+300D	」	U+300F	』	U+3011	】	U+3015	}	U+3017	」	U+3019	」	U+301B	」
U+301E	”	U+301F	”	U+FE11	、	U+FE12	。	U+FE18	」	U+FE36	）	U+FE38	）
U+FE3A	）	U+FE3C	）	U+FE3E	》	U+FE40	✓	U+FE42	—	U+FE44	」	U+FE48	」
U+FE50	，	U+FE52	。	U+FE5A	）	U+FE5C	，	U+FE5E	，	U+FF09	）	U+FF0C	，
U+FF0E	。	U+FF3D]	U+FF5D	}	U+FF60	》	U+FF61	。	U+FF63	」	U+FF64	、

以下代码的第一行是中西文共用的一些标点符号。

```
356 \clist_const:Nn \c__xeCJK_CL_chars_clist
357 {
358     "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2027 , "2500 ,
359     "232A , "3001 , "3002 , "3009 , "300B , "300D , "300F , "3011 , "3015 , "3017 ,
360     "3019 , "301B , "301E , "301F , "FE11 , "FE12 , "FE18 , "FE36 , "FE38 , "FE3A ,
361     "FE3C , "FE3E , "FE40 , "FE42 , "FE44 , "FE48 , "FE50 , "FE52 , "FE5A , "FE5C ,
362     "FE5E , "FF09 , "FF0C , "FF0E , "FF3D , "FF5D , "FF60 , "FF61 , "FF63 , "FF64
363 }
(End definition for \c__xeCJK_CL_chars_clist.)
```

`\c__xeCJK_NS_chars_clist` Nonstarter (NS)

U+3005	々	U+301C	ゝ	U+303B	ゞ	U+303C	㌿	U+309B	ゝ	U+309C	ゝ	U+309D	ゝ
U+309E	ゞ	U+30A0	=	U+30FB	・	U+30FD	ゝ	U+30FE	ゝ	U+A015	□	U+FE54	：
U+FE55	：	U+FF1A	：	U+FF1B	；	U+FF65	・	U+FF9E	ゝ	U+FF9F	ゝ		

```
364 \clist_const:Nn \c__xeCJK_NS_chars_clist
365 {
366     "3005 , "301C , "303B , "303C , "309B , "309C , "309D , "309E , "30A0 , "30FB ,
367     "30FD , "30FE , "A015 , "FE54 , "FE55 , "FF1A , "FF1B , "FF65 , "FF9E , "FF9F
368 }
(End definition for \c__xeCJK_NS_chars_clist.)
```

`\c__xeCJK_EX_chars_clist` Exclamation/Interrogation (EX)

U+FE15	!	U+FE16	?	U+FE56	?	U+FE57	!	U+FF01	!	U+FF1F	?
--------	---	--------	---	--------	---	--------	---	--------	---	--------	---

```
369 \clist_const:Nn \c__xeCJK_EX_chars_clist
370 { "FE15 , "FE16 , "FE56 , "FE57 , "FF01 , "FF1F }
(End definition for \c__xeCJK_EX_chars_clist.)
```

`\c__xeCJK_IS_chars_clist` Infix Numeric Separator (IS)

U+FE10	’	U+FE13	：	U+FE14	；
--------	---	--------	---	--------	---

```
371 \clist_const:Nn \c__xeCJK_IS_chars_clist { "FE10 , "FE13 , "FE14 }
(End definition for \c__xeCJK_IS_chars_clist.)
```

`\c__xeCJK_CJ_chars_clist` Conditional Japanese Starter (CJ)

U+3041	あ	U+3043	い	U+3045	う	U+3047	え	U+3049	お	U+3063	っ	U+3083	や
U+3085	ゆ	U+3087	よ	U+308E	わ	U+3095	か	U+3096	け	U+30A1	ア	U+30A3	イ
U+30A5	ウ	U+30A7	エ	U+30A9	オ	U+30C3	ツ	U+30E3	ヤ	U+30E5	ユ	U+30E7	ヨ
U+30EE	ワ	U+30F5	カ	U+30F6	ケ	U+30FC	一	U+31F0	ク	U+31F1	シ	U+31F2	ス
U+31F3	ト	U+31F4	ヌ	U+31F5	ハ	U+31F6	ヒ	U+31F7	フ	U+31F8	ヘ	U+31F9	ホ
U+31FA	ム	U+31FB	ラ	U+31FC	リ	U+31FD	ル	U+31FE	レ	U+31FF	ロ	U+FF67	ア
U+FF68	イ	U+FF69	ウ	U+FF6A	エ	U+FF6B	オ	U+FF6C	カ	U+FF6D	キ	U+FF6E	コ
U+FF6F	ツ	U+FF70	一										

```

372 \clist_const:Nn \c__xeCJK_CJ_chars_clist
373 {
374     "3041 , "3043 , "3045 , "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E ,
375     "3095 , "3096 , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
376     "30E7 , "30EE , "30F5 , "30F6 , "30FC , "31F0 , "31F1 , "31F2 , "31F3 , "31F4 ,
377     "31F5 , "31F6 , "31F7 , "31F8 , "31F9 , "31FA , "31FB , "31FC , "31FD , "31FE ,
378     "31FF , "FF67 , "FF68 , "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F ,
379     "FF70
380 }

```

(End definition for \c__xeCJK_CJ_chars_clist.)

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

| U+FE6A | % | U+FF05 | % | U+FFE0 | ¢ |

```

381 \clist_const:Nn \c__xeCJK_PO_chars_clist { "FE6A , "FF05 , "FFE0 }

```

(End definition for \c__xeCJK_PO_chars_clist.)

\c__xeCJK_FullRight_chars_clist 以上六类标点符号出现在文字的右边,不应出现在行首位置。

```

382 \clist_const:Nx \c__xeCJK_FullRight_chars_clist
383 {
384     \c__xeCJK_CL_chars_clist ,
385     \c__xeCJK_NS_chars_clist ,
386     \c__xeCJK_EX_chars_clist ,
387     \c__xeCJK_IS_chars_clist ,
388     \c__xeCJK_CJ_chars_clist ,
389     \c__xeCJK_PO_chars_clist
390 }

```

(End definition for \c__xeCJK_FullRight_chars_clist.)

\c__xeCJK_CJK_chars_clist CJK 字符类,包括文字和标点符号。

```

391 \clist_const:Nn \c__xeCJK_CJK_chars_clist
392 {

```

- Hangul Jamo (谚文字母)

```

393     "1100 -> "11FF ,

```

- CJK Radicals Supplement (中日韩部首补充)

```

394     "2E80 -> "2EFF ,

```

- Kangxi Radicals (康熙字典部首)

```

395     "2F00 -> "2FDF ,

```

- Ideographic Description Characters (表意文字描述符)

```

396     "2FF0 -> "2FFF ,

```

- CJK Symbols and Punctuation (中日韩符号和标点)

```

397     "3000 -> "303F ,

```

- Hiragana (日文平假名)

```

398     "3040 -> "309F ,

```

- Katakana (日文片假名)

```

399     "30A0 -> "30FF ,

```

- Bopomofo (注音字母)

```

400     "3100 -> "312F ,

```

- Hangul Compatibility Jamo (谚文兼容字母)

```

401     "3130 -> "318F ,

```

- Kanbun (象形字注释标志)
402 "3190 -> "319F ,
- Bopomofo Extended (注音字母扩展)
403 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)
404 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)
405 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
406 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
407 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
408 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
409 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
410 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
411 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
412 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
413 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
414 "AC00 -> "D7AF ,
- Hangul Jamo Extended-B (谚文扩展 B)
415 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
416 "F900 -> "FAFF ,
- Vertical Forms (竖排形式)
417 "FE10 -> "FE1F ,
- CJK Compatibility Forms (中日韩兼容形式)
418 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)
419 "FF00 -> "FFEF ,
- Kana Supplement (日文假名增补)
420 "1B000 -> "1B0FF ,

- Enclosed Ideographic Supplement (带圈表意文字增补)

421 "1F200 -> "1F2FF ,

- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)

422 "20000 -> "2A6DF ,

- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)

423 "2A700 -> "2B73F ,

- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)

424 "2B740 -> "2B81F ,

- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)

425 "2F800 -> "2FA1F

426 }

(End definition for \c__xeCJK_CJK_chars_clist.)

\c__xeCJK_IVS_chars_clist 包括日文假名浊点和异体字选择符。

427 \clist_const:Nn \c__xeCJK_IVS_chars_clist

428 {

- 日文假名浊点

429 "3099 -> "309A ,

- Variation Selectors (异体字选择符)

430 "FE00 -> "FE0F ,

- Variation Selectors Supplement (异体字选择符增补)

431 "E0100 -> "E01EF

432 }

(End definition for \c__xeCJK_IVS_chars_clist.)

5.5 字符类别处理

```
\g__xeCJK_base_class_seq
\g__xeCJK_non_CJK_class_seq
\g__xeCJK_CJK_class_seq
433 \seq_new:N \g__xeCJK_base_class_seq
434 \seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq
435 \seq_new:N \g__xeCJK_non_CJK_class_seq
436 \seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq
437 { Default , HalfLeft , HalfRight , NormalSpace , Boundary }
438 \seq_new:N \g__xeCJK_CJK_class_seq
439 \cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
440 {
441   \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
442   \tl_const:cn { \__xeCJK_CJK_class_tl:c { \__xeCJK_class_csname:n {#1} } } {#1}
443 }
444 \clist_map_function:nN { CJK , FullLeft , FullRight , IVS } \__xeCJK_save_CJK_class:n
(End definition for \g__xeCJK_base_class_seq, \g__xeCJK_non_CJK_class_seq, and \g__xeCJK_CJK_class_seq.)
```

\xeCJK_class_num:n #1 为字符类别名称,用于取得字符类别对应的编号。

445 \cs_new_nopar:Npn \xeCJK_class_num:n #1 { \use:c { __xeCJK_class_csname:n {#1} } }

(End definition for \xeCJK_class_num:n.)

\xeCJKDeclareCharClass

```
446 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
447 {
448   \xeCJK_declare_char_class:nx {#2} {#3}
449   \IfBooleanT {#1} { \xeCJKResetPunctClass }
450 }
```

(End definition for `\xeCJKDeclareCharClass`. This function is documented on page 9.)

`\xeCJK_declare_char_class:nn` 用于设置字符所属的类别, #1 为类别名称, #2 为字符的 Unicode, 相邻字符用半角逗号隔开, 支持类似 "1100 -> "11FF 起止范围的使用方式。

`_xeCJK_set_char_class_aux:Nnw`

```

451 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
452 {
453   \clist_set:Nn \l__xeCJK_tmp_clist {#2}
454   \clist_gconcat:ccN
455     { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmp_clist
456   \clist_map_inline:Nn \l__xeCJK_tmp_clist
457     {
458       \str_if_eq:nnF {##1} { -> }
459       {
460         \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
461         { \xeCJK_class_num:n {#1} }
462       }
463     }
464   \xeCJK_set_char_class:nnn { "3099 } { "309A } { \xeCJK_class_num:n { IVS } }
465 }
466 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw
467 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
468 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
469 \cs_generate_variant:Nn \xeCJK_declare_char_class:nn { nx , nv }

```

(End definition for `\xeCJK_declare_char_class:nn` and `__xeCJK_set_char_class_aux:Nnw`.)

`_xeCJK_check_num_range:nnNN`

```

470 \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
471 {
472   \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
473     {
474       \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
475       \int_set_eq:NN #3 #4
476     }
477     {
478       \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
479       \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
480     }
481 }

```

(End definition for `__xeCJK_check_num_range:nnNN`.)

`\xeCJK_set_char_class:nnn` 设置字符类别, #1 和 #2 为字符类别起止的 Unicode, #3 为类别名称对应编号。

```

482 \cs_new_protected_nopar:Npn \xeCJK_set_char_class:nnn #1#2#3
483 {
484   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
485   \int_set:Nn \l__xeCJK_tmp_int {#3}
486   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
487   {
488     \XeTeXcharclass \l__xeCJK_begin_int = \l__xeCJK_tmp_int
489     \int_incr:N \l__xeCJK_begin_int
490   }
491 }

```

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

`_xeCJK_set_char_class_eq:nn`

将字符类 #1 中的字符全部设置成字符类 #2。只适用于 #1 的字符类范围为离散的逗号列表的情况。

```

492 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
493 {
494   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#2} }
495   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
496     { \XeTeXcharclass ##1 = \l__xeCJK_tmp_int }
497 }

```

(End definition for `__xeCJK_set_char_class_eq:nn`.)

`\normalspacedchars` 声明前后不加间距的字符。

```
498 \NewDocumentCommand \normalspacedchars { m }
499 {
500   \tl_map_inline:nn {#1}
501   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
502 }
```

(End definition for `\normalspacedchars`. This function is documented on page 9.)

`\xeCJKResetPunctClass` 用于重置标点符号所属的字符类。

```
503 \NewDocumentCommand \xeCJKResetPunctClass { }
504 {
505   \xeCJK_declare_char_class:nV { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
506   \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
507   \xeCJK_declare_char_class:nV { FullLeft } \c__xeCJK_FullLeft_chars_clist
508   \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
509 }
```

(End definition for `\xeCJKResetPunctClass`. This function is documented on page 9.)

`\xeCJKResetCharClass` 用于恢复 `\xeCJK` 对字符类别的设置。

```
510 \NewDocumentCommand \xeCJKResetCharClass { }
511 {
512   \xeCJK_declare_char_class:nV { CJK } \c__xeCJK_CJK_chars_clist
513   \xeCJK_declare_char_class:nV { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
514   \xeCJK_declare_char_class:nV { IVS } \c__xeCJK_IVS_chars_clist
515   \xeCJKResetPunctClass
516 }
```

(End definition for `\xeCJKResetCharClass`. This function is documented on page 9.)

设置字符类别。

```
517 \xeCJKResetCharClass
```

`\xeCJK_inter_class_toks:nnn` 在相邻类别之间插入内容。

```
518 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
519 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }
520 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc , nnx }
```

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

`\xeCJK_get_inter_class_toks:nn` 取出相邻类别之间的内容。

```
521 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2
522 { \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} }
```

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

`\xeCJK_clear_inter_class_toks:nn` 清除相邻类别之间的内容。

```
523 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2
524 { \xeCJK_inter_class_toks:nnn {#1} {#2} { \prg_do_nothing: } }
```

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

`\xeCJK_pre_inter_class_toks:nnn` 在相邻类别之间已有的内容前增加内容。

```
525 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
526 {
527   \xeCJK_inter_class_toks:nnx {#1} {#2}
528   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
529 }
530 \cs_generate_variant:Nn \xeCJK_pre_inter_class_toks:nnn { nnx }
```

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

`\xeCJK_app_inter_class_toks:nnn` 在相邻类别之间已有的内容后追加内容。

```
531 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
532 {
533   \xeCJK_inter_class_toks:nnx {#1} {#2}
534   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
535 }
536 \cs_generate_variant:Nn \xeCJK_app_inter_class_toks:nnn { nnc , nnx }
```


(End definition for \xeCJK_app_inter_class_toks:nnn.)

\xeCJK_copy_inter_class_toks:nnnn 将 #3 和 #4 之间的内容复制到 #1 和 #2 之间。

```
537 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
538 {
539   \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
540   \tl_if_empty:NF \l__xeCJK_tmp_tl
541   { \xeCJK_inter_class_toks:nx {#1} {#2} { \exp_not:V \l__xeCJK_tmp_tl } }
542 }
(End definition for \xeCJK_copy_inter_class_toks:nnnn.)
```

\xeCJK_replace_inter_class_toks:nnnn 将 #1 和 #2 之间出现的 #3 用 #4 替换。

```
543 \cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
544 {
545   \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
546   \tl_if_empty:NF \l__xeCJK_tmp_tl
547   {
548     \tl_replace_all:Nnn \l__xeCJK_tmp_tl {#3} {#4}
549     \xeCJK_inter_class_toks:nx {#1} {#2} { \exp_not:V \l__xeCJK_tmp_tl }
550   }
551 }
(End definition for \xeCJK_replace_inter_class_toks:nnnn.)
```

\xeCJK_clear_Boundary_and_CJK_toks: 清除边界与 CJK 文字、全角左右标点之间的内容。

```
552 \cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
553 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
554 \cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
555 { \xeCJK_clear_inter_class_toks:nn { Boundary } {#1} }
(End definition for \xeCJK_clear_Boundary_and_CJK_toks:.)
```

5.6 字符输出规则

	Default	CJK	FullL	FullR	HalfL	HalfR	Normal	Bound	IVS
Default		✓	✓	✓				✓	✓
CJK	✓	✓	✓	✓	✓	✓	✓	✓	
FullLeft	✓	✓	✓	✓	✓	✓	✓	✓	✓
FullRight	✓	✓	✓	✓	✓	✓	✓	✓	✓
HalfLeft		✓	✓	✓					✓
HalfRight		✓	✓	✓				✓	✓
NormalSpace		✓	✓	✓				✓	✓
Boundary	✓	✓	✓	✓	✓		✓		✓
IVS	✓	✓	✓	✓	✓	✓	✓	✓	✓

\xeCJK_class_group_begin:

\xeCJK_class_group_end:

```
556 \cs_new_eq:NN \xeCJK_class_group_begin: \c_group_begin_token
557 \cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token
```

(End definition for \xeCJK_class_group_begin: and \xeCJK_class_group_end:.)

IVS 字符类与 CJK 字符类基本相同,只是从 CJK 转移到 IVS 时,不加入任何内容。

```
558 \AtEndOfPackage
559 {
560   \seq_map_inline:Nn \g__xeCJK_class_seq
561   {
562     \str_if_eq:nnTF {#1} { IVS }
563     { \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } { CJK } }
564     {
565       \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } {#1}
566       \str_if_eq:nnF {#1} { CJK }
567       { \xeCJK_copy_inter_class_toks:nnnn {#1} { IVS } {#1} { CJK } }
568     }
569   }
570 }
```

```

571 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
572 {
573   \xeCJK_inter_class_toks:nnn {#1} { CJK }
574   {
575     \xeCJK_class_group_begin:
576     \xeCJK_select_font:
577     \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
578     \xeCJK_clear_Boundary_and_CJK_toks:
579     \CJKsymbol
580   }
581   \xeCJK_inter_class_toks:nnn { CJK } {#1} { \xeCJK_class_group_end: }
582 }
583 \clist_map_inline:nn { Default , HalfLeft , NormalSpace }
584 {
585   \xeCJK_inter_class_toks:nnn { Boundary } {#1} { \xeCJK_Boundary_and_Default: }
586   \str_if_eq:nnF {#1} { NormalSpace }
587   { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
588 }

```

`\xeCJK_Boundary_and_Default:`

```

589 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_Default:
590 {
591   \bool_if:nTF
592   {
593     \l__xeCJK_xecglue_bool &&
594     \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
595     \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
596   }
597   {
598     \tex_unskip:D
599     \bool_if:nTF
600     {
601       \xeCJK_if_last_node_p:n { CJK } ||
602       \xeCJK_if_last_node_p:n { CJK-space }
603     }
604     { \xeCJK_remove_node: \CJKecglue } { ~ }
605   }
606   {
607     \bool_if:nTF
608     {
609       \xeCJK_if_last_node_p:n { CJK } ||
610       \xeCJK_if_last_node_p:n { CJK-nobreak }
611     }
612     { \xeCJK_remove_node: \CJKecglue }
613     {
614       \xeCJK_if_last_node:nT { CJK-space }
615       { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
616     }
617   }
618 }

```

(End definition for `\xeCJK_Boundary_and_Default:`.)

```

619 \clist_map_inline:nn { Default , HalfRight , NormalSpace }
620 {
621   \xeCJK_inter_class_toks:nnn {#1} { Boundary }
622   {
623     \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
624     \peek_meaning_remove:NTF \tex_italiccorrection:D
625     { \tex_italiccorrection:D { \xeCJK_make_node:n { default } } }
626     {
627       \token_if_space:NTF \l_peek_token
628       { { \xeCJK_make_node:n { default-space } } }
629       { { \xeCJK_make_node:n { default } } }

```

```

630     }
631   }
632   \str_if_eq:nnF {#1} { NormalSpace }
633   { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
634 }
635 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
636 {
637   \xeCJK_check_for_glue:
638   \xeCJK_class_group_begin:
639   \xeCJK_clear_Boundary_and_CJK_toks:
640   \xeCJK_select_font:
641   \CJKsymbol
642 }

\xecjk_check_for_glue:

643 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
644 {
645   \bool_if:nTF
646   { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
647   { \xeCJK_remove_node: \CJKglue }
648   {
649     \xeCJK_if_last_node:nTF { CJK-nobreak }
650     { \xeCJK_remove_node: \xeCJK_no_break: \CJKglue }
651     {
652       \bool_if:nTF
653       {
654         \xeCJK_if_last_node_p:n { default } ||
655         \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
656       }
657       { \xeCJK_remove_node: \CJKecglue }
658       {
659         \bool_if:nT
660         {
661           \l__xeCJK_xecglue_bool &&
662           \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
663           ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
664             \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_ecglue_skip )
665         }
666         {
667           \tex_unskip:D
668           \bool_if:nTF
669           {
670             \xeCJK_if_last_node_p:n { default-space } ||
671             \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
672             \xeCJK_if_last_node_p:n { default }
673           }
674           { \xeCJK_remove_node: \CJKecglue }
675           {
676             \bool_if:nTF
677             {
678               \xeCJK_if_last_node_p:n { CJK } ||
679               \xeCJK_if_last_node_p:n { CJK-space }
680             }
681             {
682               \xeCJK_remove_node:
683               \bool_if:nTF \l__xeCJK_reserve_space_bool
684               { ~ } { \CJKglue }
685             }
686             { ~ }
687           }
688         }
689       }
690     }
691   }

```

```

690     }
691   }
692 }
(End definition for \xeCJK_check_for_glue:.)

\xecjk_if_last_node_p:n
\xecjk_if_last_node:nTF
693 \prg_new_conditional:Npnn \xecjk_if_last_node:n #1 { p , T , F , TF }
694 {
695   \if_dim:w \use:c { c__xecjk_#1_node_dim } = \tex_lastkern:D
696   \prg_return_true: \else: \prg_return_false: \fi:
697 }
(End definition for \xecjk_if_last_node:n.)

```

\xecjk_def_node:nn 用于判断插入的各种 kern。

\xecjk_make_node:n

```

698 \cs_new_protected_nopar:Npn \xecjk_def_node:nn #1#2
699 {
700   \dim_if_exist:cTF { c__xecjk_#1_node_dim }
701   { \dim_gset:cn } { \dim_const:cn }
702   { c__xecjk_#1_node_dim } {#2}
703 }
704 \cs_new_protected_nopar:Npn \xecjk_make_node:n #1
705 {
706   \tex_kern:D - \use:c { c__xecjk_#1_node_dim }
707   \tex_kern:D \use:c { c__xecjk_#1_node_dim }
708 }
709 \cs_new_protected_nopar:Npn \xecjk_remove_node:
710 { \tex_unkern:D \tex_unkern:D }
711 \xecjk_def_node:nn { CJK } { 11 sp }
712 \xecjk_def_node:nn { CJK-space } { 12 sp }
713 \xecjk_def_node:nn { default } { 13 sp }
714 \xecjk_def_node:nn { default-space } { 14 sp }
715 \xecjk_def_node:nn { CJK-nobreak } { 15 sp }
(End definition for \xecjk_def_node:nn and \xecjk_make_node:n.)

```

CJKglue CJK 文字之间插入的 glue。

```

716 \keys_define:nn { xecjk / options }
717 {
718   CJKglue .code:n =
719   {
720     \cs_set_protected_nopar:Npn \CJKglue {#1}
721     \xecjk_glue_to_skip:nN {#1} \l__xecjk_ccglue_skip
722   }
723 }
724 \skip_new:N \l__xecjk_ccglue_skip
(End definition for CJKglue. This function is documented on page 3.)

```

CJKecglue CJK 与西文和数学行内数学公式之间自动添加的空白。

xCJKecglue

```

725 \keys_define:nn { xecjk / options }
726 {
727   CJKecglue .code:n =
728   {
729     \cs_set_protected_nopar:Npn \CJKecglue {#1}
730     \xecjk_glue_to_skip:nN {#1} \l__xecjk_ecglue_skip
731   } ,
732   xCJKecglue .choice: ,
733   xCJKecglue / true .code:n =
734   {
735     \bool_set_true:N \l__xecjk_xecglue_bool
736     \cs_set_eq:NN \xecjk_space_or_xecglue: \CJKecglue
737   } ,
738   xCJKecglue / false .code:n =
739   {

```

```

740     \bool_set_false:N \l__xeCJK_xecglue_bool
741     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_tl
742   } ,
743   xCJKecglue / unknown .code:n =
744   {
745     \bool_set_true:N \l__xeCJK_xecglue_bool
746     \cs_set_protected_nopar:Npn \CJKecglue {#1}
747     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
748     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
749   } ,
750   xCJKecglue .default:n = { true }
751 }
752 \skip_new:N \l__xeCJK_ecglue_skip
753 \bool_new:N \l__xeCJK_xecglue_bool
(End definition for CJKecglue and xCJKecglue. These functions are documented on page 3.)

```

CJKspace 是否保留 CJK 文字间的空白, 默认不保留。

```

754 \keys_define:nn { xeCJK / options }
755 {
756   CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
757   space .meta:n = { CJKspace = true } ,
758   nospace .meta:n = { CJKspace = false }
759 }

```

(End definition for CJKspace. This function is documented on page 3.)

```

760 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
761 {
762   \xeCJK_class_group_end:
763   { \xeCJK_make_node:n { CJK } }
764   \xeCJK_ignore_spaces:w
765 }

```

\xeCJK_ignore_spaces:w

```

766 \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
767 {
768   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
769   {
770     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
771     { \xeCJK_space_or_xecglue: } { \CJKecglue }
772   }
773   {
774     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
775     {
776       \xeCJK_remove_node:
777       { \xeCJK_make_node:n { CJK-space } }
778       \bool_if:nT
779       {
780         \token_if_macro_p:N \l_peek_token ||
781         ( \l__xeCJK_reserve_space_bool &&
782           \token_if_letter_p:N \l_peek_token )
783       }
784       { \xeCJK_space_or_xecglue: }
785     }
786     { \__xeCJK_CJK_and_Boundary_aux: }
787   }
788 }
789 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_Boundary_aux:
790 { \cs_if_exist:NF \l_peek_token \exp_not:N }
(End definition for \xeCJK_ignore_spaces:w.)

```

```

791 \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }

```

\xeCJK_CJK_and_CJK:N

```

792 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }

```

(End definition for `\xeCJK_CJK_and_CJK:N`)

```

793 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
794   { \xeCJK_FullLeft_and_CJK: \CJKsymbol }
795 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
796   { \xeCJK_FullRight_and_CJK: \CJKsymbol }
797 \seq_map_inline:Nn \g_--xeCJK_non_CJK_class_seq
798 {
799   \clist_map_inline:nn { FullLeft , FullRight }
800   {
801     \xeCJK_inter_class_toks:nnx {#1} {##1}
802     { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
803     \xeCJK_inter_class_toks:nnc {##1} {#1} { xeCJK_##1_and_Default: }
804   }
805 }
806 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
807   { \xeCJK_Boundary_and_FullLeft:N }
808 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
809   { \xeCJK_Boundary_and_FullRight:N }

```

`\xeCJK_FullRight_and_Boundary:`

```

810 \xeCJK_app_inter_class_toks:nnn { FullLeft } { Boundary } { \tex_ignorespaces:D }
811 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
812   { \xeCJK_FullRight_and_Boundary: }
(End definition for \xeCJK_FullRight_and_Boundary:.)

```

`\xeCJK_FullRight_and_Boundary:`

```

813 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Boundary:
814   { \xeCJK_FullRight_and_Default: \tex_ignorespaces:D }
(End definition for \xeCJK_FullRight_and_Boundary:.)

```

```

815 \clist_map_inline:nn { CJK , FullLeft , FullRight }
816   {
817     \clist_map_inline:nn { FullLeft , FullRight }
818       { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
819   }

```

`_xeCJK_punct_rule:NN` 用于抹去标点符号的左/右空白。

```

820 \cs_new_protected_nopar:Npn \_xeCJK_punct_rule:NN #1#2
821   {
822     \tex_vrule:D
823     width - \_xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
824     depth \c_zero_dim
825     height \c_zero_dim \scan_stop:
826   }
(End definition for \_xeCJK_punct_rule:NN.)

```

`_xeCJK_punct_glue:NN` 根据所选的标点处理方式在标点符号左/右增加的空白。

```

827 \cs_new_protected_nopar:Npn \_xeCJK_punct_glue:NN #1#2
828   {
829     \_xeCJK_punct_hskip:n
830     {
831       \_xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
832       minus \dim_eval:n { ( \_xeCJK_use_punct_dim:nnn { glue } {#1} {#2} ) / \c_two }
833     }
834   }
835 \cs_new_eq:NN \_xeCJK_punct_hskip:n \skip_horizontal:n
(End definition for \_xeCJK_punct_glue:NN.)

```

`_xeCJK_punct_kern:NN` 相邻两个标点之间的间距。

```

836 \cs_new_protected_nopar:Npn \_xeCJK_punct_kern:NN #1#2
837   { \tex_kern:D \_xeCJK_use_punct_dim:nnn { kern } {#1} {#2} }
(End definition for \_xeCJK_punct_kern:NN.)

```


\g__xeCJK_last_punct_tl 用于记录当前的标点符号。

```

838 \tl_new:N \g__xeCJK_last_punct_tl
(End definition for \g__xeCJK_last_punct_tl.)

```

\xeCJK_FullLeft_and_CJK:

```

839 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_CJK:
840 {
841   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
842   {
843     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
844     \xeCJK_no_break:
845     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
846   }
847   { \xeCJK_no_break: }
848 }
(End definition for \xeCJK_FullLeft_and_CJK:.)

```

\xeCJK_FullLeft_and_Default:

```

849 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_Default:
850 {
851   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
852   {
853     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
854     \xeCJK_class_group_end: \xeCJK_no_break:
855     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
856   }
857   { \xeCJK_class_group_end: \xeCJK_no_break: \__xeCJK_zero_glue: }
858 }
859 \cs_new_protected_nopar:Npn \__xeCJK_zero_glue:
860 { \skip_horizontal:N \c_zero_skip }
(End definition for \xeCJK_FullLeft_and_Default:.)

```

\xeCJK_FullRight_and_CJK:

```

861 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_CJK:
862 {
863   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
864   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
865   \CJKglue
866 }
(End definition for \xeCJK_FullRight_and_CJK:.)

```

\xeCJK_FullRight_and_Default:

```

867 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Default:
868 {
869   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
870   \xeCJK_class_group_end:
871   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
872 }
(End definition for \xeCJK_FullRight_and_Default:.)

```

\xeCJK_Default_and_FullLeft:nN

```

873 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullLeft:nN #1#2
874 {
875   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#2}
876   \__xeCJK_Default_and_FullLeft_glue:N {#2}
877   \xeCJK_class_group_begin:
878   \xeCJK_select_font:
879   \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
880   \xeCJK_clear_Boundary_and_CJK_toks:
881   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
882   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2}
883   \CJKpunctsymbol {#2}

```

```

884 }
885 \cs_new_protected_nopar:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
886 { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_Default_and_FullLeft:nN.)

```

\xeCJK_CJK_and_FullLeft:N

```

887 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
888 {
889   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
890   \__xeCJK_CJK_and_FullLeft_glue:N {#1}
891   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
892   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
893   \CJKpunctsymbol {#1}
894 }
895 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
896 { \CJKglue \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_CJK_and_FullLeft:N.)

```

\xeCJK_Boundary_and_FullLeft:N

```

897 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullLeft:N #1
898 {
899   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
900   \__xeCJK_Boundary_and_FullLeft_glue:N {#1}
901   \xeCJK_class_group_begin:
902   \xeCJK_select_font:
903   \xeCJK_clear_Boundary_and_CJK_toks:
904   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
905   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
906   \CJKpunctsymbol {#1}
907 }
(End definition for \xeCJK_Boundary_and_FullLeft:N.)

```

__xeCJK_Boundary_and_FullLeft_glue:N

\etex_lastnodetype:D 为 1 表示 hlist node, 在这里用来判断是否位于段首。基于正常情况下, \TeX 会在段落开头插入宽度为 \parindent 的水平盒子用于缩进。-1 表示 empty list, 常出现在盒子的起始位置, 在段落前使用 \noindent 就是这种情况。11 表示 glue node, 这里判断的目的是当全角左标点出现在 \LaTeX 表格的非 p 列行首时, 能够对齐到单元格的边界。判断基于标准 \LaTeX 表格的列格式(\@tabclassz)定义中, 在 l 列和 r 列前为了防止 \tabcolsep 被无意 \unskip 掉, 都加了 \hskip1sp, 而 c 列前则有 \hfil。13 表示 penalty node, 这里判断的目的是全角左标点出现在 \LaTeX 列表环境的 \item 后面时, 能对齐到边界。判断基于 \item 的内部定义 \@item 对 \everypar 进行了修改, 在这里起到影响作用的是 \box\@labels \penalty\z@。以上判断都比较粗略, 暂时也没有想起更好的办法。

```

908 \cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
909 {
910   \int_case:nnn { \etex_lastnodetype:D }
911   {
912     { \c_one } { \__xeCJK_zero_glue: \use_none:n }
913     { \c_minus_one } { \use_none:n }
914     { \c_eleven }
915     {
916       \bool_if:nF
917       {
918         ! ( \skip_if_finite_p:n { \tex_lastskip:D } ) ||
919         \skip_if_eq_p:nn { \tex_lastskip:D } { 1 sp }
920       }
921     }
922     { \c_thirteen }
923     {
924       \int_compare:nNnTF \tex_lastpenalty:D = \c_zero
925       {
926         \tex_unpenalty:D
927         \int_compare:nNnTF \etex_lastnodetype:D = \c_one

```

```

928         { \tex_penalty:D \c_zero \use_none:n }
929         { \tex_penalty:D \c_zero \use:n }
930     }
931     { \use:n }
932 }
933 }
934 { \use:n }
935 { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
936 }

```

(End definition for __xeCJK_Boundary_and_FullLeft_glue:N.)

\xeCJK_Default_and_FullRight:nN

```

937 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullRight:nN #1#2
938 {
939     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#2}
940     \__xeCJK_Default_and_FullRight_glue:N {#2}
941     \xeCJK_class_group_begin:
942     \xeCJK_select_font:
943     \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
944     \xeCJK_clear_Boundary_and_CJK_toks:
945     \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
946     \__xeCJK_punct_if_middle:NT {#2}
947     { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2} }
948     \xeCJK_FullRight_symbol:N {#2}
949 }

```

(End definition for \xeCJK_Default_and_FullRight:nN.)

\xeCJK_Boundary_and_FullRight:N

```

950 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullRight:N #1
951 {
952     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
953     \__xeCJK_Default_and_FullRight_glue:N {#1}
954     \xeCJK_class_group_begin:
955     \xeCJK_select_font:
956     \xeCJK_clear_Boundary_and_CJK_toks:
957     \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
958     \__xeCJK_punct_if_middle:NT {#1}
959     { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
960     \xeCJK_FullRight_symbol:N {#1}
961 }

```

(End definition for \xeCJK_Boundary_and_FullRight:N.)

\xeCJK_CJK_and_FullRight:N

```

962 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
963 {
964     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
965     \__xeCJK_CJK_and_FullRight_glue:N {#1}
966     \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
967     \__xeCJK_punct_if_middle:NT {#1}
968     { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
969     \xeCJK_FullRight_symbol:N {#1}
970 }

```

(End definition for \xeCJK_CJK_and_FullRight:N.)

__xeCJK_CJK_and_FullRight_glue:N

__xeCJK_Default_and_FullRight_glue:N

```

971 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
972 {
973     \__xeCJK_punct_if_long:NTF {#1}
974     { \CJKglue }
975     {
976         \xeCJK_no_break:
977         \__xeCJK_punct_if_middle:NT {#1}

```

```

978     { \_xeCJK_punct_glue:NN \c\_xeCJK_right_tl {#1} }
979   }
980 }
981 \cs_new_eq:NN \_xeCJK_Default_and_FullRight_glue:N \_xeCJK_CJK_and_FullRight_glue:N
(End definition for \_xeCJK_CJK_and_FullRight_glue:N and \_xeCJK_Default_and_FullRight_glue:N)

```

\xeCJK_FullLeft_and_FullLeft:N

```

982 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
983 {
984   \xeCJK_no_break:
985   \xeCJK_get_punct_bounds:NN \c\_xeCJK_left_tl {#1}
986   \xeCJK_get_punct_kerning:oN \g\_xeCJK_last_punct_tl {#1}
987   \_xeCJK_punct_kern:NN \g\_xeCJK_last_punct_tl {#1}
988   \tl_gset:Nx \g\_xeCJK_last_punct_tl {#1}
989   \CJKpunctsymbol {#1}
990 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N)

```

\xeCJK_FullLeft_and_FullRight:N

```

991 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
992 {
993   \xeCJK_no_break:
994   \xeCJK_get_punct_bounds:NN \c\_xeCJK_right_tl {#1}
995   \xeCJK_get_punct_kerning:oN \g\_xeCJK_last_punct_tl {#1}
996   \_xeCJK_punct_kern:NN \g\_xeCJK_last_punct_tl {#1}
997   \tl_gset:Nx \g\_xeCJK_last_punct_tl {#1}
998   \xeCJK_no_break:
999   \xeCJK_FullRight_symbol:N {#1}
1000 }
(End definition for \xeCJK_FullLeft_and_FullRight:N)

```

\xeCJK_FullRight_and_FullLeft:N

```

1001 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
1002 {
1003   \xeCJK_get_punct_bounds:NN \c\_xeCJK_left_tl {#1}
1004   \xeCJK_get_punct_kerning:oN \g\_xeCJK_last_punct_tl {#1}
1005   \xeCJK_punct_kern:NN \g\_xeCJK_last_punct_tl {#1}
1006   \tl_gset:Nx \g\_xeCJK_last_punct_tl {#1}
1007   \CJKpunctsymbol {#1}
1008 }
(End definition for \xeCJK_FullRight_and_FullLeft:N)

```

_xeCJK_punct_nobreak_kern:NN

```

1009 \cs_new_protected_nopar:Npn \_xeCJK_punct_nobreak_kern:NN #1#2
1010 {
1011   \_xeCJK_punct_kern:NN #1#2
1012   \xeCJK_no_break:
1013 }
1014 \cs_new_eq:NN \xeCJK_punct_kern:NN \_xeCJK_punct_nobreak_kern:NN
(End definition for \_xeCJK_punct_nobreak_kern:NN)

```

_xeCJK_punct_breakable_kern:NN

```

1015 \cs_new_protected_nopar:Npn \_xeCJK_punct_breakable_kern:NN #1#2
1016 {
1017   \_xeCJK_punct_rule:NN \c\_xeCJK_right_tl #1
1018   \_xeCJK_punct_hskip:n
1019   {
1020     \_xeCJK_use_punct_dim:nnn { kern } {#1} {#2} +
1021     \_xeCJK_use_punct_dim:nnn { bound } { \c\_xeCJK_right_tl } {#1} +
1022     \_xeCJK_use_punct_dim:nnn { bound } { \c\_xeCJK_left_tl } {#2}
1023   }
1024   \_xeCJK_punct_rule:NN \c\_xeCJK_left_tl #2
1025 }

```

(End definition for _xeCJK_punct_breakable_kern:NN)

\xeCJK_FullRight_and_FullRight:N

```
1026 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
1027 {
1028   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
1029   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
1030   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
1031   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1032   \xeCJK_no_break:
1033   \xeCJK_FullRight_symbol:N {#1}
1034 }
```

(End definition for \xeCJK_FullRight_and_FullRight:N)

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```
1035 \keys_define:nn { xecjk / options }
1036 {
1037   CheckFullRight .choice: ,
1038   CheckFullRight / true .code:n =
1039   {
1040     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
1041     {
1042       \cs_set_eq:NN \__xeCJK_save_FullRight_check: \xeCJK_FullRight_and_Boundary:
1043       \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N \xeCJK_FullRight_symbol:N
1044       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
1045       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \xeCJK_check_FullRight_symbol:Nw
1046     }
1047   } ,
1048   CheckFullRight / false .code:n =
1049   {
1050     \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
1051     {
1052       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \__xeCJK_save_FullRight_check:
1053       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \__xeCJK_save_FullRight_symbol:N
1054     }
1055   } ,
1056   CheckFullRight .default:n = { true }
1057 }
```

(End definition for *CheckFullRight*. This function is documented on page 5.)

\xeCJK_FullRight_symbol:N

```
1058 \cs_new_nopar:Npn \xeCJK_FullRight_symbol:N { \CJKpunctsymbol }
(End definition for \xeCJK_FullRight_symbol:N)
```

\xeCJK_check_FullRight:

```
1059 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight:
1060 {
1061   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1062   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1063   \tl_case:Non \l_peek_token { \l__xeCJK_no_break_cs_case_tl } { }
1064   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1065   \xeCJK_class_group_end:
1066 }
1067 \cs_generate_variant:Nn \tl_case:Nnn { No }
(End definition for \xeCJK_check_FullRight:.)
```

\xeCJK_check_FullRight_symbol:Nw

```
1068 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight_symbol:Nw #1
1069 { \xeCJK_peek_after_ignore_spaces:nw { \__xeCJK_save_FullRight_symbol:N {#1} } }
(End definition for \xeCJK_check_FullRight_symbol:Nw)
```

\xeCJK_cs_case_keys_define:nNNnn

```
1070 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
1071 {
1072   \tl_new:N #2
1073   \seq_new:N #3
1074   \keys_define:nn { xecjk / options }
1075   {
1076     #1 .code:n =
1077     {
1078       \seq_set_split:Nnn #3 { } {##1}
1079       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1080     } ,
1081     #1+ .code:n =
1082     {
1083       \tl_map_inline:nn {##1}
1084       { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
1085       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1086     } ,
1087     #1- .code:n =
1088     {
1089       \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {####1} }
1090       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1091     }
1092   }
1093 }
1094 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
1095 {
1096   \tl_clear:N #1
1097   \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
1098   #4
1099 }
```

(End definition for \xeCJK_cs_case_keys_define:nNNnn.)

NoBreakCS 设置不能在全角右标点之后断行的控制序列。

```
1100 \xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
1101 \l__xeCJK_no_break_cs_case_tl \l__xeCJK_no_break_cs_seq { \xeCJK_no_break: } { }
```

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

\xeCJKnobreak 为保险起见,我们在这里用了一个循环。

```
1102 \NewDocumentCommand \xeCJKnobreak { }
1103 {
1104   \bool_set_true:N \l__xeCJK_tmp_bool
1105   \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
1106   {
1107     \bool_if:NTF \l__xeCJK_tmp_bool
1108     {
1109       \bool_set_false:N \l__xeCJK_tmp_bool
1110       \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
1111     }
1112     { \skip_add:Nn \l__xeCJK_tmp_skip \tex_lastskip:D }
1113     \tex_unskip:D
1114   }
1115   \xeCJK_no_break:
1116   \bool_if:NF \l__xeCJK_tmp_bool { \skip_horizontal:N \l__xeCJK_tmp_skip }
1117 }
```

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

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```
1118 \keys_define:nn { xecjk / options }
1119 {
```

```

1120     CheckSingle .choice: ,
1121     CheckSingle / true .code:n =
1122     {
1123         \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1124         {
1125             \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
1126             \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1127         }
1128     } ,
1129     CheckSingle / false .code:n =
1130     {
1131         \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1132         { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
1133     } ,
1134     CheckSingle .default:n = { true } ,
1135     CJKchecksingle .meta:n = { CheckSingle = true }
1136 }

```

(End definition for *CheckSingle*. This function is documented on page 3.)

\xeCJK_check_single:Nw

```

1137 \cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
1138 {
1139     \peek_catcode:NTF \c_catcode_letter_token
1140     { \xeCJK_check_single:NNw #1 }
1141     {
1142         \token_if_other:NTF \l_peek_token
1143         { \xeCJK_check_single:NNw #1 }
1144         {
1145             \bool_if:NTF
1146             {
1147                 \xeCJK_if_blank_x_p:n { \token_get_arg_spec:N \l_peek_token } &&
1148                 \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
1149                 ( \exp_after:wN \token_if_other_p:N \l_peek_token ||
1150                   \exp_after:wN \token_if_letter_p:N \l_peek_token )
1151             }
1152             { \exp_after:wN \xeCJK_check_single:NNw \exp_after:wN #1 }
1153             { \__xeCJK_check_single_save:N #1 }
1154         }
1155     }
1156 }

```

(End definition for *\xeCJK_check_single:Nw*.)

\xeCJK_check_single:NNw 使用 \group_align_safe_begin: 和 \group_align_safe_end: 是为了防止在表格里面报错。

```

1157 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
1158 {
1159     \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
1160     {
1161         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1162         { \__xeCJK_check_single_space:NN #1#2 }
1163         { \__xeCJK_check_single_save:N #1 #2 }
1164     }
1165     {
1166         \group_align_safe_begin:
1167         \token_if_other:NTF \l_peek_token
1168         {
1169             \group_align_safe_end:
1170             \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1171             { \__xeCJK_check_single_space:NN #1#2 }
1172             { \__xeCJK_check_single_save:N #1 #2 }
1173         }
1174         {
1175             \token_if_cs:NTF \l_peek_token

```

```

1176         {
1177             \group_align_safe_end:
1178             \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1179             { \xeCJK_check_single_cs:NNn #1#2 { ~ } }
1180             { \xeCJK_check_single_cs:NNn #1#2 { } }
1181         }
1182         {
1183             \group_align_safe_end:
1184             \bool_if:nTF
1185             {
1186                 \l__xeCJK_plain_equation_bool &&
1187                 \token_if_math_toggle_p:N \l_peek_token
1188             }
1189             {
1190                 \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1191                 { \xeCJK_check_single_equation:NNnNw #1 #2 { ~ } }
1192                 { \xeCJK_check_single_equation:NNnNw #1 #2 { } }
1193             }
1194             {
1195                 \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1196                 { \__xeCJK_check_single_save:N #1 #2 ~ }
1197                 { \__xeCJK_check_single_save:N #1 #2 }
1198             }
1199         }
1200     }
1201 }
1202 }

```

(End definition for `\xeCJK_check_single:NNu`.)

`__xeCJK_check_single_space:NN`

```

1203 \cs_new_protected_nopar:Npn \__xeCJK_check_single_space:NN #1#2
1204 {
1205     \xeCJK_if_CJK_class:NTF #2
1206     {
1207         \xeCJK_if_CJK_class:NTF \l_peek_token
1208         { \__xeCJK_check_single_save:N #1 #2 }
1209         { \__xeCJK_check_single_save:N #1 #2 ~ }
1210     }
1211     { \__xeCJK_check_single_save:N #1 #2 ~ }
1212 }

```

(End definition for `__xeCJK_check_single_space:NN`.)

`\xeCJK_check_single_equation:NNnNw`

```

1213 \cs_new_protected_nopar:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
1214 {
1215     \peek_catcode:NTF \c_math_toggle_token
1216     {
1217         \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1218         \xeCJK_make_node:n { CJK-nobreak } #2 #4
1219     }
1220     { \__xeCJK_check_single_save:N #1 #2#3#4 }
1221 }

```

(End definition for `\xeCJK_check_single_equation:NNnNw`.)

`\xeCJK_check_single_cs:NNn` 在使用 CheckSingle 选项时,在 `tablists` 宏包定义的 `tabenum` 环境中会出现下面的错误:

```

! Forbidden control sequence found while scanning use of \use_ii:nn.
<inserted text>
\par
1.10 \item

```


原因在于 `tabenum` 实际上是一个 `TEX` 对齐环境(`\halign`), `\par` 在其中被重定义为 `\cr`。而在下面 `\tl_case:Nnn` 的分支里有对 `\par` 的 `\ifx` 判断。解决办法是将判断用 `\group_align_safe_begin:` 和 `\group_align_safe_end:` 包起来。或者改用原语 `\tex_par:D` 作为判断条件。

```

1222 \cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
1223 {
1224   \group_align_safe_begin:
1225   \tl_case:Non \l_peek_token
1226     { \l__xeCJK_check_single_cs_case_tl }
1227     { \group_align_safe_end: \use_iii:nnn }
1228     { \xeCJK_check_single_env:nnNn }
1229     {
1230       \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1231       \xeCJK_make_node:n { CJK-nobreak } #2
1232     }
1233     { \__xeCJK_check_single_save:N #1 #2#3 }
1234   }
1235 \tl_new:N \l__xeCJK_check_single_cs_case_tl
(End definition for \xeCJK_check_single_cs:NNn.)

```

`\xeCJK_check_single_env:nnNn`

```

1236 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
1237 {
1238   \group_align_safe_begin:
1239   \str_case:x:non {#4}
1240     { \l__xeCJK_inline_env_case_tl }
1241     { \group_align_safe_end: \use_i:nn }
1242     {#1} {#2}
1243   \scan_stop: #3 {#4}
1244 }
1245 \cs_generate_variant:Nn \str_case_x:nnn { no }
(End definition for \xeCJK_check_single_env:nnNn.)

```

NewLineCS

```

1246 \xeCJK_cs_case_keys_define:nNNnn { NewLineCS }
1247 \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
1248 { \group_align_safe_end: \use_ii:nnn }
1249 {
1250   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1251   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1252 }
(End definition for NewLineCS. This function is documented on page 3.)

```

EnvCS

```

1253 \xeCJK_cs_case_keys_define:nNNnn { EnvCS }
1254 \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
1255 { \group_align_safe_end: \use:n }
1256 {
1257   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1258   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1259 }
(End definition for EnvCS. This function is documented on page 3.)

```

InlineEnv

```

1260 \keys_define:nn { xeCJK / options }
1261 {
1262   InlineEnv .code:n =
1263     {
1264       \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
1265       \__xeCJK_update_inline_env_case_tl:
1266     } ,
1267   InlineEnv+ .code:n =
1268     {

```

```

1269     \clist_map_inline:nn {#1}
1270     {
1271         \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
1272         { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
1273     }
1274     \__xeCJK_update_inline_env_case_tl:
1275 } ,
1276 InlineEnv- .code:n =
1277 {
1278     \clist_map_inline:nn {#1}
1279     { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
1280     \__xeCJK_update_inline_env_case_tl:
1281 }
1282 }
1283 \seq_new:N \l__xeCJK_inline_env_seq
(End definition for InlineEnv. This function is documented on page 4.)

```

_xeCJK_update_inline_env_case_tl:

```

1284 \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
1285 {
1286     \tl_clear:N \l__xeCJK_inline_env_case_tl
1287     \seq_map_inline:Nn \l__xeCJK_inline_env_seq
1288     { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { \use_ii:nn } } }
1289 }
1290 \tl_new:N \l__xeCJK_inline_env_case_tl
(End definition for \__xeCJK_update_inline_env_case_tl:.)

```

PlainEquation

```

1291 \keys_define:nn { xeCJK / options }
1292 { PlainEquation .bool_set:N = \l__xeCJK_plain_equation_bool }
(End definition for PlainEquation. This function is documented on page 3.)

```

5.9 增加 CJK 子分区

\g__xeCJK_CJK_sub_class_seq

```

1293 \seq_new:N \g__xeCJK_CJK_sub_class_seq
(End definition for \g__xeCJK_CJK_sub_class_seq.)

```

\xeCJKDeclareSubCJKBlock

声明 CJK 子区范围, #1 为自定义名称, #2 为子区的 Unicode 范围。

```

1294 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
1295 { s > { \TrimSpaces } m > { \TrimSpaces } m }
1296 {
1297     \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
1298     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1299 }
1300 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock. This function is documented on page 9.)

```

\xeCJKCancelSubCJKBlock

取消和恢复对 CJK 子区的声明。

\xeCJKRestoreSubCJKBlock

```

1301 \bool_new:N \l__xeCJK_sub_cancel_bool
1302 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
1303 {
1304     \bool_if:NF \l__xeCJK_sub_cancel_bool
1305     {
1306         \bool_set_true:N \l__xeCJK_sub_cancel_bool
1307         \__xeCJK_sub_restore_or_cancel:x {#2}
1308         \IfBooleanT {#1} { \xeCJKResetPunctClass }
1309     }
1310 }
1311 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
1312 {
1313     \bool_if:NT \l__xeCJK_sub_cancel_bool

```

```

1314     {
1315         \bool_set_false:N \l__xeCJK_sub_cancel_bool
1316         \__xeCJK_sub_restore_or_cancel:x {#2}
1317         \IfBooleanT {#1} { \xeCJKResetPunctClass }
1318     }
1319 }

```

(End definition for \xeCJKCancelSubCJKBlock and \xeCJKRestoreSubCJKBlock. These functions are documented on page 9.)

__xeCJK_sub_restore_or_cancel:n

```

1320 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
1321 {
1322     \clist_map_inline:nn {#1}
1323     {
1324         \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
1325         {
1326             \xeCJK_declare_char_class:nx
1327             { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
1328             { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
1329         }
1330         { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
1331     }
1332 }
1333 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
1334 \__xeCJK_msg_new:nn { SubBlock-undefined }
1335 {
1336     The~CJK~sub~block~`#1'~is~undefined.\\\
1337     Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
1338     to~declare~it.
1339 }

```

(End definition for __xeCJK_sub_restore_or_cancel:n.)

\xeCJK_declare_sub_char_class:nnn

```

1340 \cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
1341 {
1342     \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
1343     {
1344         \xeCJK_new_class:n { #1/#2 }
1345         \__xeCJK_set_sub_class_toks:nn {#1} {#2}
1346         \xeCJK_new_sub_key:n {#2}
1347     }
1348     \xeCJK_declare_char_class:nn { #1/#2 } {#3}
1349 }
1350 \cs_generate_variant:Nn \xeCJK_declare_sub_char_class:nnn { nxx }

```

(End definition for \xeCJK_declare_sub_char_class:nnn.)

__xeCJK_set_sub_class_toks:nn

```

1351 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
1352 {
1353     \seq_map_inline:Nn \g__xeCJK_base_class_seq
1354     {
1355         \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
1356         \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
1357         \str_if_eq:nnTF {##1} { CJK }
1358         {
1359             \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
1360             { \__xeCJK_switch_font:nn {#1} {#2} }
1361         }
1362         {
1363             \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
1364             { \CJKsymbol }
1365             { \__xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
1366         }
1367     }

```

```

1367     }
1368     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
1369     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
1370     {
1371         \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
1372         \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
1373         \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
1374         { \__xeCJK_switch_font:nn {#2} {##1} }
1375         \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
1376         { \__xeCJK_switch_font:nn {##1} {#2} }
1377     }
1378     \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
1379     \__xeCJK_save_CJK_class:n { #1/#2 }
1380     \clist_map_inline:nn { CJK , FullLeft , FullRight }
1381     {
1382         \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
1383         { \__xeCJK_switch_font:nn {#2} {#1} }
1384     }
1385 }
(End definition for \__xeCJK_set_sub_class_toks:nn)

```

5.10 标点处理

`\XeTeXglyphbounds` 可以得到一个字符的左右边距,用于标点压缩。如果它不可用,则在文档中只能使用 `plain` 这一标点格式原样输出标点。

```

1386 \__xeCJK_msg_new:nn { XeTeX-too-old }
1387 {
1388     \token_to_str:N \XeTeXglyphbounds \ is-not-defined.\\
1389     CJK-punctuation-kerning~will~not~be~available.\\
1390     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
1391 }
1392 \cs_if_exist:NF \XeTeXglyphbounds
1393 {
1394     \__xeCJK_error:n { XeTeX-too-old }
1395     \__xeCJK_after_end_preamble:n { \xeCJKsetup { PunctStyle = plain } }
1396 }

```

`\xeCJKsetwidth` 手动设置参数中的标点符号的宽度。

```

1397 \NewDocumentCommand \xeCJKsetwidth { m m }
1398 { \tl_map_inline:xn {#1} { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#2} } }
1399 \cs_generate_variant:Nn \tl_map_inline:nn { x }
(End definition for \xeCJKsetwidth. This function is documented on page 10.)

```

`\xeCJKsetkern` 手动设置相邻标点的距离。

```

1400 \NewDocumentCommand \xeCJKsetkern { m m m }
1401 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
(End definition for \xeCJKsetkern. This function is documented on page 10.)

```

```

\c__xeCJK_left_tl
\c__xeCJK_right_tl
1402 \tl_const:Nn \c__xeCJK_left_tl { left }
1403 \tl_const:Nn \c__xeCJK_right_tl { right }
(End definition for \c__xeCJK_left_tl and \c__xeCJK_right_tl.)

```

`AllowBreakBetweenPuncts` 相关选项声明。

```

KaiMingPunct 1404 \keys_define:nn { xeCJK / options }
LongPunct    1405 {
MiddlePunct  1406     AllowBreakBetweenPuncts .choice: ,
PunctWidth   1407     AllowBreakBetweenPuncts / true .code:n =
1408     {
1409         \bool_set_true:N \l__xeCJK_punct_breakable_bool
1410         \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
1411     } ,

```

```

1412 AllowBreakBetweenPuncts / false .code:n =
1413 {
1414     \bool_set_false:N \l__xeCJK_punct_breakable_bool
1415     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_nobreak_kern:NN
1416 } ,
1417 AllowBreakBetweenPuncts .default:n = { true } ,
1418 KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
1419 KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
1420 KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
1421 LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
1422 LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
1423 LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
1424 MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
1425 MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
1426 MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
1427 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl
1428 }
1429 \bool_new:N \l__xeCJK_punct_breakable_bool
(End definition for AllowBreakBetweenPuncts and others. These functions are documented on page 4.)
相关选项定义的辅助函数。
1430 \clist_new:N \g__xeCJK_special_punct_clist
1431 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
1432 \cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
1433 \cs_new_nopar:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
1434 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1435 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
1436 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1437 {
1438     \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
1439     { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
1440     \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
1441     \tl_map_inline:xn {#2}
1442     {
1443         \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1444         \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1445     }
1446 }
1447 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1448 {
1449     \tl_map_inline:xn {#2}
1450     {
1451         \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
1452         {
1453             \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1454             \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1455         }
1456     }
1457 }
1458 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1459 {
1460     \tl_map_inline:xn {#2}
1461     {
1462         \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1463         \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1464     }
1465 }
判断一个标点符号是否为全角右标点和长标点符号。
1466 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
1467 {
1468     \if_int_compare:w \xeCJK_token_value_class:N #1 = \xeCJK_class_num:n { FullRight }
1469     \prg_return_true: \else: \prg_return_false: \fi:
1470 }

```

```

1471 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1472 {
1473   \exp_args:Nc
1474   \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1475   {
1476     \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
1477     \prg_return_true: \else: \prg_return_false: \fi:
1478   }
1479 }

```

一些用于记录的辅助函数。

```

1480 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
1481 { c__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
1482 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn
1483 { \__xeCJK_punct_dim_csname:nnnn { \l_xeCJK_punct_style_tl } }
1484 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnnn #1#2#3#4
1485 { c__\l_xeCJK_current_font_tl/#1/#2/#3/#4/tl }
1486 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
1487 { \tl_use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
1488 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
1489 { \tl_use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
1490 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nxx #1#2#3
1491 { \tl_const:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } {#3} }
1492 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnxx #1#2#3#4
1493 { \tl_const:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } {#4} }

```

定义标点处理模板。

```

1494 \DeclareObjectType { xeCJK / punctuation } { \c_three }
1495 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_three }
1496 {
1497   enabled-global-setting : boolean = true ,
1498   fixed-punct-width     : length  = \c_max_dim ,
1499   fixed-punct-ratio     : real    = \c_one_fp ,
1500   mixed-punct-width     : length  = \KeyValue { fixed-punct-width } ,
1501   mixed-punct-ratio     : real    = \KeyValue { fixed-punct-ratio } ,
1502   middle-punct-width    : length  = \KeyValue { fixed-punct-width } ,
1503   middle-punct-ratio    : real    = \KeyValue { fixed-punct-ratio } ,
1504   fixed-margin-width    : length  = \c_max_dim ,
1505   fixed-margin-ratio    : real    = \c_one_fp ,
1506   mixed-margin-width    : length  = \KeyValue { fixed-margin-width } ,
1507   mixed-margin-ratio    : real    = \KeyValue { fixed-margin-ratio } ,
1508   middle-margin-width   : length  = \KeyValue { fixed-margin-width } ,
1509   middle-margin-ratio   : real    = \KeyValue { fixed-margin-ratio } ,
1510   add-min-bound-to-margin : boolean = false ,
1511   optimize-margin       : boolean = false ,
1512   margin-minimum        : length  = \c_zero_dim ,
1513   enabled-kerning       : boolean = true ,
1514   min-bound-to-kerning  : boolean = false ,
1515   kerning-total-width   : length  = \c_max_dim ,
1516   kerning-total-ratio   : real    = 0.75 ,
1517   optimize-kerning      : boolean = false ,
1518   same-align-margin     : length  = \c_max_dim ,
1519   same-align-ratio      : real    = \c_zero_fp ,
1520   different-align-margin : length  = \c_max_dim ,
1521   different-align-ratio : real    = \c_zero_fp ,
1522   kerning-margin-width  : length  = \c_max_dim ,
1523   kerning-margin-ratio  : real    = \c_one_fp ,
1524   kerning-margin-minimum : length  = \c_zero_dim
1525 }
1526 \DeclareTemplateCode { xeCJK / punctuation } { basic } { \c_three }
1527 {
1528   enabled-global-setting = \l__xeCJK_enabled_global_setting_bool ,
1529   fixed-punct-width     = \l__xeCJK_fixed_punct_width_dim ,

```

```

1530 fixed-punct-ratio      = \l__xeCJK_fixed_punct_ratio_fp ,
1531 mixed-punct-width      = \l__xeCJK_mixed_punct_width_dim ,
1532 mixed-punct-ratio      = \l__xeCJK_mixed_punct_ratio_fp ,
1533 middle-punct-width     = \l__xeCJK_middle_punct_width_dim ,
1534 middle-punct-ratio     = \l__xeCJK_middle_punct_ratio_fp ,
1535 fixed-margin-width     = \l__xeCJK_fixed_margin_width_dim ,
1536 fixed-margin-ratio     = \l__xeCJK_fixed_margin_ratio_fp ,
1537 mixed-margin-width     = \l__xeCJK_mixed_margin_width_dim ,
1538 mixed-margin-ratio     = \l__xeCJK_mixed_margin_ratio_fp ,
1539 middle-margin-width    = \l__xeCJK_middle_margin_width_dim ,
1540 middle-margin-ratio    = \l__xeCJK_middle_margin_ratio_fp ,
1541 add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
1542 optimize-margin        = \l__xeCJK_optimize_margin_bool ,
1543 margin-minimum         = \l__xeCJK_margin_minimum_dim ,
1544 enabled-kerning        = \l__xeCJK_enabled_kerning_bool ,
1545 min-bound-to-kerning   = \l__xeCJK_min_bound_to_kerning_bool ,
1546 kerning-total-width    = \l__xeCJK_kerning_total_width_dim ,
1547 kerning-total-ratio    = \l__xeCJK_kerning_total_ratio_fp ,
1548 optimize-kerning       = \l__xeCJK_optimize_kerning_bool ,
1549 same-align-margin      = \l__xeCJK_same_align_margin_dim ,
1550 same-align-ratio       = \l__xeCJK_same_align_ratio_fp ,
1551 different-align-margin  = \l__xeCJK_different_align_margin_dim ,
1552 different-align-ratio  = \l__xeCJK_different_align_ratio_fp ,
1553 kerning-margin-width   = \l__xeCJK_kerning_margin_width_dim ,
1554 kerning-margin-ratio   = \l__xeCJK_kerning_margin_ratio_fp ,
1555 kerning-margin-minimum = \l__xeCJK_kerning_margin_minimum_dim
1556 }
1557 {
1558   \AssignTemplateKeys
1559   \tl_if_blank:nTF {#3}
1560     { \xeCJK_punct_margin_process:NN {#1} {#2} }
1561     { \xeCJK_punct_kerning_process:NN {#2} {#3} }
1562 }

```

\xeCJK_punct_margin_process:NN

```

1563 \cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
1564 {
1565   \dim_set:Nn \l__xeCJK_bound_dim { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
1566   \dim_set:Nn \l__xeCJK_reverse_bound_dim
1567     {
1568       \__xeCJK_use_punct_dim:nnn { bound }
1569       { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl } {#2}
1570     }
1571   \dim_set:Nn \l__xeCJK_tmp_dim
1572     {
1573       \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1574       {
1575         \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
1576         { \use_none:n }
1577         {
1578           \xeCJK_if_blank_x:nTF \g__xeCJK_punct_width_tl
1579           { \use:n }
1580           { \g__xeCJK_punct_width_tl \use_none:n }
1581         }
1582       }
1583       { \use:n }
1584     }
1585   \__xeCJK_punct_if_middle:NTF {#2}
1586   { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
1587   {
1588     \__xeCJK_punct_if_mixed_width:NTF {#2}
1589     { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
1590     { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }

```

```

1591     }
1592   }
1593 }
1594 \__xeCJK_gset_punct_dim:nnx { glue } {#1} {#2}
1595 {
1596   \dim_max:nn
1597   { \l__xeCJK_margin_minimum_dim }
1598   {
1599     \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1600     {
1601       \__xeCJK_punct_if_middle:NTF {#2}
1602       {
1603         (
1604           \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1605         ) / \c_two
1606       }
1607       {
1608         \bool_if:NTF \l__xeCJK_optimize_margin_bool
1609         {
1610           \dim_max:nn
1611           { \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
1612         }
1613         { \use:n }
1614         {
1615           \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim
1616           - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1617         }
1618       }
1619     }
1620   }
1621   \bool_if:NTF \l__xeCJK_optimize_margin_bool
1622   { \dim_min:nn { \l__xeCJK_bound_dim } }
1623   { \use:n }
1624   {
1625     \__xeCJK_punct_if_middle:NTF {#2}
1626     {
1627       \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
1628       { \l__xeCJK_middle_margin_width_dim }
1629       {
1630         \fp_use:N \l__xeCJK_middle_margin_ratio_fp
1631         \etex_dimexpr:D
1632         ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / \c_two
1633         \scan_stop:
1634       }
1635     }
1636     {
1637       \__xeCJK_punct_if_mixed_width:NTF {#2}
1638       { \__xeCJK_margin_width_or_ratio:n { mixed } }
1639       { \__xeCJK_margin_width_or_ratio:n { fixed } }
1640     }
1641   }
1642 }
1643 }
1644 }
1645 }
1646 \dim_new:N \l__xeCJK_bound_dim
1647 \dim_new:N \l__xeCJK_reverse_bound_dim
1648 (End definition for \xeCJK_punct_margin_process:NN.)
1649 \__xeCJK_punct_width_or_ratio:nN
1650 \cs_new_nopar:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
1651 {

```



```

1650 \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
1651 { \use:c { l__xeCJK_#1_punct_width_dim } }
1652 {
1653 \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } = \c_zero_fp
1654 { \c_max_dim }
1655 {
1656 \fp_use:c { l__xeCJK_#1_punct_ratio_fp }
1657 \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
1658 }
1659 }
1660 }
(End definition for \__xeCJK_punct_width_or_ratio:nN)

```

__xeCJK_margin_width_or_ratio:n

```

1661 \cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
1662 {
1663 \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
1664 { \use:c { l__xeCJK_#1_margin_width_dim } }
1665 {
1666 \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
1667 \etex_dimexpr:D \l__xeCJK_bound_dim \scan_stop:
1668 }
1669 \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
1670 { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
1671 }
(End definition for \__xeCJK_margin_width_or_ratio:n)

```

\xeCJK_punct_kerning_process:NN

```

1672 \cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
1673 {
1674 \__xeCJK_gset_punct_dim:nnx { kern } {#1} {#2}
1675 {
1676 \dim_eval:n
1677 {
1678 \bool_if:nTF
1679 {
1680 \l__xeCJK_enabled_global_setting_bool &&
1681 \tl_if_exist_p:c { g__xeCJK_punct/kern/#1/#2/tl }
1682 }
1683 { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }
1684 {
1685 \bool_if:NTF \l__xeCJK_enabled_kerning_bool
1686 { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
1687 { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
1688 }
1689 - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1690 - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} )
1691 }
1692 }
1693 }
(End definition for \xeCJK_punct_kerning_process:NN)

```

__xeCJK_original_kerning_margin:NN 相邻两个标点符号之间的本来空白宽度。

```

1694 \cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
1695 {
1696 \dim_eval:n
1697 {
1698 \__xeCJK_use_punct_dim:nnn
1699 { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
1700 { \c__xeCJK_right_tl } {#1} +
1701 \__xeCJK_use_punct_dim:nnn
1702 { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }

```

```

1703     { \c__xeCJK_left_tl } {#2}
1704   }
1705 }
(End definition for \__xeCJK_original_kerning_margin:NN.)

```

_xeCJK_calc_kerning_margin:NN

```

1706 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:NN #1#2
1707 {
1708   \dim_max:nn
1709   { \l__xeCJK_kerning_margin_minimum_dim }
1710   {
1711     \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
1712     { \__xeCJK_punct_min_bound:NN {#1} {#2} }
1713     {
1714       \bool_if:NTF \l__xeCJK_optimize_kerning_bool
1715       { \dim_max:nn { \__xeCJK_punct_min_bound:NN {#1} {#2} } }
1716       { \use:n }
1717       {
1718         \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
1719         { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
1720         {
1721           \fp_compare:nNnTF \l__xeCJK_kerning_total_ratio_fp = \c_zero_fp
1722           {
1723             \xeCJK_if_same_class:NNTF {#1} {#2}
1724             { \__xeCJK_kerning_width_or_ratio:nNN { same } }
1725             { \__xeCJK_kerning_width_or_ratio:nNN { different } }
1726           }
1727           {
1728             \__xeCJK_calc_kerning_margin:nNN
1729             {
1730               \fp_use:N \l__xeCJK_kerning_total_ratio_fp
1731               \etex_dimexpr:D
1732               \__xeCJK_use_punct_dim:nn { width } {#1} +
1733               \__xeCJK_use_punct_dim:nn { width } {#2}
1734               \scan_stop:
1735             }
1736           }
1737         }
1738         {#1} {#2}
1739       }
1740     }
1741   }
1742 }
(End definition for \__xeCJK_calc_kerning_margin:NN.)

```

_xeCJK_kerning_width_or_ratio:nNN

```

1743 \cs_new_nopar:Npn \__xeCJK_kerning_width_or_ratio:nNN #1#2#3
1744 {
1745   \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
1746   { \use:c { l__xeCJK_#1_align_margin_dim } \use_none:nn }
1747   {
1748     \fp_compare:nNnTF { \use:c { l__xeCJK_#1_align_ratio_fp } } = \c_zero_fp
1749     { \use:n }
1750     { \fp_use:c { l__xeCJK_#1_align_ratio_fp } \use_ii:nn }
1751   }
1752   {
1753     \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
1754     { \l__xeCJK_kerning_margin_width_dim \use_none:n }
1755     { \fp_use:N \l__xeCJK_kerning_margin_ratio_fp \use:n }
1756   }
1757   { \etex_dimexpr:D \__xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
1758 }

```

(End definition for _xeCJK_kerning_width_or_ratio:nnN)

_xeCJK_punct_min_bound:NN

```

1759 \cs_new_nopar:Npn \_xeCJK_punct_min_bound:NN #1#2
1760 {
1761   \dim_max:nn
1762   {
1763     \dim_min:nn
1764     { \_xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} }
1765     { \_xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
1766   }
1767   {
1768     \dim_min:nn
1769     { \_xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} }
1770     { \_xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
1771   }
1772 }

```

(End definition for _xeCJK_punct_min_bound:NN)

_xeCJK_calc_kerning_margin:nnN #2 和 #3 为相邻的两个标点, #1 为要确定的相邻两个标点总共占的宽度。

```

1773 \cs_new_nopar:Npn \_xeCJK_calc_kerning_margin:nnN #1#2#3
1774 {
1775   \dim_eval:n
1776   {
1777     (#1)
1778     - ( \_xeCJK_use_punct_dim:nnn
1779         { \_xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1780         { \c__xeCJK_left_tl } {#2} )
1781     - ( \_xeCJK_use_punct_dim:nnn
1782         { \_xeCJK_punct_if_right:NTF {#3} { glue } { bound } }
1783         { \c__xeCJK_right_tl } {#3} )
1784     - ( \_xeCJK_use_punct_dim:nn { dimen } {#2} )
1785     - ( \_xeCJK_use_punct_dim:nn { dimen } {#3} )
1786   }
1787 }

```

(End definition for _xeCJK_calc_kerning_margin:nnN)

\xeCJK_get_punct_bounds:NN #1 为 \c__xeCJK_left_tl 或 \c__xeCJK_right_tl, #2 为标点符号。

```

1788 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
1789 {
1790   \tl_if_exist:cF { \_xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }
1791   {
1792     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1793     {
1794       \_xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim }
1795       \_xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
1796       \_xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
1797     }
1798     {
1799       { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
1800       \_xeCJK_punct_if_long:NTF {#2}
1801       { \_xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim } }
1802       {
1803         \UseInstance { xeCJK / punctuation }
1804         { \l_xeCJK_punct_style_tl } {#1} {#2} { }
1805       }
1806     }
1807   }
1808 }

```

(End definition for \xeCJK_get_punct_bounds:NN)

`\xeCJK_calc_punct_dimen:N` 计算标点的左右实际边距和实际尺寸。对于破折号, 计算两标点之间的空白, 保证它中间不被断开。

```

1809 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
1810 {
1811   \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl {#1}
1812   { \xeCJK_glyph_bounds:NN \c_one {#1} }
1813   \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_right_tl {#1}
1814   { \xeCJK_glyph_bounds:NN \c_three {#1} }
1815   \dim_set:Nn \l__xeCJK_tmp_dim
1816   {
1817     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} ) +
1818     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1819   }
1820   \__xeCJK_gset_punct_dim:nnx { width } {#1}
1821   { \dim_use:N \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
1822   \__xeCJK_gset_punct_dim:nnx { dimen } {#1}
1823   { \dim_eval:n { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim } }
1824   \__xeCJK_punct_if_long:NT {#1}
1825   {
1826     \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#1}
1827     {
1828       \bool_if:nTF
1829       {
1830         \str_if_eq_p:nn {#1} { ^^^^2025 } ||
1831         \str_if_eq_p:nn {#1} { ^^^^2026 }
1832       }
1833       { \c_zero_dim }
1834       { - \dim_use:N \l__xeCJK_tmp_dim }
1835     }
1836   }
1837 }
1838 \cs_generate_variant:Nn \xeCJK_calc_punct_dimen:N { f }
(End definition for \xeCJK_calc_punct_dimen:N.)

```

`\xeCJK_glyph_bounds:NN` 用 `\XeTeXglyphbounds` 取得标点符号的上下左右空白。

```

1839 \cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
1840 {
1841   \dim_use:N \XeTeXglyphbounds #1 ~
1842   \XeTeXcharglyph \xeCJK_token_value_charcode:N #2 \exp_stop_f:
1843 }
(End definition for \xeCJK_glyph_bounds:NN.)

```

`\xeCJK_get_punct_kerning:NN` 标点挤压。

```

1844 \cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
1845 {
1846   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
1847   {
1848     \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1849     { \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#2} { \c_zero_dim } }
1850     {
1851       \UseInstance { xeCJK / punctuation }
1852       { \l__xeCJK_punct_style_tl } { } {#1} {#2}
1853     }
1854   }
1855 }
1856 \cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
(End definition for \xeCJK_get_punct_kerning:NN.)

```

PunctStyle

```

1857 \keys_define:nn { xeCJK / options }
1858 {
1859   PunctStyle .choice: ,

```

```

1860 PunctStyle .default:n = { quanjiao } ,
1861 PunctStyle / halfwidth .meta:n = { PunctStyle = banjiao } ,
1862 PunctStyle / fullwidth .meta:n = { PunctStyle = quanjiao } ,
1863 PunctStyle / mixedwidth .meta:n = { PunctStyle = kaiming } ,
1864 PunctStyle / marginkerning .meta:n = { PunctStyle = hangmobanjiao } ,
1865 PunctStyle / plain .code:n =
1866 { \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl } ,
1867 PunctStyle / unknown .code:n =
1868 {
1869 \IfInstanceExistTF { xeCJK / punctuation } { \l_keys_value_tl }
1870 { \tl_set:Nx \l_xeCJK_punct_style_tl { \l_keys_value_tl } }
1871 { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
1872 }
1873 }
1874 \tl_new:N \l_xeCJK_punct_style_tl
1875 \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
1876 \__xeCJK_msg_new:nn { punct-style-unknown }
1877 {
1878 Punctuation~style~"#1"~is~unknown. \\\
1879 The~available~styles~are~listed~as~follow.\\\
1880 "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
1881 }

```

(End definition for *PunctStyle*. This function is documented on page 4.)

\xeCJKDeclarePunctStyle 定义新的标点处理风格, 已经存在的同名风格将被覆盖。

```

1882 \NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
1883 {
1884 \IfInstanceExistTF { xeCJK / punctuation } {#1}
1885 { \__xeCJK_warning:nx { punct-style-already-defined } {#1} }
1886 { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
1887 \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
1888 }
1889 \seq_new:N \g__xeCJK_punct_style_seq
1890 \__xeCJK_msg_new:nn { punct-style-already-defined }
1891 {
1892 Punctuation~style~"#1"~is~already~defined!. \\\
1893 The~existing~style~of~"#1"~will~be~overwritten.\\
1894 }
1895 \@onlypreamble \xeCJKDeclarePunctStyle

```

(End definition for *\xeCJKDeclarePunctStyle*. This function is documented on page 10.)

\xeCJKEditPunctStyle 对已有的标点处理风格进行修改。

```

1896 \NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
1897 {
1898 \IfInstanceExistTF { xeCJK / punctuation } {#1}
1899 { \EditInstance { xeCJK / punctuation } {#1} {#2} }
1900 { \__xeCJK_error:nx { punct-style-unknown } {#1} }
1901 }
1902 \@onlypreamble \xeCJKEditPunctStyle

```

(End definition for *\xeCJKEditPunctStyle*. This function is documented on page 10.)

默认设置即为全角格式。

```

1903 \xeCJKDeclarePunctStyle { quanjiao } { }
1904 \xeCJKDeclarePunctStyle { hangmobanjiao } { enabled-kerning = false }
1905 \xeCJKDeclarePunctStyle { banjiao }
1906 {
1907 fixed-punct-ratio = 0.5 ,
1908 optimize-margin = true ,
1909 kerning-total-ratio = 0.5 ,
1910 optimize-kerning = true
1911 }

```

```

1912 \xeCJKDeclarePunctStyle { kaiming }
1913 {
1914     fixed-punct-ratio    = 0.5 ,
1915     mixed-punct-ratio    = 0.8 ,
1916     optimize-margin      = true ,
1917     kerning-total-ratio  = 0.5 ,
1918     optimize-kerning     = true
1919 }
1920 \xeCJKDeclarePunctStyle { CCT }
1921 {
1922     fixed-punct-ratio    = 0.7 ,
1923     optimize-margin      = true ,
1924     kerning-total-ratio  = 0.6 ,
1925     optimize-kerning     = true
1926 }

```

5.11 后备字体

AutoFallBack 后备字体的宏包选项声明。

```

1927 \keys_define:nn { xeCJK / options }
1928 {
1929     AutoFallBack .choice: ,
1930     AutoFallBack / true .code:n =
1931     {
1932         \cs_if_eq:NNF \CJKsymbol \xeCJK_fallback_test_glyph:N
1933         {
1934             \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1935             \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
1936         }
1937     } ,
1938     AutoFallBack / false .code:n =
1939     {
1940         \cs_if_eq:NNT \CJKsymbol \xeCJK_fallback_test_glyph:N
1941         { \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N }
1942     } ,
1943     AutoFallBack .default:n = { true } ,
1944     fallback .meta:n = { AutoFallBack = true }
1945 }

```

(End definition for AutoFallBack. This function is documented on page 4.)

\xeCJK_fallback_test_glyph:N 测试当前字体中是否存在当前字符,如存在则直接输出,否则启用后备字体。

```

1946 \cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
1947 {
1948     \xeCJK_glyph_if_exist:NTF {#1}
1949     { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1950     {
1951         \xeCJK_class_group_begin:
1952         \tl_set_eq:NN \l__xeCJK_fallback_family_tl \l_xeCJK_family_tl
1953         \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_fallback_family_tl/FallBack }
1954         \xeCJK_class_group_end:
1955     }
1956 }

```

(End definition for \xeCJK_fallback_test_glyph:N)

\xeCJK_fallback_loop:Nn 循环测试后备字体是否包含字符 #1。若后备字体中存在该字符或者再没有后备字体,则结束循环。当前字体族没有备用字体时,使用 \CJKfamilydefault 的设置。

```

1957 \cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
1958 {
1959     \xeCJK_family_if_exist:xF {#2}
1960     {
1961         \xeCJK_select_font:x {#2}

```

```

1962 \tl_set:Nx \l__xeCJK_fallback_family_tl {#2}
1963 \xeCJK_glyph_if_exist:NTF {#1}
1964 { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1965 { \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_fallback_family_tl/FallBack } }
1966 }
1967 {
1968 \str_if_eq_x:nnTF { \CJKfamilydefault } { \l__xeCJK_family_tl }
1969 {
1970 \use:x
1971 {
1972 \msg_warning:nnnnn { xeCJK } { missing-glyph }
1973 { \l__xeCJK_fallback_family_tl } {#1} { \int_to_hexadecimal:n { `#1 } }
1974 }
1975 \__xeCJK_fallback_save_CJKsymbol:N {#1}
1976 }
1977 {
1978 \tl_set:Nx \l__xeCJK_family_tl { \CJKfamilydefault }
1979 \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_family_tl }
1980 }
1981 }
1982 }
1983 \__xeCJK_msg_new:nn { missing-glyph }
1984 {
1985 CJKfamily~\__xeCJK_msg_family_map:n {#1}'~
1986 ( \prop_get:Nn \g__xeCJK_family_font_name_prop {#1} )~
1987 does~not~contain~glyph~`#2'~(U+#3).\
1988 }
(End definition for \xeCJK_fallback_loop:Nn.)

```

\setCJKfallbackfamilyfont

```

1989 \NewDocumentCommand \setCJKfallbackfamilyfont { m O { } m }
1990 { \use:x { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} } }
(End definition for \setCJKfallbackfamilyfont. This function is documented on page 7.)

```

\xeCJK_set_family_fallback:nnn

```

1991 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnn #1#2#3
1992 {
1993 \group_begin:
1994 \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
1995 \prop_get:NVNF \g__xeCJK_family_font_name_prop
1996 \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
1997 { \tl_clear:N \l__xeCJK_font_name_tl }
1998 \clist_map_inline:nn {#3}
1999 {
2000 \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
2001 \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
2002 \clist_put_left:Nn \l__xeCJK_sub_font_options_clist {#2}
2003 \xeCJK_set_family:VVV \l__xeCJK_fallback_family_tl
2004 \l__xeCJK_sub_font_options_clist \l__xeCJK_sub_font_name_tl
2005 }
2006 \group_end:
2007 }
2008 \tl_new:N \l__xeCJK_fallback_family_tl
(End definition for \xeCJK_set_family_fallback:nnn.)

```

5.12 CJK 字体族声明方式

```

2009 \bool_new:N \g__xeCJK_auto_fake_bold_bool
2010 \bool_new:N \g__xeCJK_auto_fake_slant_bool
2011 \fp_new:N \g__xeCJK_embolden_factor_fp
2012 \fp_new:N \g__xeCJK_slant_factor_fp

```

AutoFakeBold 伪粗体和伪斜体的宏包选项声明。

```
AutoFakeSlant 2013 \keys_define:nn { xeCJK / options }
EmboldenFactor 2014 {
SlantFactor 2015   AutoFakeBold .choices:nn = { true , false }
2016   { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,
2017   AutoFakeBold / unknown .code:n =
2018   {
2019     \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
2020     \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
2021   } ,
2022   AutoFakeBold .default:n = { true } ,
2023   AutoFakeSlant .choices:nn = { true , false }
2024   { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
2025   AutoFakeSlant / unknown .code:n =
2026   {
2027     \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
2028     \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
2029   } ,
2030   AutoFakeSlant .default:n = { true } ,
2031   EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
2032   SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
2033   BoldFont .meta:n = { AutoFakeBold = true } ,
2034   boldfont .meta:n = { AutoFakeBold = true } ,
2035   SlantFont .meta:n = { AutoFakeSlant = true } ,
2036   slantfont .meta:n = { AutoFakeSlant = true }
2037 }
```

(End definition for *AutoFakeBold* and others. These functions are documented on page 4.)

\xeCJK_new_sub_key:n 用于定义 CJK 子区字体和备用字体的选项。

```
\g__xeCJK_sub_key_seq 2038 \seq_new:N \g__xeCJK_sub_key_seq
2039 \cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
2040 {
2041   \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
2042   \keys_define:nn { xeCJK / features }
2043   {
2044     #1 .code:n =
2045     {
2046       \tl_if_blank:nTF {##1}
2047       {
2048         \prop_clear:N \l__xeCJK_sub_key_prop
2049         \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
2050         \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
2051       }
2052       {
2053         \str_if_eq:nnTF {##1} { * }
2054         { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
2055         { \__xeCJK_get_sub_features:nn {#1} {##1} }
2056       }
2057     } ,
2058     #1 .default:n = { }
2059   }
2060 }
```

(End definition for *\xeCJK_new_sub_key:n* and *\g__xeCJK_sub_key_seq*.)

__xeCJK_get_sub_features:nn

_xeCJK_get_sub_features:w

```
2061 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
2062 {
2063   \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
2064   \clist_clear:N \l__xeCJK_sub_font_options_clist
2065   \exp_after:wN \_xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
2066   \q_mark [ \q_nil ] \q_mark \q_stop
2067   \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
```



```

2068 { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
2069 { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
2070 \prop_put:Nnx \l__xeCJK_sub_key_prop {#1}
2071 {
2072   { \exp_not:V \l__xeCJK_sub_font_options_clist }
2073   { \exp_not:V \l__xeCJK_sub_font_name_tl }
2074 }
2075 }
2076 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
2077 {
2078   \quark_if_nil:nTF {#2}
2079   { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
2080   {
2081     \tl_set:Nx \l__xeCJK_sub_font_name_tl
2082     { \xeCJK_tl_remove_outer_braces:n {#3} }
2083     \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
2084     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
2085     { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
2086   }
2087 }
2088 \tl_new:N \l__xeCJK_sub_family_name_tl
2089 \tl_new:N \l__xeCJK_sub_font_name_tl
2090 \clist_new:N \l__xeCJK_sub_font_options_clist
2091 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
2092 \cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
(End definition for \__xeCJK_get_sub_features:nn and \__xeCJK_get_sub_features:w)

```

FallBack

```

2093 \xeCJK_new_sub_key:n { FallBack }
(End definition for FallBack. This function is documented on page 7.)

```

\g__xeCJK_features_id_prop 调用字体的属性声明,同 fontspec 宏包。

```

      BoldFont      2094 \prop_new:N \g__xeCJK_features_id_prop
      ItalicFont    2095 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
      BoldItalicFont 2096 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
      SlantedFont   2097 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
      BoldSlantedFont 2098 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
      BoldFeatures  2099 \prop_put:Nnn \g__xeCJK_features_id_prop { bfs1 } { BoldSlanted }
      ItalicFeatures 2100 \prop_map_inline:Nn \g__xeCJK_features_id_prop
      BoldItalicFeatures 2101 {
      SlantedFeatures 2102   \keys_define:nn { xeCJK / features }
      BoldSlantedFeatures 2103   {
2104     #2Font .tl_set:c = { l__xeCJK_font_name_#1_tl } ,
2105     #2Features .clist_set:c = { l__xeCJK_font_feat_#1_clist }
2106   }
2107 }
(End definition for \g__xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

```

2108 \keys_define:nn { xeCJK / features }
2109 {
2110   AutoFakeBold .choice: ,
2111   AutoFakeBold / false .code:n =
2112   { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
2113   AutoFakeBold / unknown .code:n =
2114   {
2115     \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
2116     \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
2117   } ,
2118   AutoFakeBold .default:n = { \g__xeCJK_embolden_factor_fp } ,
2119   AutoFakeSlant .choice: ,
2120   AutoFakeSlant / false .code:n =

```

```

2121     { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
2122     AutoFakeSlant / unknown .code:n =
2123     {
2124         \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
2125         \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
2126     } ,
2127     AutoFakeSlant .default:n = { \g__xeCJK_slant_factor_fp }
2128 }

```

(End definition for AutoFakeBold and AutoFakeSlant.)

__xeCJK_set_family_initial:

```

2129 \cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
2130 {
2131     \int_gincr:N \g__xeCJK_family_int
2132     \prop_clear:N \l__xeCJK_add_font_prop
2133     \prop_clear:N \l__xeCJK_sub_key_prop
2134     \prop_map_inline:Nn \g__xeCJK_features_id_prop
2135     {
2136         \tl_clear:c { l__xeCJK_font_name_##1_tl }
2137         \clist_clear:c { l__xeCJK_font_feat_##1_clist }
2138     }
2139     \clist_clear:N \l__xeCJK_fontspec_options_clist
2140     \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
2141     \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
2142     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
2143     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
2144 }
2145 \int_new:N \g__xeCJK_family_int
2146 \prop_new:N \l__xeCJK_add_font_prop
2147 \prop_new:N \l__xeCJK_sub_key_prop
2148 \clist_new:N \l__xeCJK_fontspec_options_clist
2149 \bool_new:N \l__xeCJK_auto_fake_bold_bool
2150 \bool_new:N \l__xeCJK_auto_fake_slant_bool
2151 \fp_new:N \l__xeCJK_embolden_factor_fp
2152 \fp_new:N \l__xeCJK_slant_factor_fp

```

(End definition for __xeCJK_set_family_initial:.)

\xeCJK_set_family:nnn 设置一个 CJK 新字体族,与 \newfontfamily 类似,增加 FallBack 选项。

```

2153 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
2154 {
2155     \group_begin:
2156     \__xeCJK_set_family_initial:
2157     \tl_set:Nn \l__xeCJK_family_name_tl {#1}
2158     \clist_set:Nn \l__xeCJK_font_options_clist {#2}
2159     \tl_set:Nn \l__xeCJK_font_name_tl {#3}
2160     \clist_concat:NNN \l__xeCJK_font_options_clist
2161         \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
2162     \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_options_clist
2163     \keys_set_known:nVN { xeCJK / features }
2164         \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
2165     \__xeCJK_parse_features:
2166     \__xeCJK_pass_features:
2167     \__xeCJK_check_family:V \l__xeCJK_family_name_tl
2168     \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
2169     \__xeCJK_save_family_info:
2170     \__xeCJK_set_sub_block_family:
2171     \group_end:
2172 }
2173 \tl_new:N \l__xeCJK_family_name_tl
2174 \tl_new:N \l__xeCJK_font_name_tl
2175 \clist_new:N \l__xeCJK_font_options_clist
2176 \cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , VVV , Voo }

```

```

2177 \cs_new_protected_nopar:Npn \xeCJK_set_family:xxx #1#2#3
2178 { \use:x { \xeCJK_set_family:nnn {#1} {#2} {#3} } }
(End definition for \xeCJK_set_family:nnn.)

```

_xeCJK_remove_duplicate_keys:N

```

2179 \cs_new_protected_nopar:Npn \_xeCJK_remove_duplicate_keys:N #1
2180 {
2181   \prop_clear:N \l__xeCJK_font_options_prop
2182   \keyval_parse:NNV \_xeCJK_prop_put_aux:n \_xeCJK_prop_put_aux:nn #1
2183   \clist_clear:N #1
2184   \prop_map_inline:Nn \l__xeCJK_font_options_prop
2185   {
2186     \tl_set:No \l__xeCJK_tmp_tl { \use_ii:nn ##2 }
2187     \tl_if_blank:VTF \l__xeCJK_tmp_tl
2188     { \clist_put_right:No #1 { \use_i:nn ##2 } }
2189     {
2190       \clist_put_right:Nx #1
2191       { \exp_not:o { \use_i:nn ##2 } = { \exp_not:V \l__xeCJK_tmp_tl } }
2192     }
2193   }
2194 }
2195 \prop_new:N \l__xeCJK_font_options_prop
2196 \cs_generate_variant:Nn \keyval_parse:NNn { NNV }
2197 \cs_new_protected_nopar:Npn \_xeCJK_prop_put_aux:n #1
2198 { \prop_put:Nnn \l__xeCJK_font_options_prop {#1} { {#1} { } } }
2199 \cs_new_protected_nopar:Npn \_xeCJK_prop_put_aux:nn #1#2
2200 { \prop_put:Nnn \l__xeCJK_font_options_prop {#1} { {#1} {#2} } }
(End definition for \_xeCJK_remove_duplicate_keys:N.)

```

_xeCJK_gset_family_cs:x

```

2201 \cs_new_protected_nopar:Npn \_xeCJK_gset_family_cs:x #1
2202 {
2203   \cs_gset_protected_nopar:cpx { \_xeCJK_family_csname:n {#1} }
2204   {
2205     \group_begin:
2206     \exp_not:n { \cs_set_eq:NN \_xeCJK_update_family:n \use_none:n }
2207     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
2208     { \exp_not:V \l__xeCJK_fontspec_options_clist }
2209     { \exp_not:V \l__xeCJK_font_name_tl }
2210     \_xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
2211     \group_end:
2212   }
2213 }
2214 \tl_new:N \l__xeCJK_fontspec_family_tl
(End definition for \_xeCJK_gset_family_cs:x.)

```

_xeCJK_check_family:n

```

2215 \cs_new_protected_nopar:Npn \_xeCJK_check_family:n #1
2216 {
2217   \prop_get:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
2218   {
2219     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
2220     { \cs_undefine:c { \_xeCJK_family_nfss_csname:n {#1} } }
2221     \_xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
2222   }
2223 }
2224 \cs_generate_variant:Nn \_xeCJK_check_family:n { V }
2225 \_xeCJK_msg_new:nn { CJKfamily-redef }
2226 { Redefining-CJKfamily~\_xeCJK_msg_family_map:n {#1}'~(#2). }
(End definition for \_xeCJK_check_family:n.)

```

```

    \_xeCJK_add_font:nn
\_xeCJK_add_font_if_new:nn 2227 \cs_new_protected_nopar:Npn \_xeCJK_add_font:nn #1#2
2228 { \prop_put:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2229 \cs_new_protected_nopar:Npn \_xeCJK_add_font_if_new:nn #1#2
2230 { \prop_put_if_new:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2231 \cs_generate_variant:Nn \_xeCJK_add_font:nn { nx , nV , nv }
2232 \cs_generate_variant:Nn \_xeCJK_add_font_if_new:nn { nx , nV , nv }
(End definition for \_xeCJK_add_font:nn and \_xeCJK_add_font_if_new:nn.)

\_xeCJK_add_fake_bold:n
\_xeCJK_add_fake_slant:n 2233 \cs_new_protected_nopar:Npn \_xeCJK_add_fake_bold:n #1
2234 {
2235   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2236   { FakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
2237 }
2238 \cs_new_protected_nopar:Npn \_xeCJK_add_fake_slant:n #1
2239 {
2240   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2241   { FakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
2242 }
(End definition for \_xeCJK_add_fake_bold:n and \_xeCJK_add_fake_slant:n.)

\_xeCJK_parse_features:
\_xeCJK_parse_features:nn 2243 \cs_new_protected_nopar:Npn \_xeCJK_parse_features:
2244 { \prop_map_function:NN \g__xeCJK_features_id_prop \_xeCJK_parse_features:nn }
2245 \cs_new_protected_nopar:Npn \_xeCJK_parse_features:nn #1#2
2246 {
2247   \_xeCJK_if_font_select:nTF {#1}
2248   {
2249     \_xeCJK_add_font:nv {#2} { l__xeCJK_font_name_#1_tl }
2250     \_xeCJK_if_it_or_sl:nTF {#1}
2251     {
2252       \_xeCJK_if_font_select:nF { bf #1 }
2253       { \_xeCJK_add_font_if_new:nv {#2} { l__xeCJK_font_name_#1_tl } }
2254     }
2255     {
2256       \str_if_eq:nnT {#1} { bf }
2257       {
2258         \_xeCJK_if_font_select:nF { bfit }
2259         { \_xeCJK_add_font:nV { BoldItalic } \l__xeCJK_font_name_bf_tl }
2260         \_xeCJK_if_font_select:nF { bfsi }
2261         { \_xeCJK_add_font:nV { BoldSlanted } \l__xeCJK_font_name_bf_tl }
2262       }
2263     }
2264   }
2265   { \_xeCJK_set_fake_font:nn {#1} {#2} }
2266 }
(End definition for \_xeCJK_parse_features: and \_xeCJK_parse_features:nn.)

\_xeCJK_if_font_select_p:n
\_xeCJK_if_it_or_sl_p:n 2267 \prg_new_conditional:Npnn \_xeCJK_if_font_select:n #1 { p , T , F , TF }
\_xeCJK_if_font_select:nTF 2268 {
\_xeCJK_if_it_or_sl:nTF 2269   \tl_if_blank:vTF { l__xeCJK_font_name_#1_tl }
2270   { \prg_return_false: } { \prg_return_true: }
2271 }
2272 \prg_new_conditional:Npnn \_xeCJK_if_it_or_sl:n #1 { p , T , F , TF }
2273 {
2274   \if_predicate:w
2275     \bool_if_p:n { \str_if_eq_p:nn { it } {#1} || \str_if_eq_p:nn { sl } {#1} }
2276     \prg_return_true: \else: \prg_return_false: \fi:
2277   }
2278 \cs_generate_variant:Nn \tl_if_blank:nTF { v }

```

(End definition for _xeCJK_if_font_select:n and _xeCJK_if_it_or_sl:n)

_xeCJK_set_fake_font:nn

```

2279 \cs_new_protected_nopar:Npn \_xeCJK_set_fake_font:nn #1#2
2280 {
2281   \str_if_eq:nnTF {#1} { bf }
2282   { \bool_if:NT \l_xeCJK_auto_fake_bold_bool { \_xeCJK_add_fake_bold:n {#1} } }
2283   {
2284     \bool_if:NTF \l_xeCJK_auto_fake_slant_bool
2285     {
2286       \bool_if:nT
2287       { \_xeCJK_if_it_or_sl_p:n {#1}
2288         || ( \str_if_eq_p:nn {#1} { bfit }
2289           && ! ( \_xeCJK_if_font_select_p:n { it } ) )
2290         || ( \str_if_eq_p:nn {#1} { bfsl }
2291           && ! ( \_xeCJK_if_font_select_p:n { sl } ) )
2292       }
2293       { \_xeCJK_add_fake_slant:n {#1} }
2294     }
2295     { \_xeCJK_if_it_or_sl:nT {#1} { \_xeCJK_map_it_sl:nn {#1} {#2} } }
2296     \bool_if:nT
2297     { \l_xeCJK_auto_fake_bold_bool
2298       && ! ( \_xeCJK_if_it_or_sl_p:n {#1} )
2299       && ! ( \_xeCJK_if_font_select_p:n { bf } )
2300     }
2301     { \_xeCJK_add_fake_bold:n {#1} }
2302   }
2303   \_xeCJK_add_font_if_new:nn {#2} { * }
2304 }

```

(End definition for _xeCJK_set_fake_font:nn.)

_xeCJK_map_it_sl:n

```

2305 \cs_new_protected_nopar:Npn \_xeCJK_map_it_sl:nn #1#2
2306 {
2307   \_xeCJK_if_map_font_select:nT {#1}
2308   {
2309     \_xeCJK_add_font:nx {#2} { \_xeCJK_get_map_font:n {#1} }
2310     \_xeCJK_if_font_select:nF { bf #1 }
2311     { \_xeCJK_add_font_if_new:nx { Bold #2 } { \_xeCJK_get_map_font:n {#1} } }
2312   }
2313 }

```

(End definition for _xeCJK_map_it_sl:n.)

_xeCJK_get_map_font:n

```

2314 \cs_new_nopar:Npn \_xeCJK_get_map_font:n #1
2315 { \exp_not:v { l_xeCJK_font_name_ \xeCJK_reverse:nnn {#1} { it } { sl } _tl } }

```

(End definition for _xeCJK_get_map_font:n.)

_xeCJK_if_map_font_select_p:n

_xeCJK_if_map_font_select:nTF

```

2316 \prg_new_conditional:Npnn \_xeCJK_if_map_font_select:n #1 { p , T , F , TF }
2317 {
2318   \xeCJK_if_blank_x:nTF { \_xeCJK_get_map_font:n {#1} }
2319   { \prg_return_false: } { \prg_return_true: }
2320 }

```

(End definition for _xeCJK_if_map_font_select:n.)

_xeCJK_pass_features:

```

2321 \cs_new_protected_nopar:Npn \_xeCJK_pass_features:
2322 {
2323   \prop_map_inline:Nn \g_xeCJK_features_id_prop
2324   {
2325     \clist_if_empty:cF { l_xeCJK_font_feat_#1_clist }

```

```

2326         {
2327             \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2328             { ##2Features = { \exp_not:v { l__xeCJK_font_feat_##1_clist } } }
2329         }
2330     }
2331     \prop_map_inline:Nn \l__xeCJK_add_font_prop
2332     { \clist_put_right:Nn \l__xeCJK_fontspec_options_clist { ##1 = {##2} } }
2333 }
(End definition for \__xeCJK_pass_features:.)

```

\g__xeCJK_family_name_prop

\g__xeCJK_family_font_name_prop

\g__xeCJK_family_font_options_prop

```

2334 \prop_new:N \g__xeCJK_family_name_prop
2335 \prop_new:N \g__xeCJK_family_font_name_prop
2336 \prop_new:N \g__xeCJK_family_font_options_prop
(End definition for \g__xeCJK_family_name_prop, \g__xeCJK_family_font_name_prop, and \g__xeCJK_family_font_options_prop)

```

__xeCJK_save_family_info:

```

2337 \cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
2338 {
2339     \group_begin:
2340     \prop_gput:NVV \g__xeCJK_family_font_name_prop
2341     \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
2342     \prop_gput:NVV \g__xeCJK_family_font_options_prop
2343     \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
2344     \group_end:
2345 }
(End definition for \__xeCJK_save_family_info:.)

```

__xeCJK_set_sub_block_family:

```

2346 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_block_family:
2347 {
2348     \prop_map_inline:Nn \l__xeCJK_sub_key_prop
2349     {
2350         \tl_set:Nx \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/##1 }
2351         \quark_if_no_value:nTF {##2}
2352         { \__xeCJK_copy_sub_family:n {##1} }
2353         {
2354             \xeCJK_set_family:Voo \l__xeCJK_sub_family_name_tl
2355             { \use_i:nn ##2 } { \use_ii:nn ##2 }
2356         }
2357     }
2358 }
2359 \cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
2360 {
2361     \__xeCJK_check_family:V \l__xeCJK_sub_family_name_tl
2362     \prop_get:NVNT \g__xeCJK_family_font_name_prop
2363     \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
2364     {
2365         \prop_gput:NVV \g__xeCJK_family_font_name_prop
2366         \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
2367     }
2368     \prop_get:NVNT \g__xeCJK_family_font_options_prop
2369     \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
2370     {
2371         \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
2372         \prop_gput:NVV \g__xeCJK_family_font_options_prop
2373         \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist
2374     }
2375     \cs_gset_protected_nopar:cpx
2376     { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
2377     {
2378         \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }

```

```

2379     {
2380     \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
2381     { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_fontspec_family_tl
2382     {
2383     \__xeCJK_gset_family_nfss_cs:xx
2384     { \l__xeCJK_sub_family_name_tl }
2385     { \exp_not:N \l__xeCJK_fontspec_family_tl }
2386     }
2387     }
2388   }
2389 }
(End definition for \__xeCJK_set_sub_block_family:.)

```

`__xeCJK_copy_family:nn`

```

2390 \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
2391 {
2392   \xeCJK_family_if_exist:xT {#2}
2393   {
2394     \tl_map_inline:nn
2395     {
2396       \g__xeCJK_family_name_prop
2397       \g__xeCJK_family_font_name_prop
2398       \g__xeCJK_family_font_options_prop
2399     }
2400     {
2401       \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
2402       { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmp_tl }
2403     }
2404     \cs_gset_eq:cc
2405     { \__xeCJK_family_nfss_csname:n {#1} }
2406     { \__xeCJK_family_nfss_csname:n {#2} }
2407   }
2408 }
2409 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }
(End definition for \__xeCJK_copy_family:nn.)

```

5.13 字体切换

`\l_xeCJK_current_font_tl`

缓存当前字体的原始格式,以加速编译。

`\xeCJK_select_font:`
`\xeCJK_select_font:x`

```

2410 \tl_new:N \l_xeCJK_current_font_tl
2411 \tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
2412 \cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
2413 \cs_new_protected_nopar:Npn \xeCJK_select_font:
2414 {
2415   \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
2416   {
2417     \tl_set:Nx \l_xeCJK_current_font_tl { \l_xeCJK_current_font_tl }
2418     \__xeCJK_family_use:x { \l_xeCJK_family_tl }
2419     \xeCJK_font_gset_to_current:c { \l_xeCJK_current_font_tl }
2420   }
2421 }
2422 \cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
2423 {
2424   \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
2425   {
2426     \tl_set:Nx \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n {#1} }
2427     \__xeCJK_family_use:x {#1}
2428     \xeCJK_font_gset_to_current:c { \l_xeCJK_current_font_tl }
2429   }
2430 }
2431 \tl_new:N \l_xeCJK_current_font_tl
2432 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:

```

(End definition for \l_xeCJK_current_font_tl, \xeCJK_select_font:, and \xeCJK_select_font:x.)

_xeCJK_switch_font:nn 两个 CJK 分区之间的字体切换。

```
2433 \cs_new_protected_nopar:Npn \_xeCJK_switch_font:nn #1#2
2434 {
2435   \str_if_eq:nnF {#1} {#2}
2436   {
2437     \_xeCJK_info:nxx { CJK-block } {#1} {#2}
2438     \str_if_eq:nnTF {#2} { CJK }
2439     { \xeCJK_select_font: }
2440     { \_xeCJK_block_select_font:n {#2} }
2441   }
2442 }
2443 \_xeCJK_msg_new:nn { CJK-block } { Switch-from-block~`#1'~to~`#2'. }
```

(End definition for _xeCJK_switch_font:nn.)

_xeCJK_block_select_font:n 若当前 CJK 字体族没有定义子分区 #1 的字体,则使用 \CJKfamilydefault 的对应分区字体;若 \CJKfamilydefault 也没有定义该分区字体,则使用当前 CJK 字体族的主分区字体。

```
2444 \cs_new_protected_nopar:Npn \_xeCJK_block_select_font:n #1
2445 {
2446   \cs_if_exist_use:cF { \_xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
2447   {
2448     \tl_set:Nx \l_xeCJK_current_coord_tl
2449     { \_xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
2450     \xeCJK_family_if_exist:xF { \l_xeCJK_family_tl/#1 }
2451     {
2452       \_xeCJK_copy_family:xx { \l_xeCJK_family_tl/#1 }
2453       {
2454         \cs_if_exist:cTF
2455         { \_xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
2456         { \CJKfamilydefault/#1 } { \l_xeCJK_family_tl }
2457       }
2458     }
2459     \_xeCJK_family_use:x { \l_xeCJK_family_tl/#1 }
2460     \xeCJK_font_gset_to_current:c { \l_xeCJK_current_coord_tl }
2461   }
2462 }
2463 (End definition for \_xeCJK_block_select_font:n.)
```

_xeCJK_family_csname:n

```
\_xeCJK_family_nfss_csname:n 2463 \cs_new_nopar:Npn \_xeCJK_family_csname:n #1 { xeCJK/family/#1 }
\_xeCJK_family_use:x 2464 \cs_new_nopar:Npn \_xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
\_xeCJK_gset_family_nfss_cs:xx 2465 \cs_new_nopar:Npn \_xeCJK_family_use:x #1 { \use:c { \_xeCJK_family_nfss_csname:n {#1} } }
2466 \cs_new_protected_nopar:Npn \_xeCJK_gset_family_nfss_cs:xx #1#2
2467 {
2468   \prop_gput:Nxx \g_xeCJK_family_name_prop {#1} {#2}
2469   \cs_gset_protected_nopar:cpx { \_xeCJK_family_nfss_csname:n {#1} }
2470   {
2471     \tl_set:Nx \exp_not:N \f@encoding { \_xeCJK_encoding_tl }
2472     \tl_set:Nx \exp_not:N \f@family {#2}
2473     \exp_not:N \selectfont
2474   }
2475 }
2476 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
2477 (End definition for \_xeCJK_family_csname:n and others.)
```

\xeCJK_family_if_exist:xTF

```
2477 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
2478 {
2479   \cs_if_exist:cTF { \_xeCJK_family_nfss_csname:n {#1} }
2480   { \use_i:nn }
2481   { \cs_if_exist_use:cTF { \_xeCJK_family_csname:n {#1} } }
```



```

2482     { \prg_return_true: } { \prg_return_false: }
2483   }
(End definition for \xeCJK_family_if_exist:xF.)

```

\CJKfamily 用于切换 CJK 字体族。

```

2484 \NewDocumentCommand \CJKfamily { t+ t- m }
2485 {
2486   \xeCJK_if_blank_x:nTF {#3}
2487   {
2488     \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
2489     \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
2490   }
2491   {
2492     \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
2493     {
2494       \xeCJK_family_if_exist:xF {#3}
2495       {
2496         \tl_set:Nx \l_xeCJK_family_tl {#3}
2497         \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2498         \IfBooleanT {#1} { \__xeCJK_family_use:x {#3} }
2499       }
2500       { \__xeCJK_family_unknown_warning:x {#3} }
2501     }
2502   }
2503   \tex_ignorespaces:D
2504 }
2505 \cs_new_protected_nopar:Npn \xeCJK_switch_family:n #1
2506 {
2507   \xeCJK_family_if_exist:xF {#1}
2508   {
2509     \tl_set:Nx \l_xeCJK_family_tl {#1}
2510     \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2511   }
2512   { \__xeCJK_family_unknown_warning:x {#1} }
2513 }

```

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

\l_xeCJK_family_tl 用于保存文档当前正在使用的 CJK 字体族。

```

2514 \tl_new:N \l_xeCJK_family_tl
(End definition for \l_xeCJK_family_tl.)

```

__xeCJK_gobble_CJKfamily:

```

2515 \cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
2516 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
2517 \DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }
(End definition for \__xeCJK_gobble_CJKfamily:.)

```

\xeCJK_family_if_exist_use:x

```

2518 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
2519 {
2520   \xeCJK_family_if_exist:xF {#1}
2521   { \__xeCJK_family_use:x {#1} }
2522   { \__xeCJK_family_unknown_warning:x {#1} }
2523 }
(End definition for \xeCJK_family_if_exist_use:x)

```

_xeCJK_family_unknown_warning:n

```

2524 \cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:n #1
2525 {
2526   \prop_if_empty:NF \g__xeCJK_family_font_name_prop
2527   {
2528     \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}

```

```

2529     {
2530         \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
2531         \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
2532     }
2533 }
2534 }
2535 \cs_generate_variant:Nn \__xeCJK_family_unknown_warning:n { x }
2536 \seq_new:N \g__xeCJK_unknown_family_seq
2537 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
2538 {
2539     Unknown~CJK~family~\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
2540     Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2541 }
2542 \cs_new_nopar:Npn \__xeCJK_msg_def_family_map:n #1
2543 {
2544     \str_case:x:nnn {#1}
2545     {
2546         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
2547         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
2548         \CJKttdefault { \token_to_str:N \setCJKmonofont }
2549     }
2550     { \token_to_str:N \setCJKfamilyfont {#1} }
2551     [...]{}
2552 }
2553 \cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1
2554 {
2555     \str_case:x:nnn {#1}
2556     {
2557         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
2558         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
2559         \CJKttdefault { \token_to_str:N \CJKttdefault }
2560     }
2561     {#1}
2562 }

```

(End definition for `__xeCJK_family_unknown_warning:n`.)

`\setCJKmainfont` 设置文档的 CJK 普通字体、无衬线和等宽字体。

```

\setCJKmainfont 2563 \NewDocumentCommand \setCJKmainfont { 0 { } m }
\setCJKsansfont 2564 {
\setCJKmonofont 2565     \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2}
2566     \normalfont
2567 }
2568 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
2569 \NewDocumentCommand \setCJKsansfont { 0 { } m }
2570 {
2571     \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2}
2572     \normalfont
2573 }
2574 \NewDocumentCommand \setCJKmonofont { 0 { } m }
2575 {
2576     \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2}
2577     \normalfont
2578 }

```

(End definition for `\setCJKmainfont`, `\setCJKsansfont`, and `\setCJKmonofont`. These functions are documented on page 5.)

```

2579 \@onlypreamble \setCJKmainfont
2580 \@onlypreamble \setCJKmathfont
2581 \@onlypreamble \setCJKsansfont
2582 \@onlypreamble \setCJKmonofont
2583 \@onlypreamble \setCJKromanfont

```

`\setCJKfamilyfont` 分别用于预声明 CJK 字体和随机调用 CJK 字体。

```

\newCJKfontfamily 2584 \NewDocumentCommand \setCJKfamilyfont { m 0 { } m }
\CJKfontspec

```

```

2585 { \xeCJK_set_family:xxx {#1} {#2} {#3} }
2586 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
2587 {
2588   \tl_set:Nx \l__xeCJK_tmp_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
2589   \cs_new_protected_nopar:Npx #2 { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
2590   \xeCJK_set_family:xxx { \l__xeCJK_tmp_tl } {#3} {#4}
2591 }
2592 \NewDocumentCommand \CJKfontspec { O { } m }
2593 {
2594   \use:x { \xeCJK_fontspec:nn {#1} {#2} }
2595   \tex_ignorespaces:D
2596 }

```

(End definition for `\setCJKfamilyfont`, `\newCJKfontfamily`, and `\CJKfontspec`. These functions are documented on page 6.)

`\xeCJK_fontspec:nn`

```

2597 \cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
2598 {
2599   \prop_get:NnNTF \g__xeCJK_fontspec_prop
2600   { CJKfontspec/#1/#2/id } \l_xeCJK_family_tl
2601   { \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl }
2602   {
2603     \__xeCJK_fontspec:xnn
2604     { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
2605     {#1} {#2}
2606   }
2607 }
2608 \cs_new_protected_nopar:Npn \__xeCJK_fontspec:nnn #1#2#3
2609 {
2610   \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
2611   \xeCJK_set_family:nnn {#1} {#2} {#3}
2612   \xeCJK_switch_family:n {#1}
2613 }
2614 \cs_generate_variant:Nn \xeCJK_fontspec:nn { VV }
2615 \cs_generate_variant:Nn \__xeCJK_fontspec:nnn { x }
2616 \prop_new:N \g__xeCJK_fontspec_prop

```

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

`\defaultCJKfontfeatures`
`\addCJKfontfeatures`

分别用于设置 CJK 字体的默认属性和增加当前 CJK 字体的属性。

```

2617 \clist_new:N \g__xeCJK_default_features_clist
2618 \NewDocumentCommand \defaultCJKfontfeatures { m }
2619 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
2620 \@onlypreamble \defaultCJKfontfeatures
2621 \NewDocumentCommand \addCJKfontfeatures { s O { } m }
2622 {
2623   \xeCJK_add_font_features:Nxx #1 {#2} {#3}
2624   \tex_ignorespaces:D
2625 }
2626 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

(End definition for `\defaultCJKfontfeatures` and `\addCJKfontfeatures`. These functions are documented on page 6.)

`\xeCJK_add_font_features:Nnn`

```

2627 \cs_new_protected_nopar:Npn \xeCJK_add_font_features:Nnn #1#2#3
2628 {
2629   \prop_get:NVNTF \g__xeCJK_family_font_name_prop
2630   \l_xeCJK_family_tl \l__xeCJK_font_name_tl
2631   {
2632     \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
2633     \seq_map_inline:Nn \g__xeCJK_sub_key_seq
2634     { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
2635     \seq_clear:N \l__xeCJK_sub_key_seq
2636     \clist_clear:N \l__xeCJK_add_block_features_clist

```

```

2637 \clist_map_inline:nn {#2}
2638 {
2639     \seq_if_in:NnTF \g__xeCJK_sub_key_seq {##1}
2640     {
2641         \seq_put_right:Nn \l__xeCJK_sub_key_seq {##1}
2642         \__xeCJK_add_sub_class_features:n {##1}
2643     }
2644     { \__xeCJK_warning:nx { SubBlock-undefined } {##1} }
2645 }
2646 \bool_if:nT { #1 && \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
2647 {
2648     \seq_map_function:NN
2649     \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
2650 }
2651 \prop_get:NVNT \g__xeCJK_family_font_options_prop
2652 \l_xeCJK_family_tl \l__xeCJK_font_options_clist
2653 {
2654     \bool_if:nT
2655     { \seq_if_empty_p:N \l__xeCJK_sub_key_seq || #1 }
2656     {
2657         \clist_concat:NNN \l__xeCJK_font_options_clist
2658         \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
2659     }
2660     \clist_concat:NNN \l__xeCJK_font_options_clist
2661     \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
2662 }
2663 \xeCJK_fontspec:VV \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
2664 }
2665 { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
2666 }
2667 \clist_new:N \l__xeCJK_add_font_features_clist
2668 \clist_new:N \l__xeCJK_add_block_features_clist
2669 \cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nxx , Nnx }
2670 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
2671 {
2672     \token_to_str:N \addCJKfontfeature (s)-ignored.\\
2673     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
2674 }

```

(End definition for \xeCJK_add_font_features:Nnn.)

_xeCJK_add_sub_class_features:n

```

2675 \cs_new_protected_nopar:Npn \__xeCJK_add_sub_class_features:n #1
2676 {
2677     \prop_get:NoNTF \g__xeCJK_family_font_name_prop
2678     { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
2679     {
2680         \prop_get:NoN \g__xeCJK_family_font_options_prop
2681         { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
2682     }
2683     {
2684         \prop_get:NxNTF \g__xeCJK_family_font_name_prop
2685         { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
2686         {
2687             \prop_get:NxN \g__xeCJK_family_font_options_prop
2688             { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
2689         }
2690         {
2691             \prop_get:NVN \g__xeCJK_family_font_options_prop
2692             \l_xeCJK_family_tl \l__xeCJK_sub_font_options_clist
2693             \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl
2694         }
2695     }

```

```

2696 \clist_concat:NNN \l__xeCJK_sub_font_options_clist
2697 \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
2698 \clist_put_right:Nx \l__xeCJK_add_block_features_clist
2699 {
2700     #1 =
2701     {
2702         [ \exp_not:V \l__xeCJK_sub_font_options_clist ]
2703         { \exp_not:V \l__xeCJK_sub_font_name_tl }
2704     }
2705 }
2706 }
2707 \cs_generate_variant:Nn \prop_get:NnN { Nx }
2708 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
(End definition for \__xeCJK_add_sub_class_features:n)
在导言区结束的时候,若没有声明 CJK 字体,则给出一个警告。如果 \CJKfamilydefault 没有被更改,则在此时根据西文字体的情况更新 \CJKfamilydefault。如果 \CJKfamilydefault 对应的字体族没有定义,则使用 \CJKrmdefault 作为默认字体族。若 \CJKrmdefault 也没有定义,则使用在导言区设置的第一个 CJK 字体作为默认字体族。最后设置数学字体。
2709 \__xeCJK_at_end_preamble:n
2710 {
2711     \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
2712     \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
2713     {
2714         \tl_gset:Nx \CJKfamilydefault
2715         {
2716             \str_case_x:nnn { \familydefault }
2717             {
2718                 { \rmdefault } { \exp_not:N \CJKrmdefault }
2719                 { \sfdefault } { \exp_not:N \CJKsfdefault }
2720                 { \ttdefault } { \exp_not:N \CJKttdefault }
2721             }
2722             { \CJKfamilydefault }
2723         }
2724     }
2725     \cs_undefine:N \__xeCJK_family_default_wrap:n
2726     \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
2727     { \__xeCJK_warning:nx { no-CJKfamily } { \CJKfamilydefault } }
2728     {
2729         \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2730         {
2731             \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
2732             \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
2733             { \use:n }
2734             {
2735                 \xeCJK_family_if_exist:xF { \CJKrmdefault }
2736                 { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
2737             }
2738             {
2739                 \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
2740                 {
2741                     \prop_map_break:n
2742                     { \tl_gset_rescan:Nnn \CJKfamilydefault { } {#1} }
2743                 }
2744             }
2745             \__xeCJK_warning:nxx { CJKfamilydefault-undefined }
2746             { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
2747         }
2748         \xeCJK_switch_family:n { \CJKfamilydefault }
2749         \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
2750     }
2751 }
2752 \__xeCJK_msg_new:nn { no-CJKfamily }

```

```

2753 {
2754   It~seems~that~you~have~not~declare~a~CJKfamily.\\
2755   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
2756   `\\_xeCJK_msg_def_family_map:n {#1}'\\\\
2757   in~the~preamble~to~declare~the~default~CJKfamily.\\
2758 }
2759 \\_xeCJK_msg_new:nn { CJKfamilydefault-undefined }
2760 {
2761   Undefined~CJK~default~family~`\\_xeCJK_msg_family_map:n {#1}'~
2762   has~been~replaced~by~`\\_xeCJK_msg_family_map:n {#2}'~.\\\\
2763   Try~to~use~`\\_xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2764 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```

2765 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }
(End definition for CJKmath. This function is documented on page 3.)

```

\setCJKmathfont 设置 CJK 数学字体。

```

2766 \NewDocumentCommand \setCJKmathfont { 0 { } m }
2767 { \xeCJK_set_family:xxx { \c__xeCJK_math_tl } {#1} {#2} }
2768 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }

```

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

\xeCJK_set_mathfont: 当没有设置 CJK 数学字体时,使用 \CJKfamilydefault 作为数学字体。

```

2769 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
2770 {
2771   \xeCJK_family_if_exist:xF { \c__xeCJK_math_tl }
2772   { \tl_set:Nx \l__xeCJK_tmp_tl { \c__xeCJK_math_tl } }
2773   {
2774     \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2775     { \tl_set:Nx \l__xeCJK_tmp_tl { \CJKfamilydefault } }
2776     { \use_none:nnnnn }
2777   }
2778   \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmp_tl \l__xeCJK_tmp_tl
2779   {
2780     \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmp_tl }
2781     \DeclareSymbolFont \c__xeCJK_math_tl
2782     \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \mddefault \shapedefault
2783     \SetSymbolFont \c__xeCJK_math_tl { bold }
2784     \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \bfdefault \shapedefault
2785     \int_const:Nn \c__xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }
2786     \clist_concat:NNN \l__xeCJK_tmp_clist
2787     \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
2788     \clist_concat:NNN \l__xeCJK_tmp_clist
2789     \l__xeCJK_tmp_clist \c__xeCJK_FullRight_chars_clist
2790     \clist_map_inline:Nn \l__xeCJK_tmp_clist
2791     {
2792       \_xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
2793       { \c_seven } { \c__xeCJK_math_fam_int }
2794     }
2795   }
2796 }

```

(End definition for \xeCJK_set_mathfont:.)

\xeCJK_gset_mathcode:nnnn

```

2797 \cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
2798 {
2799   \_xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
2800   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
2801   {

```

```

2802 \tex_global:D \xeCJK_xetex_mathcode:w
2803 \l__xeCJK_begin_int = #3 ~ #4 \l__xeCJK_begin_int
2804 \int_incr:N \l__xeCJK_begin_int
2805 }
2806 }
(End definition for \xeCJK_gset_mathcode:nnnn.)

```

5.15 抄录环境中的间距调整

Verb 如果设置为 `env`, 则只在 \LaTeX 的抄录环境里使用 `\xeCJKVerbAddon`, 而不包括 `\verb`。对当前使用环境的判断基于在标准 \LaTeX 的环境定义里使用 `\begingroup` 和 `\endgroup` 来分组。

```

2807 \keys_define:nn { xecjk / options }
2808 {
2809   Verb .choice: ,
2810   Verb / true .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_zero } ,
2811   Verb / env .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_one } ,
2812   Verb / env+ .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_two } ,
2813   Verb / false .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_three } ,
2814   Verb .default:n = { env }
2815 }
2816 \int_new:N \l__xeCJK_verb_case_int
2817 \__xeCJK_after_preamble:n
2818 {
2819   \tl_put_right:Nn \verbatim@font
2820   {
2821     \if_case:w \l__xeCJK_verb_case_int
2822       \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2823       \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2824     \or:
2825       \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
2826         { \xeCJKVerbAddon }
2827         {
2828           \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2829           \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2830         }
2831     \or:
2832       \int_compare:nNnT \etex_currentgrouptype:D = \c_fourteen
2833         { \xeCJKVerbAddon }
2834     \fi:
2835   }
2836 }

```

(End definition for *Verb*. This function is documented on page 5.)

`\xeCJKVerbAddon` 进行了比较大的调整, 应该只在分组环境里使用。为了方便调整间距以利于对齐, 这里只把字符分成了两类, 并且在 `CJK` 类与边界(空格)之间也插入 `\CJKecglue`。当然, 这样做之后, 关于标点符号的禁则就没有了。

```

2837 \NewDocumentCommand \xeCJKVerbAddon { }
2838 {
2839   \int_compare:nNnF \etex_currentgrouplevel:D = \c_zero
2840   {
2841     \bool_if:NF \l__xeCJK_listings_env_bool
2842     {
2843       \__xeCJK_set_verb_exspace:
2844       \__xeCJK_verb_addon:
2845     }
2846   }
2847 }
2848 \bool_new:N \l__xeCJK_listings_env_bool
2849 \NewDocumentCommand \xeCJKOffVerbAddon { } { }
2850 \cs_new_protected_nopar:Npn \__xeCJK_verb_addon:
2851 {
2852   \bool_if:NF \l__xeCJK_verb_addon_bool

```

```

2853 {
2854   \bool_set_true:N \l__xeCJK_verb_addon_bool
2855   \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
2856   \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
2857   \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
2858   \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
2859   \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
2860   \cs_set_eq:NN \__xeCJK_verb_CJKglue: \CJKglue
2861   \cs_set_eq:NN \__xeCJK_verb_CJKecglue: \CJKecglue
2862   \cs_set_eq:NN \__xeCJK_verb_check_for_glue: \xeCJK_check_for_glue:
2863   \cs_set_eq:NN \__xeCJK_verb_ignore_spaces:w \xeCJK_ignore_spaces:w
2864   \cs_set_protected_nopar:Npx \xeCJKOffVerbAddon
2865   {
2866     \__xeCJK_reset_char_class:n { FullLeft }
2867     \__xeCJK_reset_char_class:n { FullRight }
2868     \__xeCJK_reset_char_class:n { HalfLeft }
2869     \__xeCJK_reset_char_class:n { HalfLeft }
2870     \__xeCJK_reset_char_class:n { NormalSpace }
2871     \skip_set:Nn \tex_rightskip:D { \skip_use:N \tex_rightskip:D }
2872     \exp_not:c
2873     { \bool_set_ \bool_if:NTF \l__xeCJK_xecglue_bool { true } { false } :N }
2874     \l__xeCJK_xecglue_bool
2875     \exp_not:n
2876     {
2877       \cs_set_eq:NN \CJKglue \__xeCJK_verb_CJKglue:
2878       \cs_set_eq:NN \CJKecglue \__xeCJK_verb_CJKecglue:
2879       \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_verb_check_for_glue:
2880       \cs_set_eq:NN \xeCJK_ignore_spaces:w \__xeCJK_verb_ignore_spaces:w
2881     }
2882   }
2883   \tex_output:D \exp_after:wN \exp_after:wN \exp_after:wN
2884   {
2885     \exp_after:wN \exp_after:wN
2886     \exp_after:wN \xeCJKOffVerbAddon
2887     \exp_after:wN \use:n
2888     \tex_the:D \tex_output:D
2889   }
2890   \xeCJKsetup { xCJKecglue = false }
2891   \cs_set_protected_nopar:Npn \CJKglue
2892   { \skip_horizontal:N \l__xeCJK_verb_exspace_skip }
2893   \cs_set_protected_nopar:Npn \CJKecglue
2894   { \skip_horizontal:n { 0.5 \l__xeCJK_verb_exspace_skip } }
2895   \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
2896   \cs_set_eq:NN \xeCJK_ignore_spaces:w \CJKecglue
2897   \skip_set:Nn \tex_rightskip:D { \c_zero_dim plus 1 fil }
2898 }
2899 }
2900 \cs_new_protected_nopar:Npn \__xeCJK_reset_char_class:n #1
2901 {
2902   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
2903   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
2904   { \XeTeXcharclass ##1 = \l__xeCJK_tmp_int }
2905 }
2906 \bool_new:N \l__xeCJK_verb_addon_bool
2907 \cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon

```

(End definition for `\xeCJKVerbAddon` and `\xeCJKOffVerbAddon`. These functions are documented on page 12.)

`__xeCJK_set_verb_exspace:` 在抄录环境中, CJK 文字之间的间距为当前西文字体两个空格的宽度与当前字体大小之差, 而与西文和空格的间距为 CJK 文字之间的间距的一半。

```

2908 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
2909 {
2910   \tl_if_exist:cTF { xeCJK/verb/\l__xeCJK_family_tl/\curr@fontshape/\f@size }

```



```

2911 {
2912   \skip_set:Nn \l__xeCJK_verb_exspace_skip
2913   { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2914 }
2915 {
2916   \tl_set:Nx \l__xeCJK_current_coor_tl { \l_xeCJK_family_tl/\curr@fontshape }
2917   \prop_get:NWTF \g__xeCJK_scale_family_prop
2918   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
2919   {
2920     \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2921     \skip_zero:N \l__xeCJK_verb_exspace_skip
2922   }
2923   {
2924     \group_begin: \xeCJK_select_font: \exp_after:wN \group_end:
2925     \exp_after:wN \__xeCJK_set_verb_exspace:n
2926     \exp_after:wN { \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00 }
2927   }
2928 }
2929 }
2930 \skip_new:N \l__xeCJK_verb_exspace_skip
(End definition for \__xeCJK_set_verb_exspace:.)

```

`__xeCJK_set_verb_exspace:n` 当两个西文空格的宽度小于一个 CJK 文字的宽度时,对目前使用的 CJK 字体进行适当缩小。

```

2931 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:n #1
2932 {
2933   \skip_set:Nn \l__xeCJK_verb_exspace_skip
2934   { \c_two \tex_fontdimen:D \c_two \tex_font:D - #1 }
2935   \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
2936   {
2937     \skip_zero:N \l__xeCJK_verb_exspace_skip
2938     \use:x
2939     {
2940       \__xeCJK_set_verb_scale:nn
2941       { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } }
2942       { \dim_to_fp:n {#1} }
2943     }
2944   }
2945   {
2946     \tl_const:cx { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
2947     { \skip_use:N \l__xeCJK_verb_exspace_skip }
2948   }
2949 }
(End definition for \__xeCJK_set_verb_exspace:n)

```

`__xeCJK_set_verb_scale:nn` 缩小 CJK 字体,并保存相关信息。

```

2950 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_scale:nn #1#2
2951 {
2952   \fp_set:Nn \l__xeCJK_scale_factor_fp { #1 / #2 }
2953   \__xeCJK_warning:nxx { scale-factor }
2954   { \fp_eval:n { round - ( \l__xeCJK_scale_factor_fp , 2 ) } }
2955   { \fp_eval:n { round + ( #2 / #1 , 2 ) } }
2956   \xeCJK_add_font_features:Nnx \c_true_bool
2957   { } { Scale = { \fp_use:N \l__xeCJK_scale_factor_fp } }
2958   \prop_gput:NVV \g__xeCJK_scale_family_prop
2959   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
2960 }
2961 \__xeCJK_msg_new:nn { scale-factor }
2962 {
2963   \token_to_str:N \xeCJKVerbAddon'-may~not~work~properly.\\
2964   You~may~set~`Scale=#1'~to~CJKfamily~
2965   \__xeCJK_msg_family_map:n { \l_xeCJK_family_tl }',\\
2966   or~set~`Scale=#2'~to~family~

```

```

2967 \str_if_eq:x:nnTF \f@family \ttdefault
2968 { \token_to_str:N \ttdefault } { \f@family }'.
2969 }
2970 \fp_new:N \l__xeCJK_scale_factor_fp
2971 \prop_new:N \g__xeCJK_scale_family_prop
(End definition for \__xeCJK_set_verb_scale:nm.)

```

\xeCJK_visible_space_fallback: fontspec 使用 lmtt 字体中的可视空格符号(U+2423)作为当前字体中相应符号的后备。但是 lmtt 的字体大小未必与当前字体匹配。因此,我们在这里做一些调整,以保证使用后备可视空格符号时,也能保证对齐。这里加入 \scan_stop: 的目的是强制发生状态转移。这样当空格出现在 CJK 文字后面时,使字体回到西文,以便让 fontspec 在当前西文字体而不是在 CJK 字体中检查有没有 U+2423。

```

2972 \cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
2973 { {
2974   \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
2975   { \xeCJK_set_visible_space_font: }
2976   \textvisiblespace
2977 } }
2978 \AtEndOfPackage
2979 {
2980   \tl_put_left:Nn \fontspec_visible_space: { \scan_stop: }
2981   \cs_gset_eq:NN \fontspec_visible_space_fallback: \xeCJK_visible_space_fallback:
2982 }
(End definition for \xeCJK_visible_space_fallback:.)

```

\xeCJK_set_visible_space_font: 当前字体空格的宽度与后备字体 lmtt 不一样时,就对 \textvisiblespace 的字体尺寸按相应的比例放缩。

```

2983 \cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
2984 {
2985   \tl_set:Nx \l__xeCJK_current_coor_tl { xeCJK/space/\curr@fontshape/\f@size }
2986   \exp_after:wN \__xeCJK_set_visible_space_size:n
2987   \exp_after:wN { \dim_use:N \tex_fontdimen:D \c_two \tex_font:D }
2988   \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2989 }
2990 \cs_new_protected_nopar:Npn \__xeCJK_set_visible_space_size:n #1
2991 {
2992   \tl_set:Nx \f@encoding { \g_fontspec_encoding_tl }
2993   \tl_set:Nx \f@family { lmtt }
2994   \selectfont
2995   \dim_compare:nNnF {#1} = { \tex_fontdimen:D \c_two \tex_font:D }
2996   {
2997     \fontsize
2998     {
2999       \dim_eval:n
3000       {
3001         \f@size pt *
3002         \dim_ratio:nn {#1} { \tex_fontdimen:D \c_two \tex_font:D }
3003       }
3004     }
3005     { \f@baselineskip }
3006     \selectfont
3007   }
3008 }
(End definition for \xeCJK_set_visible_space_font:.)

```

5.16 xeCJK 其它选项

LocalConfig 声明载入本地配置文件的选项。

```

3009 \keys_define:nn { xeCJK / options }
3010 {
3011   LocalConfig .choice: ,

```

```

3012 LocalConfig / false .code:n =
3013 { \bool_gset_false:N \g__xeCJK_config_bool } ,
3014 LocalConfig / true .code:n =
3015 {
3016 \bool_gset_true:N \g__xeCJK_config_bool
3017 \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
3018 } ,
3019 LocalConfig / unknown .code:n =
3020 {
3021 \bool_gset_true:N \g__xeCJK_config_bool
3022 \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
3023 } ,
3024 LocalConfig .default:n = { true }
3025 }
3026 \tl_new:N \g__xeCJK_config_name_tl
3027 \bool_new:N \g__xeCJK_config_bool
(End definition for LocalConfig. This function is documented on page 2.)

```

indentfirst 首行是否缩进。

```

3028 \keys_define:nn { xeCJK / options }
3029 {
3030 CJKnumber .bool_gset:N = \g__xeCJK_number_bool ,
3031 indentfirst .bool_gset:N = \g__xeCJK_indent_bool ,
3032 normalindentfirst .meta:n = { indentfirst = false }
3033 }
(End definition for indentfirst. This function is documented on page 3.)

```

quiet 将调用 xeCJK 时使用的未知的选项传递给 fontspec 宏包。对 fontspec 的 quiet 和 silent 选项进行修改,使其适用于 xeCJK。
silent

```

3034 \keys_define:nn { xeCJK / options }
3035 {
3036 quiet .code:n =
3037 {
3038 \msg_redirect_module:nnn { xeCJK } { warning } { info }
3039 \msg_redirect_module:nnn { xeCJK } { info } { none }
3040 \xeCJK_if_package_loaded:nF { fontspec }
3041 { \PassOptionsToPackage { quiet } { fontspec } }
3042 } ,
3043 silent .code:n =
3044 {
3045 \msg_redirect_module:nnn { xeCJK } { warning } { none }
3046 \msg_redirect_module:nnn { xeCJK } { info } { none }
3047 \xeCJK_if_package_loaded:nF { fontspec }
3048 { \PassOptionsToPackage { silent } { fontspec } }
3049 } ,
3050 unknown .code:n =
3051 {
3052 \xeCJK_if_package_loaded:nTF { fontspec }
3053 { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
3054 { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
3055 }
3056 }
3057 \__xeCJK_msg_new:nn { key-unknown }
3058 {
3059 Sorry,~but~\l_keys_module_tl \ does~not~have~a~key~called~`#1'.\\\\
3060 The~key~`#1'~is~being~ignored.
3061 }
(End definition for quiet and silent.)

```

5.17 xeCJK 初始化设置

```

\CJKsymbol
\CJKpunctsymbol 3062 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
3063 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}
(End definition for \CJKsymbol and \CJKpunctsymbol.)
xeCJK 宏包的初始化设置。

3064 \keys_set:nn { xeCJK / options }
3065 {
3066   CJKglue      = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
3067   CJKecglue    = { ~ } ,
3068   xCJKecglue   = false ,
3069   CheckSingle  = false ,
3070   PlainEquation = false ,
3071   CheckFullRight = false ,
3072   CJKspace     = false ,
3073   CJKmath      = false ,
3074   CJKnumber    = false ,
3075   xeCJKactive  = true  ,
3076   LocalConfig  = true  ,
3077   indentfirst  = true  ,
3078   Verb         = env   ,
3079   EmboldenFactor = 4    ,
3080   SlantFactor  = 0.167 ,
3081   PunctStyle   = quanjiao ,
3082   NewLineCS    = { \par \[ } ,
3083   EnvCS        = { \begin \end } ,
3084   NoBreakCS    = { \footnote \footnotemark \nobreak } ,
3085   KaiMingPunct = { ~~~~3002 ~~~~ff0e ~~~~ff1f ~~~~ff01 } ,
3086   LongPunct    = { ~~~~2014 ~~~~2015 ~~~~2500 ~~~~2025 ~~~~2026 } ,
3087   MiddlePunct  = { ~~~~2014 ~~~~2015 ~~~~2027 ~~~~2500 ~~~~00b7 ~~~~30fb ~~~~ff65 } ,
3088   AllowBreakBetweenPuncts = false
3089 }
3090 \defaultCJKfontfeatures { Script = CJK }

执行宏包选项,并载入 fontspec 宏包。

3091 \ProcessKeysOptions { xeCJK / options }
3092 \RequirePackage { fontspec } [ 2011/09/13 ]

\c__xeCJK_encoding_tl 保存 fontspec 声明字体时使用的字体编码。

3093 \tl_const:Nx \c__xeCJK_encoding_tl { \g_fontspec_encoding_tl }
(End definition for \c__xeCJK_encoding_tl.)
章节标题后面的首个段落的首行是否缩进。

3094 \bool_if:NT \g__xeCJK_indent_bool { \RequirePackage { indentfirst } }
对不能通过 \xeCJKsetup 设置的选项给出警告。

3095 \keys_define:nn { xeCJK / options }
3096 {
3097   LocalConfig .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
3098   CJKnumber   .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
3099   indentfirst .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } }
3100 }
3101 \__xeCJK_msg_new:nn { option-invalid }
3102 {
3103   The~`#1'~option-only-can-be-set-in-the-optional-argument-to-the\\
3104   \token_to_str:N \usepackage \ command-when-xeCJK-is-being-loaded.\\
3105   Please-do-not-set-it-via-the-\token_to_str:N \xeCJKsetup \ command.
3106 }

\CJKrmdefault
\JKsfdefault 3107 \tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
\JKttdefault 3108 \tl_if_exist:NF \JKsfdefault { \tl_gset:Nn \JKsfdefault { sf } }
\JKfamilydefault

```

```

3109 \tl_if_exist:NF \CJKttdefault { \tl_gset:Nn \CJKttdefault { tt } }
3110 \tl_new:N \l__xeCJK_family_default_init_tl
3111 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
3112 \tl_gset:Nx \l__xeCJK_family_default_init_tl
3113 {
3114   \exp_not:N \__xeCJK_family_default_wrap:n
3115   {
3116     \tl_if_exist:NTF \CJKfamilydefault
3117       { \exp_not:V \CJKfamilydefault }
3118       { \exp_not:N \CJKrmdefault }
3119   }
3120 }
3121 \tl_gset_eq:NN \CJKfamilydefault \l__xeCJK_family_default_init_tl
(End definition for \CJKrmdefault and others. These variables are documented on page 6.)

```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```

3122 \NewDocumentCommand \xeCJKsetup { +m }
3123 {
3124   \keys_set:nn { xeCJK / options } {#1}
3125   \tex_ignorespaces:D
3126 }
(End definition for \xeCJKsetup. This function is documented on page 2.)

```

\xeCJKsetemboldenfactor
\xeCJKsetslantfactor

```

3127 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
3128 { \xeCJKsetup { EmboldenFactor = {#1} } }
3129 \NewDocumentCommand \xeCJKsetslantfactor { m }
3130 { \xeCJKsetup { SlantFactor = {#1} } }
(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor.)

```

\punctstyle
\xeCJKplainchr

```

3131 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
3132 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }
(End definition for \punctstyle and \xeCJKplainchr.)

```

\CJKsetecglue

```

3133 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
3134 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
(End definition for \CJKsetecglue.)

```

\CJKspace
\CJKnospace

```

3135 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
3136 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }
(End definition for \CJKspace and \CJKnospace.)

```

\xeCJKallowbreakbetweenpuncts

\xeCJKnobreakbetweenpuncts

```

3137 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
3138 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
3139 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
3140 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }
(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts.)

```

\xeCJKenablefallback
\xeCJKdisablefallback

```

3141 \NewDocumentCommand \xeCJKenablefallback { }
3142 { \xeCJKsetup { AutoFallBack = true } }
3143 \NewDocumentCommand \xeCJKdisablefallback { }
3144 { \xeCJKsetup { AutoFallBack = false } }
(End definition for \xeCJKenablefallback and \xeCJKdisablefallback.)

```

`\xeCJKsetcharclass`

```
3145 \NewDocumentCommand \xeCJKsetcharclass { m m m }
3146 {
3147   \xeCJK_set_char_class:nnn {#1} {#2} {#3}
3148   \xeCJKResetPunctClass
3149 }
(End definition for \xeCJKsetcharclass.)
```

5.18 兼容性修补

`\nobreakspace` 按照 xunicode 的设计,使用 `\DeclareUTFcharacter` 定义符号命令时,如果命令已经存在,那么它会检查它是否是“健壮”的。如果是,则什么也不做;否则使用 `\DeclareTextCommand` 来重定义它,实际意义与原来相同,原来的定义保存在名为 `?-\cs` 的宏中。这里的主要问题是 xunicode 的健壮性检查对像 `\nobreakspace` 这种原本由 `\DeclareRobustCommand` 定义的命令返回结果是“假”,因而重定义它。如果编码发生变化,使用 `\nobreakspace` 时,NFSS 就会报错。按照 NFSS 的机制,我们用 `\ProvideTextCommandDefault` 对未声明的编码给出默认定义即可避免报错。其实 xunicode 重定义 `\nobreakspace` 的方式并不恰当。因为在 $\text{\LaTeX} 2_{\epsilon}$ 中,`\nobreakspace` 原本并不是由 `\DeclareTextCommand` 或 `\DeclareTextSymbol` 来定义,这样按照 xunicode 的实现方式,`\nobreakspace` 永远都不会输出 `\char"00A0`。要想让 `\nobreakspace` 在 `\UTFencname` 编码下面输出 `\char"00A0`,除非在 xunicode 载入前用 `\DeclareTextCommandDefault` 重新定义它:

```
\DeclareTextCommandDefault\nobreakspace{\leavevmode\nobreak\ }
```

这样我们也就不需要再做下面的处理。另外,如果作者使用名字 `?-\cs` 来保存原来的定义的话,也不需要下面的处理。因为下面实际上是定义了一个名为 `?-\nobreakspace` 的宏作为未声明编码的默认定义。但是作者使用了 `?-\cs`,不知道这是有意为之还是一个疏忽。

```
3150 \cs_if_exist:cT { ? - \token_to_str:N \nobreakspace }
3151 {
3152   \exp_args:NNv \ProvideTextCommandDefault \nobreakspace
3153   { ? - \token_to_str:N \nobreakspace }
3154 }
(End definition for \nobreakspace.)
```

`\xeCJK@text@composite` 这里修改的目的是让 `\'{\~e}` 等直接展开成在 xunicode 中由下面定义的 `\char"1EBF\relax`,也就是让这些定义生效:

```
\DeclareUTFcomposite[\UTFencname]{x1EBF}{\'}{\~e}
```

按照 source2e 中 `\@text@composite` 的定义,目前 xunicode 对这种三个记号组合的定义不会起作用。`\'{\~e}` 将会被展开成 `\char"00EA\relax\char"0301\relax` 这种组合字符。由于中间的 `\relax` 会造成状态转移,按照 xeCJK 的机制,就会插入 kern,最终破坏了 \XeTeX 将两个字符正确的组合起来。除此之外,还可以将 xunicode 中所有 `\char` 数字后面的 `\relax` 全部改成普通空格,这样就不会留下多余的 `\relax`。或者使用 lowercase 技巧,直接由 Unicode 得到实际字符,而不需要通过 `\char`。但是,如果在 xeCJK 中修改的话,还要 `\ReloadXunicode` 才会起作用。需要注意的是,如果原来字体中缺少 `é` 这个字符,但是有 `ê` 这个字符的话,`\'{\~e}` 会输出为 `ê`,现在则是留下一个空白。

```
3155 \cs_new_nopar:Npn \xeCJK@text@composite #1#2#3 \@text@composite
3156 {
3157   \str_if_eq:nnTF {#3} { } \@empty {
3158     { \orig@text@composite #1#2#3 \@text@composite }
3159     { \_xeCJK_text_composite:Nnw #1#2#3 }
3160   }
3161   \cs_new_nopar:Npn \_xeCJK_text_composite:Nnw #1#2 \@empty
3162   {
3163     \tl_if_empty:nTF {#2}
3164     { \orig@text@composite #1 \@empty \@text@composite }
3165     { \cs_if_exist_use:cF { \token_to_str:N #1 - \token_to_str:N #2 } }
3166   }
3167   \cs_new_eq:NN \orig@text@composite \@text@composite
3168   \cs_set_eq:NN \@text@composite \xeCJK@text@composite
```

(End definition for `\xeCJK@text@composite`.)

```

3169 \_xeCJK_msg_new:nn { conflict-package }
3170 {
3171   The~`#1'~package~is~too~old. \\
3172   Please~update~an~up~to~date~version~of~it\\
3173   using~your~TeX~package~manager~or~from~CTAN.
3174 }
3175 \xeCJK_if_package_loaded:nTF { realscripts }
3176 {
3177   \@ifpackagelater { realscripts } { 2010/10/10 } { }
3178   {
3179     \_xeCJK_error:nx { conflict-package }
3180     {
3181       \xeCJK_if_package_loaded:nTF { xltextra }
3182       { xltextra } { realscripts }
3183     }
3184   }
3185 }
3186 {
3187   \cs_new_eq:NN \_xeCJK_dim_max:nn \dim_max:nn
3188   \cs_new_eq:NN \_xeCJK_dim_min:nn \dim_min:nn
3189   \_xeCJK_at_end_preamble:n
3190   {
3191     \xeCJK_if_package_loaded:nT { realscripts }
3192     {
3193       \@ifpackagelater { realscripts } { 2010/10/10 } { }
3194       {
3195         \cs_gset_eq:NN \dim_max:nn \_xeCJK_dim_max:nn
3196         \cs_gset_eq:NN \dim_min:nn \_xeCJK_dim_min:nn
3197       }
3198     }
3199     \cs_undefine:N \_xeCJK_dim_max:nn
3200     \cs_undefine:N \_xeCJK_dim_min:nn
3201   }
3202 }

```

`\fontfamily` 修改 `\fontfamily`, 使主要 CJK 字体族能随西文主要字体更新。

```

3203 \RenewDocumentCommand \fontfamily { m }
3204 {
3205   \tl_set:Nx \f@family {#1}
3206   \_xeCJK_update_family:n
3207   {
3208     \str_case:nnn {#1}
3209     {
3210       { \rmdefault } { \xeCJK_switch_family:n { \CJKrmdefault } }
3211       { \sfdefault } { \xeCJK_switch_family:n { \CJKsfdefault } }
3212       { \ttdefault } { \xeCJK_switch_family:n { \CJKttdefault } }
3213       { \familydefault } { \xeCJK_switch_family:n { \CJKfamilydefault } }
3214     }
3215     { }
3216   }
3217 }
3218 \cs_new_eq:NN \_xeCJK_update_family:n \use:n

```

(End definition for `\fontfamily`.)

`\xeCJK@fix@penalty` 对 $\text{\LaTeX 2}_{\epsilon}$ 内核中的 `\fix@penalty` 被用于诸如 `\textit` 之类的文档字体转换命令的定义之中。这里对它进行补丁的目的是修复其中的倾斜校正, 并使得这些文档命令与紧随其后的汉字之间可以正确的插入 `\CJKecglue` 或者忽略其中的空格。例如 这是`\emph{强调}`文本, 第二个空格可以被忽略掉。如果使用 `\xeCJKecglue` 选项, 第一个空格也可以被省略。事实上, 在 `\sw@slant` 的定义中, `\@@italiccorr` 前面的 `\lastskip` 和 `\lastpenalty` 有四种情况, 这里只对它们都为零的情况进行处理。

```

3220 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty

```

```

3221 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
3222 \tl_replace_once:Nnn \sw@slant { \fix@penalty } { \xeCJK@fix@penalty }
(End definition for \xeCJK@fix@penalty.)

```

`\xeCJK@italiccorr` 修复倾斜校正, 并处理汉字后面的空格。

```

3223 \cs_new_protected_nopar:Npn \xeCJK@italiccorr
3224 {
3225   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
3226   {
3227     \xeCJK_if_last_node:nTF { default }
3228     {
3229       \xeCJK_remove_node: \@@italiccorr
3230       { \xeCJK_make_node:n { default } }
3231     }
3232     {
3233       \xeCJK_if_last_node:nTF { CJK }
3234       {
3235         \xeCJK_remove_node: \@@italiccorr
3236         { \xeCJK_make_node:n { CJK } } \use:n
3237       }
3238       {
3239         \xeCJK_if_last_node:nTF { CJK-space }
3240         {
3241           \xeCJK_remove_node: \@@italiccorr
3242           { \xeCJK_make_node:n { CJK-space } } \use:n
3243         }
3244         { \@@italiccorr \use_none:n }
3245       }

```

`\xeCJK_ignore_spaces:w` 里面用到 `peek` 函数来判断后面是不是空格, 而此时它后面还有 4 个 `\fi` 或者 `\else...\fi` 没有被展开, 将影响 `peek` 函数的判断。因此我们需要用 $2^4 - 1 = 15$ 个 `\exp_after:wN` 来展开它们。显然, 这里用 `\exp_last_unbraced:Nf` 会比较方便, 但是它会吃掉 `\textit{...}` 等后面原来存在的空格作为完全展开的结束。要正确使用它还需要另外的处理 (使用 `\exp_stop_f:`)。

```

3246   {
3247     \exp_after:wN \exp_after:wN \exp_after:wN
3248     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3249     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3250     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3251     \xeCJK_ignore_spaces:w
3252   }
3253 }
3254 }
3255 { \@@italiccorr }
3256 }

```

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

`__xeCJK_set_others_toks:n` 简单处理与同样使用 `\XeTeXinterchartoks` 机制的宏包的兼容问题。

```

3258 __xeCJK_after_end_preamble:n
3259 {
3260   \int_compare:nNnF
3261   { \c_three + \seq_count:N \g__xeCJK_new_class_seq } = \xe@alloc@intercharclass
3262   {
3263     \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
3264     {
3265       \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
3266       { __xeCJK_set_others_toks:n {#1} }
3267     }
3268   }
3269 }
3270 \cs_new_protected_nopar:Npn __xeCJK_set_others_toks:n #1
3271 {

```



```

3272 \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
3273 \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
3274 {
3275   \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
3276   \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
3277   \xeCJK_app_inter_class_toks:nnx {##1} { Others }
3278   { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
3279   \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
3280   { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
3281   \xeCJK_if_blank_x:nT
3282   { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
3283   {
3284     \xeCJK_copy_inter_class_toks:nnnn
3285     { Others } { Boundary } { Default } { Boundary }
3286   }
3287   \xeCJK_if_blank_x:nT
3288   { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
3289   {
3290     \xeCJK_copy_inter_class_toks:nnnn
3291     { Boundary } { Others } { Boundary } { Default }
3292   }
3293 }
3294 }
(End definition for \__xeCJK_set_others_toks:n.)

```

__xeCJK_group_begin: 用于保护下面歧义宽度标点的分组。

```

\__xeCJK_group_end:
3295 \cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
3296 \cs_new_eq:NN \__xeCJK_group_end: \group_end:

```

(End definition for __xeCJK_group_begin: and __xeCJK_group_end:.)

单独处理宽度有分歧的几个标点：包括省略号、破折号、间隔号、引号等中西文混用的符号，保证其命令形式输出的是西文字体。并对一些编码的符号宏包做特殊处理。在使用 T1 编码的时候，\r{u} 的实际定义是 \char183 与常被用作中文间隔号的 U+00B7 冲突。

```

3297 \__xeCJK_after_preamble:n
3298 {
3299   \tl_map_inline:nn
3300   {
3301     \textellipsis \textendash \textperiodcentered \textcentereddot
3302     \textquoteleft \textquoteright \textquotedblleft \textquotedblright
3303   }
3304   {
3305     \cs_gset_nopar:Npx #1
3306     { \__xeCJK_group_begin: \makexeCJKinactive \exp_not:o {#1} \__xeCJK_group_end: }
3307   }
3308   \tl_put_left:Nn \tipaencoding { \makexeCJKinactive }
3309   \cs_new_eq:NN \__xeCJK_aux_r:n \r
3310   \cs_set_nopar:Npn \r #1
3311   {
3312     \bool_if:nTF
3313     {
3314       \str_if_eq_x_p:nn { \f@encoding } { T1 } &&
3315       \str_if_eq_x_p:nn {#1} { u }
3316     }
3317     { { \makexeCJKinactive \__xeCJK_aux_r:n {#1} } }
3318     { \__xeCJK_aux_r:n {#1} }
3319   }
3320   \xeCJK_if_package_loaded:nT { pifont }
3321   {
3322     \RenewDocumentCommand \Pifont { m }
3323     { \makexeCJKinactive \usefont { U } {#1} { m } { n } }
3324   }
3325 }

```

简单处理与 hyperref 宏包的兼容问题。

```

3326 \__xeCJK_after_end_preamble:n
3327 {
3328   \xeCJK_if_package_loaded:nT { hyperref }
3329   {
3330     \pdfstringdefDisableCommands
3331     {
3332       \__xeCJK_gobble_CJKfamily:
3333       \xeCJK_cs_clear:N \makeCJKinactive
3334       \xeCJK_cs_clear:N \__xeCJK_group_begin:
3335       \xeCJK_cs_clear:N \__xeCJK_group_end:
3336     }
3337   }
3338 }

```

当探测到 cprotect 宏包被引入时,则取消 \cprotect 宏的 \outer 定义。

```

3339 \__xeCJK_after_end_preamble:n
3340 {
3341   \bool_if:nT
3342   { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
3343   { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
3344 }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

3345 \cs_if_exist:NF \CJK@ifundefined
3346 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
3347 \NewDocumentCommand \xeCJKcaption { o m }
3348 {
3349   \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
3350   \use:x
3351   {
3352     \char_set_catcode_letter:n { 64 }
3353     \file_input:n { #2.cpx }
3354     \char_set_catcode:nm { 64 } { \char_value_catcode:n { 64 } }
3355   }
3356   \XeTeXdefaultencoding "UTF-8"
3357 }

```

(End definition for \xeCJKcaption.)

由于 xeCJK 禁止 CJKulem 的载入,因此当使用 ctex 宏包的 fntef 选项时,就会出现 \normalem 没有定义的问题。此时改用 xeCJKfntef 以便载入 ulem。

判断过于繁琐,应该在 ctex 包中妥善处理。这段代码应在 ctex 包发布新版本后删去。

```

3358 \cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
3359 { \AtEndOfPackage { \RequirePackage { xeCJKfntef } } }
3360 {
3361   \__xeCJK_at_end_preamble:n
3362   {
3363     \xeCJK_if_package_loaded:nF { xeCJKfntef }
3364     {
3365       \xeCJK_if_package_loaded:nTF { CJKfntef }
3366       { \RequirePackage { xeCJKfntef } }
3367       {
3368         \xeCJK_if_package_loaded:nT { ulem }
3369         { \RequirePackage { xeCJKfntef } }
3370       }
3371     }
3372   }
3373 }

```

导言区末尾检测到 listings 时,自动载入 xeCJK-listings。

```

3374 \__xeCJK_at_end_preamble:n
3375 {
3376   \xeCJK_if_package_loaded:nT { listings }
3377   { \RequirePackage { xeCJK-listings } }

```

3378 }

\CJKaddEncHook 为使用 CJKnumb 宏包而作一些处理。另外 CJKnumb 使用的是传统汉字“萬”和“億”，我们在这里把它们修正为简体字。

```
3379 \cs_new_protected:Npn \CJKaddEncHook #1#2
3380 {
3381   \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
3382   {
3383     \group_begin:
3384     \cs_set_nopar:Npn \Unicode ##1##2
3385       { (##1) * \c_two_hundred_fifty_six + (##2) }
3386     \cs_set_eq:NN \def \xeCJK_char_from_charcode:Nn
3387       #2
3388     \group_end:
3389     \tl_gset:Nn \CJK@tenthousand { ~~~~4e07 }
3390     \tl_gset:Nn \CJK@hundredmillion { ~~~~4ebf }
3391   }
3392 }
3393 \cs_new_protected_nopar:Npn \xeCJK_char_from_charcode:Nn #1#2
3394 {
3395   \group_begin:
3396   \char_set_lccode:nn { "4E00 } {#2}
3397   \tl_to_lowercase:n
3398   {
3399     \group_end:
3400     \tl_const:Nn #1 { ~~~~4e00 }
3401   }
3402 }
3403 \bool_if:NT \g__xeCJK_number_bool { \RequirePackage { CJKnumb } }
```

(End definition for \CJKaddEncHook.)

最后引入本地配置文件。

```
3404 \bool_if:NT \g__xeCJK_config_bool
3405 {
3406   \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
3407   \@onefilewithoptions
3408   { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
3409 }
3410 </package>
```

5.19 xeCJKfntef

3411 < *fntef >

xeCJKfntef 不需要 CJKulem 宏包的支持,因此当使用 CJKfntef 时,需要另行载入 ulem。

```
3412 \PassOptionsToPackage { normalem } { ulem }
3413 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
3414 \ProcessOptions \scan_stop:
3415 \RequirePackage { xeCJK }
3416 \RequirePackage { ulem }
3417 \RequirePackage { CJKfntef }
3418 \RequirePackage { environ }
3419 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }
```

\xeCJK_hook_for_ulem:

```
3420 \cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
3421 {
3422   \bool_if:NF \l__xeCJK_ulem_hook_used_bool
3423   {
3424     \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
3425     \xeCJKsetup { CheckFullRight = false , xCJKecglue = false }
3426     \bool_if:NTF \l__xeCJK_ulem_skip_punct_bool
```

```

3427         { \cs_set_eq:NN \__xeCJK_ulem_leader_type: \UL@leadtype }
3428     {
3429         \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
3430         \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
3431     }
3432     \__xeCJK_ulem_initial:
3433     \xeCJK_glue_to_skip:nN
3434     {
3435         \cs_set_eq:NN \ \tex_space:D
3436         \cs_set_eq:NN \penalty \tex_penalty:D
3437         \cs_set_eq:NN \hskip \skip_horizontal:N
3438         \CJKglue
3439     } \l__xeCJK_ccglue_skip
3440     \xeCJK_glue_to_skip:nN
3441     {
3442         \cs_set_eq:NN \ \tex_space:D
3443         \cs_set_eq:NN \penalty \tex_penalty:D
3444         \cs_set_eq:NN \hskip \skip_horizontal:N
3445         \CJKkecglue
3446     } \l__xeCJK_ecglue_skip
3447     \cs_set_protected_nopar:Npn \CJKglue
3448     { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
3449     \cs_set_protected_nopar:Npn \CJKkecglue
3450     { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
3451 }
3452 }
3453 \bool_new:N \l__xeCJK_ulem_hook_used_bool
(End definition for \xeCJK_hook_for_ulem:.)

```

\CJK@UL 修改 CJKfntef 中的 \CJK@UL 和 \CJK@@UL 以适应下面的修改。
\CJK@@UL

```

3455 \cs_set_eq:NN \CJK@UL \CJK@@UL
3456 \tl_replace_once:Nnn \CJK@UL { \ULon }
3457 { \bool_set_true:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3458 \tl_replace_once:Nnn \CJK@@UL { \ULon }
3459 { \bool_set_false:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3460 \bool_new:N \l__xeCJK_ulem_skip_punct_bool
(End definition for \CJK@UL and \CJK@@UL.)

```

```

\__xeCJK_ulem_skip_punct_begin:
\__xeCJK_ulem_skip_punct_end:
3462 \cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_begin:
3463 { \xeCJK_cs_clear:N \UL@leadtype }
3464 \cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_end:
3465 { \cs_set_eq:NN \UL@leadtype \__xeCJK_ulem_leader_type: }
3466 \xeCJK_cs_clear:N \__xeCJK_ulem_leader_type:
(End definition for \__xeCJK_ulem_skip_punct_begin: and \__xeCJK_ulem_skip_punct_end:.)

```

__xeCJK_ulem_initial: 这里的设置是为了在下划线状态下,下划线可以自动跳过全角标点符号和正确的在它们前/后断行,并且与行首行末对齐。

```

3467 \cs_new_protected_nopar:Npn \__xeCJK_ulem_initial:
3468 {
3469     \__xeCJK_ulem_swap_cs:NN
3470     \xeCJK_FullLeft_and_Default: \__xeCJK_ulem_FullLeft_and_Default:
3471     \xeCJK_FullLeft_and_CJK: \__xeCJK_ulem_FullLeft_and_CJK:
3472     \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
3473     \xeCJK_FullRight_and_CJK: \__xeCJK_ulem_FullRight_and_CJK:
3474     \xeCJK_CJK_and_CJK:N \__xeCJK_ulem_CJK_and_CJK:N
3475     \xeCJK_Boundary_and_Default: \__xeCJK_ulem_Boundary_and_Default:
3476     \xeCJK@fix@penalty \__xeCJK_ulem_fix_penalty:
3477     \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
3478     \__xeCJK_CJK_and_Boundary_aux: \__xeCJK_ulem_CJK_and_Boundary_aux:
3479     \__xeCJK_Default_and_FullLeft_glue:N \__xeCJK_ulem_Default_and_FullLeft_glue:N
3480     \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N

```

```

3481 \__xeCJK_CJK_and_FullLeft_glue:N \__xeCJK_ulem_CJK_and_FullLeft_glue:N
3482 \__xeCJK_CJK_and_FullRight_glue:N \__xeCJK_ulem_CJK_and_FullRight_glue:N
3483 \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
3484 \q_recursion_tail \q_nil \q_recursion_stop
3485 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3486 {
3487   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3488   {
3489     \str_if_eq:nnTF {##1} {####1}
3490     {
3491       \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
3492       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3493       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
3494       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3495     }
3496     {
3497       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
3498       { \__xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
3499     }
3500   }
3501 }
3502 }
3503 \cs_new_protected_nopar:Npn \__xeCJK_ulem_swap_cs:NN #1#2
3504 {
3505   \quark_if_recursion_tail_stop:N #1
3506   \xeCJK_swap_cs:NN #1#2
3507   \__xeCJK_ulem_swap_cs:NN
3508 }
(End definition for \__xeCJK_ulem_initial:.)

```

`\xeCJK_if_ulem_patch:TF` 在下划线状态下, `ulem` 宏包在数学模式或者盒子中使用 `\UL@hrest` 恢复 `_` 等的定义, 此时不需要使用 `\UL@stop` 和 `\UL@start` 来断开下划线而产生断点。

```

3509 \cs_new_nopar:Npn \xeCJK_if_ulem_patch:TF
3510 {
3511   \if_meaning:w \ LA@space
3512   \exp_after:wN \use_ii:nn
3513   \else:
3514     \exp_after:wN \use_i:nn
3515   \fi:
3516 }
(End definition for \xeCJK_if_ulem_patch:TF.)

```

`_xeCJK_ulem_Boundary_and_Default:`

```

3517 \cs_new_protected_nopar:Npn \_xeCJK_ulem_Boundary_and_Default:
3518 {
3519   \xeCJK_if_ulem_patch:TF
3520   {
3521     \xeCJK_if_last_node:nTF { CJK }
3522     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
3523     { \xeCJK_if_last_node:nT { CJK-space } { \xeCJK_remove_node: \c_space_tl } }
3524   }
3525   { \_xeCJK_ulem_Boundary_and_Default: }
3526 }
(End definition for \_xeCJK_ulem_Boundary_and_Default:.)

```

`_xeCJK_ulem_CJK_and_Boundary_aux:`

```

3527 \cs_new_protected_nopar:Npn \_xeCJK_ulem_CJK_and_Boundary_aux:
3528 {
3529   \xeCJK_if_ulem_patch:TF
3530   {
3531     \UL@stop \_xeCJK_ulem_hskip:n { \c_zero_skip } \UL@start
3532     { \xeCJK_make_node:n { CJK } }

```

```

3533     }
3534     { \_xeCJK_ulem_CJK_and_Boundary_aux: }
3535   }
(End definition for \_xeCJK_ulem_CJK_and_Boundary_aux:.)

```

_xeCJK_ulem_fix_penalty:

```

3536 \cs_new_protected_nopar:Npn \_xeCJK_ulem_fix_penalty:
3537 {
3538   \xeCJK_if_ulem_patch:TF
3539   { \fix@penalty }
3540   { \_xeCJK_ulem_fix_penalty: }
3541 }
(End definition for \_xeCJK_ulem_fix_penalty:.)

```

_xeCJK_ulem_CJK_and_CJK:N

```

3542 \cs_new_protected_nopar:Npn \_xeCJK_ulem_CJK_and_CJK:N
3543 {
3544   \xeCJK_if_ulem_patch:TF
3545   {
3546     \xeCJK_class_group_end:
3547     \UL@stop \_xeCJK_ulem_ccglue: \UL@start
3548     \_xeCJK_ulem_class_group_begin:
3549     \CJKsymbol
3550   }
3551   { \_xeCJK_ulem_CJK_and_CJK:N }
3552 }
(End definition for \_xeCJK_ulem_CJK_and_CJK:N.)

```

_xeCJK_ulem_class_group_begin:

```

3553 \cs_new_protected_nopar:Npn \_xeCJK_ulem_class_group_begin:
3554 {
3555   \xeCJK_class_group_begin:
3556   \xeCJK_clear_Boundary_and_CJK_toks:
3557   \xeCJK_select_font:
3558 }
(End definition for \_xeCJK_ulem_class_group_begin:.)

```

_xeCJK_ulem_between_CJK_blocks:nnN

```

3559 \cs_new_protected_nopar:Npn \_xeCJK_ulem_between_CJK_blocks:nnN #1#2
3560 {
3561   \xeCJK_if_ulem_patch:TF
3562   {
3563     \xeCJK_class_group_end:
3564     \UL@stop \_xeCJK_ulem_ccglue: \UL@start
3565     \xeCJK_class_group_begin:
3566     \xeCJK_clear_Boundary_and_CJK_toks:
3567     \_xeCJK_switch_font:nn {#1} {#2}
3568     \CJKsymbol
3569   }
3570   {
3571     \skip_horizontal:N \l_~xeCJK_ccglue_skip
3572     \_xeCJK_switch_font:nn {#1} {#2}
3573     \CJKsymbol
3574   }
3575 }
(End definition for \_xeCJK_ulem_between_CJK_blocks:nnN.)

```

_xeCJK_ulem_Default_and_FullLeft_glue:N

```

3576 \cs_new_protected_nopar:Npn \_xeCJK_ulem_Default_and_FullLeft_glue:N #1
3577 {
3578   \xeCJK_if_ulem_patch:TF
3579   {

```

```

3580         \UL@stop
3581         \__xeCJK_ulem_skip_punct_begin:
3582         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3583         \UL@start
3584     }
3585     { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
3586 }

```

(End definition for __xeCJK_ulem_Default_and_FullLeft_glue:N.)

__xeCJK_ulem_Boundary_and_FullLeft_glue:N

```

3587 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
3588 {
3589     \xeCJK_if_ulem_patch:TF
3590     {
3591         \UL@stop
3592         \__xeCJK_ulem_skip_punct_begin:
3593         \int_compare:nNnF \etex_lastnodetype:D = \c_one
3594         { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
3595         \UL@start
3596     }
3597     { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
3598 }

```

(End definition for __xeCJK_ulem_Boundary_and_FullLeft_glue:N.)

__xeCJK_ulem_CJK_and_FullLeft_glue:N

```

3599 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
3600 {
3601     \xeCJK_if_ulem_patch:TF
3602     {
3603         \xeCJK_class_group_end:
3604         \UL@stop
3605         \__xeCJK_ulem_skip_punct_begin:
3606         \__xeCJK_ulem_ccglue:
3607         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3608         \UL@start
3609         \__xeCJK_ulem_class_group_begin:
3610     }
3611     { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
3612 }

```

(End definition for __xeCJK_ulem_CJK_and_FullLeft_glue:N.)

__xeCJK_ulem_Default_and_FullRight_glue:N

```

3613 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
3614 {
3615     \xeCJK_if_ulem_patch:TF
3616     {
3617         \UL@stop
3618         \__xeCJK_ulem_skip_punct_begin:
3619         \__xeCJK_punct_if_long:NTF {#1}
3620         { \__xeCJK_ulem_ccglue: }
3621         {
3622             \xeCJK_no_break:
3623             \__xeCJK_punct_if_middle:NT {#1}
3624             { \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1} }
3625         }
3626         \UL@start
3627     }
3628     { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
3629 }

```

(End definition for __xeCJK_ulem_Default_and_FullRight_glue:N.)

_xeCJK_ulem_CJK_and_FullRight_glue:N

```
3630 \cs_new_protected_nopar:Npn \_xeCJK_ulem_CJK_and_FullRight_glue:N #1
3631 {
3632   \xeCJK_if_ulem_patch:TF
3633   {
3634     \xeCJK_class_group_end:
3635     \_xeCJK_Default_and_FullRight_glue:N {#1}
3636     \_xeCJK_ulem_class_group_begin:
3637   }
3638   { \_xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
3639 }
```

(End definition for _xeCJK_ulem_CJK_and_FullRight_glue:N)

_xeCJK_ulem_FullLeft_and_Default:

```
3640 \cs_new_protected_nopar:Npn \_xeCJK_ulem_FullLeft_and_Default:
3641 {
3642   \xeCJK_if_ulem_patch:TF
3643   {
3644     \_xeCJK_punct_if_middle:NTF \g_xeCJK_last_punct_tl
3645     {
3646       \xeCJK_get_punct_bounds:NN \c_xeCJK_left_tl \g_xeCJK_last_punct_tl
3647       \_xeCJK_punct_rule:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
3648       \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
3649       \_xeCJK_punct_glue:NN \c_xeCJK_left_tl \g_xeCJK_last_punct_tl
3650     }
3651     { \xeCJK_class_group_end: \UL@stop }
3652     \_xeCJK_ulem_skip_punct_end:
3653     \xeCJK_no_break:
3654     \UL@start
3655   }
3656   { \_xeCJK_ulem_FullLeft_and_Default: }
3657 }
```

(End definition for _xeCJK_ulem_FullLeft_and_Default:.)

_xeCJK_ulem_FullLeft_and_CJK:

```
3658 \cs_new_protected_nopar:Npn \_xeCJK_ulem_FullLeft_and_CJK:
3659 {
3660   \xeCJK_if_ulem_patch:TF
3661   {
3662     \xeCJK_FullLeft_and_Default:
3663     \_xeCJK_ulem_class_group_begin:
3664   }
3665   { \_xeCJK_ulem_FullLeft_and_CJK: }
3666 }
```

(End definition for _xeCJK_ulem_FullLeft_and_CJK:.)

_xeCJK_ulem_FullRight_and_Default:

```
3667 \cs_new_protected_nopar:Npn \_xeCJK_ulem_FullRight_and_Default:
3668 {
3669   \xeCJK_if_ulem_patch:TF
3670   {
3671     \_xeCJK_punct_rule:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
3672     \xeCJK_class_group_end:
3673     \UL@stop
3674     \_xeCJK_punct_glue:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
3675     \_xeCJK_ulem_skip_punct_end:
3676     \UL@start
3677   }
3678   { \_xeCJK_ulem_FullRight_and_Default: }
3679 }
```

(End definition for _xeCJK_ulem_FullRight_and_Default:.)

_xeCJK_ulem_FullRight_and_CJK:

```

3680 \cs_new_protected_nopar:Npn \_xeCJK_ulem_FullRight_and_CJK:
3681 {
3682   \xeCJK_if_ulem_patch:TF
3683   {
3684     \_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3685     \xeCJK_class_group_end:
3686     \UL@stop
3687     \_xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3688     \_xeCJK_ulem_ccglue:
3689     \_xeCJK_ulem_skip_punct_end:
3690     \UL@start
3691     \_xeCJK_ulem_class_group_begin:
3692   }
3693   { \_xeCJK_ulem_FullRight_and_CJK: }
3694 }

```

(End definition for _xeCJK_ulem_FullRight_and_CJK:.)

_xeCJK_ulem_punct_hskip:n

```

3695 \cs_new_protected_nopar:Npn \_xeCJK_ulem_punct_hskip:n
3696 {
3697   \xeCJK_if_ulem_patch:TF
3698   { \_xeCJK_ulem_hskip:n }
3699   { \_xeCJK_ulem_punct_hskip:n }
3700 }

```

(End definition for _xeCJK_ulem_punct_hskip:n.)

_xeCJK_ulem_glue:n

在下划线状态下的分别代替 \CJKglue 等。

_xeCJK_ulem_ccglue:

_xeCJK_ulem_hskip:n

```

3701 \cs_new_protected_nopar:Npn \_xeCJK_ulem_glue:n #1
3702 {
3703   \xeCJK_if_ulem_patch:TF
3704   { \UL@stop \_xeCJK_ulem_hskip:n {#1} \UL@start }
3705   { \skip_horizontal:n {#1} }
3706 }
3707 \cs_new_protected_nopar:Npn \_xeCJK_ulem_ccglue:
3708 { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders }
3709 \cs_new_protected_nopar:Npn \_xeCJK_ulem_hskip:n #1
3710 {
3711   \int_compare:nNnTF \tex_lastkern:D = \c_three
3712   { \skip_horizontal:n {#1} }
3713   { \skip_set:Nn \UL@skip {#1} \UL@leaders }
3714 }

```

(End definition for _xeCJK_ulem_glue:n, _xeCJK_ulem_ccglue:, and _xeCJK_ulem_hskip:n.)

\CJKunderdot

使用 xeCJK 时, CJKfntef 中的 \CJKunderdot 和 \CJKunderanysymbol 在汉字之间不能断行。因此需要我们在这一处修改它们。

```

3715 \RenewDocumentCommand \CJKunderdot { m }
3716 {
3717   \bool_if:NT \l__xeCJK_ulem_hook_used_bool
3718   { \UL@stop \_xeCJK_ulem_restore_CJK_and_Boundary: }
3719   \CJK@preUnderdot
3720   \_xeCJK_make_under_symbol:n { \CJK@underdotSkip }
3721   \cs_gset_eq:NN \_xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3722   \cs_set_eq:NN \CJKsymbol \_xeCJK_under_CJKsymbol:N
3723   \_xeCJK_restore_output_CJKsymbol:
3724   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3725   #1
3726   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3727   \cs_set_eq:NN \CJKsymbol \_xeCJK_save_under_dot_CJKsymbol:N
3728   \tex_output:D \exp_after:wN { \l__xeCJK_underdot_output_tl }
3729   \CJK@postUnderdot
3730   \bool_if:NT \l__xeCJK_ulem_hook_used_bool

```

```

3731     { \_xeCJK_ulem_restore_CJK_and_Boundary: \UL@start }
3732     \tex_ignorespaces:D
3733   }
3734   \box_new:N \g__xeCJK_under_symbol_box
3735   \cs_new_protected_nopar:Npn \_xeCJK_ulem_restore_CJK_and_Boundary:
3736   {
3737     \xeCJK_if_ulem_patch:TF
3738     {
3739       \xeCJK_swap_cs:NN
3740       \_xeCJK_CJK_and_Boundary_aux: \_xeCJK_ulem_CJK_and_Boundary_aux:
3741     }
3742     { }
3743   }
(End definition for \CJKunderdot.)

```

\CJKunderanysymbol

```

3744 \RenewDocumentCommand \CJKunderanysymbol { m m m }
3745 {
3746   \group_begin:
3747   \hbox_set:Nn \CJK@underdotBox {#2}
3748   \_xeCJK_make_under_symbol:n {#1}
3749   \cs_gset_eq:NN \_xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3750   \cs_set_eq:NN \CJKsymbol \_xeCJK_under_CJKsymbol:N
3751   \_xeCJK_restore_output_CJKsymbol:
3752   #3
3753   \group_end:
3754   \tex_ignorespaces:D
3755 }
(End definition for \CJKunderanysymbol.)

```

_xeCJK_restore_output_CJKsymbol: \CJKunderdot 中对 \CJKsymbol 的修改会影响到页眉和页脚,需要小心处理。

```

3756 \cs_new_protected:Npn \_xeCJK_restore_output_CJKsymbol:
3757 {
3758   \tl_set:Nx \l__xeCJK_underdot_output_tl
3759   { \exp_after:wN \exp_not:n \tex_the:D \tex_output:D }
3760   \tex_output:D \exp_after:wN
3761   {
3762     \exp_after:wN \cs_set_eq:NN
3763     \exp_after:wN \CJKsymbol
3764     \exp_after:wN \_xeCJK_save_under_dot_CJKsymbol:N
3765     \l__xeCJK_underdot_output_tl
3766   }
3767 }
3768 \tl_new:N \l__xeCJK_underdot_output_tl
(End definition for \_xeCJK_restore_output_CJKsymbol:.)

```

_xeCJK_make_under_symbol:n

```

3769 \cs_new_protected:Npn \_xeCJK_make_under_symbol:n #1
3770 {
3771   \hbox_set:Nn \l__xeCJK_tmp_box { ^^^^4e00 }
3772   \vbox_gset_to_ht:Nnn \g__xeCJK_under_symbol_box \c_zero_dim
3773   {
3774     \skip_vertical:n {#1}
3775     \hbox_to_zero:n
3776     {
3777       \tex_kern:D - \box_wd:N \l__xeCJK_tmp_box
3778       \tex_hss:D \box_use:N \CJK@underdotBox \tex_hss:D
3779     }
3780     \tex_vss:D
3781   }
3782 }
(End definition for \_xeCJK_make_under_symbol:n.)

```

_xeCJK_under_CJKsymbol:N

```
3783 \cs_new_protected_nopar:Npn \_xeCJK_under_CJKsymbol:N #1
3784 {
3785   \_xeCJK_save_under_dot_CJKsymbol:N {#1}
3786   \hbox_overlap_left:n { \box_use:N \g_xeCJK_under_symbol_box }
3787   { \xeCJK_make_node:n { CJK } }
3788   \xeCJK_ignore_spaces:w
3789 }
(End definition for \_xeCJK_under_CJKsymbol:N)
```

CJKfilltwosides 使用 minipage 和 L^AT_EX 表格(tabular)来定义 CJKfilltwosides 环境。可选参数 #1 表示环境的垂直对齐位置,默认居中;参数 #2 表示环境的宽度。带星号的环境,如果 #2 不大于零或者不大于环境最长文本行的宽度,则取环境的自然宽度。

```
3790 \RenewDocumentEnvironment { CJKfilltwosides } { 0 { c } m }
3791 {
3792   \use:x { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
3793   \cs_set_eq:NN \CJKglue \tex_hfill:D
3794 }
3795 {
3796   \endminipage
3797   \ignorespacesafterend
3798 }
3799 \NewEnviron { CJKfilltwosides* } [ 2 ] [ c ]
3800 {
3801   \cs_set_eq:NN \CJKglue \tex_hfill:D
3802   \tl_set:Nn \arraystretch { 1 }
3803   \token_if_dim_register:NT \extrarowheight
3804   { \dim_set_eq:NN \extrarowheight \c_zero_dim }
3805   \dim_compare:nNnTF {#2} > \c_zero_dim
3806   {
3807     \hbox_set:Nn \l_xeCJK_tmp_box
3808     {
3809       \tabular [#1] { @ { } c @ { } }
3810       \BODY
3811       \endtabular
3812     }
3813     \dim_compare:nNnTF {#2} > { \box_wd:N \l_xeCJK_tmp_box }
3814     {
3815       \tabular [#1] { @ { } p { \dim_eval:n {#2} } @ { } }
3816       \BODY
3817       \endtabular
3818     }
3819     { \hbox_unpack:N \l_xeCJK_tmp_box }
3820   }
3821   {
3822     \tabular [#1] { @ { } c @ { } }
3823     \BODY
3824     \endtabular
3825   }
3826 }
3827 [ \ignorespacesafterend ]
(End definition for CJKfilltwosides.)
3828 \fntef
```

5.20 xeCJK-listings

仿照 luatexja 宏包中 lltjp-listings 的处理,支持 listings 宏包。

```
3829 (*listings)
3830 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
3831 \ProcessOptions \scan_stop:
```

```

3832 \RequirePackage { xeCJK }
3833 \RequirePackage { listings }

3834 \lst@AddToHook { Init } { \_xeCJK_listings_initial_hook: }
3835 \lst@AddToHook { SelectCharTable } { \_xeCJK_listings_toks_hook: }
3836 \lst@AddToHook { OutputBox }
3837 {
3838     \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
3839     \l__xeCJK_restore_listings_toks_tl
3840     \_xeCJK_listings_output_IVS:
3841 }
3842 \lst@AddToHook { PreSet } { \bool_set_true:N \l__xeCJK_listings_env_bool }

```

`_xeCJK_listings_initial_hook:` 为使代码行号结果正确,需要在 `\lst@numberstyle` 中恢复 `\XeTeXinterchartoks`。在 `listings` 环境中换页时,对 `\XeTeXinterchartoks` 的修改会影响到页眉和页脚,需要在 `\output` 中恢复成正常定义。这里使用 `\use:n` 是为了在 `\tex_output:D` 中不增加额外的分组。加入 `\tex_noindent:D` 是为了防止汉字出现在首行的时候可能会产生额外空行。

```

3843 \cs_new_protected_nopar:Npn \_xeCJK_listings_initial_hook:
3844 {
3845     \tex_noindent:D
3846     \bool_gset_false:N \g__xeCJK_listings_IVS_bool
3847     \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
3848     \tex_output:D \exp_after:wN \exp_after:wN \exp_after:wN
3849     {
3850         \exp_after:wN \exp_after:wN
3851         \exp_after:wN \l__xeCJK_restore_listings_toks_tl
3852         \exp_after:wN \use:n
3853         \tex_the:D \tex_output:D
3854     }
3855     \lst@ifbreaklines
3856     \cs_set_eq:NN \_xeCJK_listings_CJK_toks: \_xeCJK_listings_breaklines_toks:
3857     \fi:
3858 }

```

(End definition for `_xeCJK_listings_initial_hook:.`)

`_xeCJK_listings_toks_hook:` 采用不同的 `\XeTeXinterchartoks` 处理方式,输入的时候是将汉字加入到 `listings` 的输出队列,实际输出的时候是普通文字。

```

3859 \cs_new_protected_nopar:Npn \_xeCJK_listings_toks_hook:
3860 {
3861     \tl_set:Nx \l__xeCJK_restore_listings_toks_tl
3862     {
3863         \_xeCJK_backup_inter_class_toks:nn { Boundary } { Default }
3864         \_xeCJK_backup_inter_class_toks:nn { Boundary } { CJK }
3865         \_xeCJK_backup_inter_class_toks:nn { Boundary } { IVS }
3866         \_xeCJK_backup_inter_class_toks:nn { Boundary } { FullLeft }
3867         \_xeCJK_backup_inter_class_toks:nn { Boundary } { FullRight }
3868     }
3869     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3870     {
3871         \tl_put_right:Nx \l__xeCJK_restore_listings_toks_tl
3872         { \_xeCJK_backup_inter_class_toks:nn { Boundary } { CJK/##1 } }
3873     }
3874     \xeCJK_inter_class_toks:nnn { Boundary } { Default }
3875     { \_xeCJK_listings_process_Default:N }
3876     \xeCJK_inter_class_toks:nnn { Boundary } { IVS }
3877     { \_xeCJK_listings_process_IVS:nN { \c_zero } }
3878     \_xeCJK_listings_CJK_toks_hook:
3879 }
3880 \tl_new:N \l__xeCJK_restore_listings_toks_tl
3881 \cs_new_nopar:Npn \_xeCJK_backup_inter_class_toks:nn #1#2
3882 {
3883     \xeCJK_inter_class_toks:nnn {#1} {#2}
3884     { \xeCJK_get_inter_class_toks:nn {#1} {#2} }

```

```
3885 }
(End definition for \_xeCJK_listings_toks_hook:.)
```

_xeCJK_listings_CJK_toks_hook: 根据 **breaklines** 选项的使用与否,选择不同的处理方式。

```
\_xeCJK_listings_breaklines_toks: 3886 \cs_new_protected_nopar:Npn \_xeCJK_listings_CJK_toks_hook:
3887 {
3888   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
3889   { \_xeCJK_listings_process_CJK:nN { \c_two } }
3890   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
3891   { \_xeCJK_listings_process_CJK:nN { \c_two } }
3892   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
3893   { \_xeCJK_listings_process_CJK:nN { \c_two } }
3894   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3895   {
3896     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
3897     { \_xeCJK_listings_process_CJK:nN { \c_two } }
3898   }
3899 }
3900 \cs_new_protected_nopar:Npn \_xeCJK_listings_breaklines_toks:
3901 {
3902   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
3903   { \_xeCJK_listings_process_breaklines_CJK:nN { \c_two } }
3904   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
3905   { \_xeCJK_listings_process_FullLeft:nN { \c_two } }
3906   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
3907   { \_xeCJK_listings_process_FullRight:nN { \c_two } }
3908   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3909   {
3910     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
3911     { \_xeCJK_listings_process_breaklines_CJK:nN { \c_two } }
3912   }
3913 }
```

(End definition for _xeCJK_listings_CJK_toks_hook: and _xeCJK_listings_breaklines_toks:.)

_xeCJK_listings_process_Default:N 对于 \charcode 大于 255 的字符,根据 \catcode 进行处理。

```
\_xeCJK_listings_process_CJK:nN 3914 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_Default:N #1
3915 {
3916   \token_if_letter:NTF #1
3917   { \lst@ProcessLetter #1 }
3918   { \lst@ProcessOther #1 }
3919 }
3920 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_CJK:nN #1#2
3921 {
3922   \token_if_letter:NTF #2
3923   { \_xeCJK_listings_process_letter:nN {#1} #2 }
3924   { \_xeCJK_listings_process_other:nN {#1} #2 }
3925 }
```

(End definition for _xeCJK_listings_process_Default:N and _xeCJK_listings_process_CJK:nN.)

_xeCJK_listings_append:nN 普通 CJK 字符的宽度为一般基本宽度的两倍,IVS 类不增加宽度。这里有一个问题,对 CJK 字符类中的一些半角字符(例如半角日文假名)没有区分开。listings 通过重定义 \lst@Append 将代码写入外部文件,因此需要保留。

```
3926 \cs_new_protected_nopar:Npn \_xeCJK_listings_append:nN #1#2
3927 {
3928   \int_add:Nn \lst@length { #1 - \c_one }
3929   \lst@Append #2
3930 }
```

(End definition for _xeCJK_listings_append:nN.)

_xeCJK_listings_process_letter:nN 在 **letter** 类中区分汉字和西文字母。

```
\_xeCJK_listings_process_other:nN 3931 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_letter:nN
3932 {
```

```

3933 \lst@whitespacefalse
3934 \bool_if:NTF \l__xeCJK_listings_letter_bool
3935 { \lst@lettertrue }
3936 {
3937 \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
3938 \bool_set_true:N \l__xeCJK_listings_letter_bool
3939 }
3940 \__xeCJK_listings_append:nN
3941 }
3942 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_other:nN #1#2
3943 {
3944 \lst@whitespacefalse
3945 \bool_if:NTF \l__xeCJK_listings_letter_bool
3946 {
3947 \lst@Output \lst@letterfalse
3948 \bool_set_false:N \l__xeCJK_listings_letter_bool
3949 }
3950 { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
3951 \cs_set_eq:NN \lst@lastother #2
3952 \__xeCJK_listings_append:nN {#1} #2
3953 }

```

(End definition for __xeCJK_listings_process_letter:nN and __xeCJK_listings_process_other:nN.)

当使用 **breaklines** 选项时, 立即输出之前的单个文字, 以便于断行。并将标点与它前/后的 CJK 文字放在同一个盒子中, 以保持禁则。但是不能区分 letter 和 other。

```

\__xeCJK_listings_process_breaklines_CJK:nN
\__xeCJK_listings_process_FullLeft:nN
\__xeCJK_listings_process_FullRight:nN
3954 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_breaklines_CJK:nN
3955 {
3956 \lst@whitespacefalse
3957 \bool_if:NTF \l__xeCJK_listings_letter_bool
3958 {
3959 \int_compare:nNnF \l__xeCJK_listings_flag_int = \c_two { \lst@Output }
3960 \lst@lettertrue
3961 }
3962 {
3963 \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
3964 \bool_set_true:N \l__xeCJK_listings_letter_bool
3965 }
3966 \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one
3967 \__xeCJK_listings_append:nN
3968 }
3969 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_FullLeft:nN #1#2
3970 {
3971 \lst@whitespacefalse
3972 \bool_if:NTF \l__xeCJK_listings_letter_bool
3973 {
3974 \bool_if:nF
3975 {
3976 \int_compare_p:nNn \l__xeCJK_listings_flag_int = \c_two ||
3977 ( \int_compare_p:nNn \l__xeCJK_listings_flag_int = \c_three &&
3978 ! \l__xeCJK_punct_breakable_bool )
3979 }
3980 { \lst@Output }
3981 \lst@lettertrue
3982 }
3983 {
3984 \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
3985 \bool_set_true:N \l__xeCJK_listings_letter_bool
3986 }
3987 \int_set_eq:NN \l__xeCJK_listings_flag_int \c_two
3988 \__xeCJK_listings_append:nN {#1} #2
3989 }
3990 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_FullRight:nN #1#2

```

```

3991 {
3992   \lst@whitespacefalse
3993   \bool_if:NTF \l__xeCJK_listings_letter_bool
3994   {
3995     \bool_if:nT
3996     {
3997       \int_compare_p:nNn \l__xeCJK_listings_flag_int < \c_two &&
3998       \__xeCJK_punct_if_long_p:N #2
3999     }
4000     { \lst@Output }
4001     \lst@lettertrue
4002   }
4003   {
4004     \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
4005     \bool_set_true:N \l__xeCJK_listings_letter_bool
4006   }
4007   \int_set_eq:NN \l__xeCJK_listings_flag_int \c_three
4008   \__xeCJK_listings_append:nN {#1} #2
4009 }
4010 \int_new:N \l__xeCJK_listings_flag_int
(End definition for \__xeCJK_listings_process_breaklines_CJK:nN, \__xeCJK_listings_process_FullLeft:nN, and
\__xeCJK_listings_process_FullRight:nN.)

```

\lst@AppendLetter

\lst@AppendOther

```

4011 \cs_set_protected_nopar:Npn \lst@AppendLetter
4012 {
4013   \bool_if:NTF \l__xeCJK_listings_letter_bool
4014   {
4015     \lst@Output \lst@lettertrue
4016     \bool_set_false:N \l__xeCJK_listings_letter_bool
4017   }
4018   { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
4019   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
4020   \lst@Append
4021 }
4022 \cs_set_protected_nopar:Npn \lst@AppendOther
4023 {
4024   \bool_if:NTF \l__xeCJK_listings_letter_bool
4025   {
4026     \lst@Output \lst@letterfalse
4027     \bool_set_false:N \l__xeCJK_listings_letter_bool
4028   }
4029   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
4030   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
4031   \tex_futurelet:D \lst@lastother \lst@Append
4032 }
(End definition for \lst@AppendLetter and \lst@AppendOther.)

```

__xeCJK_listings_process_IVS:nN IVS 类作为 letter 处理,不用增加 \lst@length。

```

4033 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_IVS:nN
4034 {
4035   \reverse_if:N \lst@iflexible
4036   \bool_gset_true:N \g__xeCJK_listings_IVS_bool
4037   \fi:
4038   \__xeCJK_listings_process_letter:nN
4039 }
(End definition for \__xeCJK_listings_process_IVS:nN.)

```

__xeCJK_listings_output_IVS: 在使用 columns=fixed 选项时, listings 会在输出盒子中的每个字符之间加入 \hss,这就破坏了 X_YTeX 将基本字和 IVS 正确的组合起来。

```

4040 \cs_new_protected_nopar:Npn \__xeCJK_listings_output_IVS:
4041 {

```

```

4042 \reverse_if:N \lst@iflexible
4043 \bool_if:NT \g__xeCJK_listings_IVS_bool
4044 {
4045     \bool_gset_false:N \g__xeCJK_listings_IVS_bool
4046     \xeCJK_cs_clear:N \lst@FillOutputBox
4047     \cs_set_eq:NN \CJKglue \tex_hss:D
4048 }
4049 \fi:
4050 }
4051 \bool_new:N \g__xeCJK_listings_IVS_bool
(End definition for \__xeCJK_listings_output_IVS:.)

```

_xeCJK_listings_peek_active_loop:TF \lstinline 通过判断参数中第一个字符是否是 **active** 类来区分它是否被用在其它宏的参数之中。如果这第一个字符不在 **listings** 预定义的符号表中,判断就会出问题。我们在这里通过一个循环跳过这些字符。

```

4052 \cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
4053 {
4054     \token_if_active:NTF #3
4055     { #1#3 }
4056     {
4057         \token_if_cs:NTF #3
4058         { #2#3 }
4059         {
4060             \int_compare:nNnTF { `#3 } > { \lst@ifec 255 \else: 127 \fi: }
4061             { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
4062             { #2#3 }
4063         }
4064     }
4065 }
4066 \cs_set_eq:NN \lst@ifNextCharActive \__xeCJK_listings_peek_active_loop:TF
(End definition for \__xeCJK_listings_peek_active_loop:TF.)

```

_xeCJK_listings_inside_convert:nw 当 \lstinline 被使用在参数中时,listings 会使用一个循环逐个将 \lstinline 参数中的字符设置为活动字符。我们可以通过 \tl_set_rescan:Nnn 来完成这里的 \catcode 转换,避免将 \charcode 超过 255 的字符都设置为活动字符。

```

4067 \cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
4068 {
4069     \tl_set_rescan:Nnn \l__xeCJK_tmp_tl { } {#1}
4070     \__xeCJK_set_listings_escape:
4071     \tl_put_right:NV \lst@arg \l__xeCJK_tmp_tl
4072 }
4073 \cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
4074 \cs_new_protected_nopar:Npn \__xeCJK_listings_inline_group:w
4075 {
4076     \exp_after:wN \__xeCJK_listings_inline_group:n
4077     \exp_after:wN { \if_false: } \fi:
4078 }
4079 \cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
4080 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
4081 {
4082     \tl_set_rescan:Nnn \lst@arg { } {#1}
4083     \__xeCJK_set_listings_escape:
4084     \lst@InlineGJEnd
4085 }
(End definition for \__xeCJK_listings_inside_convert:nw and \__xeCJK_listings_inline_group:w)

```

_xeCJK_set_listings_escape: 由于我们在上面的修改,需要保留 \ 用于转义 \lstinline 参数中的某些 TeX 特殊字符,与原来宏包一致。

```

4086 \group_begin:
4087 \cs_set:Npn \__xeCJK_tmp:w #1
4088 {
4089     \group_end:

```



```

4090     \cs_new_protected:Npn \__xeCJK_set_listings_escape:
4091     { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
4092     \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
4093     { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
4094 }
4095 \use:n
4096 {
4097     \char_set_catcode_active:N \
4098     \__xeCJK_tmp:w
4099 }
4100 { \ }
(End definition for \__xeCJK_set_listings_escape:.)
4101 </listings>

```

5.21 xeCJK.cfg

```

4102 <*config>

```

预设的配置文件 `xeCJK.cfg` 为一个空文件。可以在里面增加设置,然后保存到本地目录下面。

```

4103
4104 </config>

```

版本历史

v3.1.0

General: 使用 <code>xtemplate</code> 宏包的机制来组织标点符号的处理。.....	46
删除多余的 <code>default-itcorr</code> 结点。.....	30
取消 <code>\cprotect</code> 的外部宏限制。.....	82
改用 <code>indentfirst</code> 宏包处理缩进的问题。.....	76
放弃使用放缩字体大小的方式,而只采用调整间距的方式与西文等宽字体对齐。并且只适用于与抄录环境下。..	71
放弃对 <code>\outer</code> 宏的特殊处理。.....	1
<code>_xeCJK_switch_font:n</code> : 改进定义,加快切换速度。..	64
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑 <code>\spaceskip</code> 不为零的情况。.....	15
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。.....	74
<code>\xeCJK@fix@penalty</code> : 采用通过不修改原语 <code>\/</code> 的方式对修复倾斜校正。.....	79
<code>\xeCJK_fallback_loop:N</code> : 调整备用字体的循环方式。..	54
<code>\xeCJK_glyph_if_exist:N</code> : 改进 <code>fontspec</code> 宏包中定义的 <code>\font_glyph_if_exist:NnTF</code> 。.....	15
<code>\xeCJK_hook_for_ulem::</code> : 简化对 <code>ulem</code> 宏包的兼容补丁。..	83
<code>\xeCJK_visible_space_fallback::</code> : 调整 <code>fontspec</code> 的后备可视空格符号,以便于使用时对齐。.....	74
<code>\xeCJKOffVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。.....	71

v3.1.1

General: 不再依赖 <code>xpatch</code> 宏包。.....	1
增加 <code>NewLineCS</code> 和 <code>EnvCS</code> 选项。.....	41
增加小宏包 <code>xeCJKfntef</code> ,用于处理下划线的问题。.....	83
对于与 <code>xltxtra</code> 的冲突给出错误警告。.....	79
<code>_xeCJK_check_single_space:NN</code> : <code>CheckSingle</code> 支持段末“汉字+汉字+空格+汉字/标点”的形式。.....	40
<code>_xeCJK_set_char_class_eq:n</code> : 交换参数的顺序。..	25
<code>_xeCJK_set_verb_exspace::</code> : 调整间距的计算方法。..	72
<code>\fontfamily</code> : 修改主要 CJK 字体族的自动更新方式。....	79
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。....	37
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。.....	42
<code>InlineEnv</code> : 改变行内环境的设置方式,从而使用 <code>\str_case:x:nnn</code> 代替原来的 <code>\clist_if_in:NnTF</code> 来判断是否是行内环境。.....	41
<code>\xeCJK_check_single:NNw</code> : 改进定义,减少使用 <code>peek</code> 函数的次数。.....	39
<code>\xeCJK_hook_for_ulem::</code> : 完全处理下划线里的标点符号的有关问题。.....	83
<code>\xeCJK_peek_catcode_ignore_spaces:N</code> : 新增有省略空格标识的 <code>peek</code> 函数。.....	17
<code>\xeCJK_save_class:n</code> : 使用 <code>\xeCJK_save_class:n</code> 保存 <code>X_gTeX</code> 预定义的字符类别。.....	19
<code>\xeCJK_set_char_class:nnn</code> : 在文档中设置字符类别时不重复设置 <code>\catcode</code> 。.....	25
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。.....	38

v3.1.2

General: 修正重定义 <code>\CJKfamilydefault</code> 无效的问题,恢复容错能力。.....	69
解决在下划线状态下使用 <code>\makebox</code> 时的错误。.....	85
<code>_xeCJK_check_single_space:NN</code> : 使用 <code>\xeCJK_if_CJK_class:N</code> 来代替 <code>\int_case:nnn</code> 判断是否是 CJK 字符类。.....	40
<code>_xeCJK_family_unknown_warning:n</code> : 在没有定义任何 CJK 字体的情况下,不再重复给出字体没有定义的警告。..	65
<code>\fontfamily</code> : 不将参数完全展开。.....	79

<code>\nobreakspace</code> : 修正非 <code>\UTFencname</code> 编码下面 <code>xunicode</code> 重定义的 <code>\nobreakspace</code> 会失效的问题。.....	78
---	----

v3.2.0

General: 增加 <code>IVS</code> 字符类用于处理异体字选择符。.....	19
增加 <code>Verb</code> 选项。.....	71
<code>_xeCJK_Boundary_and_FullLeft_glue:N</code> : 当全角左标点前面是 <code>hlist</code> 、 <code>none</code> 、 <code>glue</code> 和 <code>penalty</code> 等节点时,压缩其左空白。.....	34
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑到 <code>\spacefactor</code> 和 <code>\xspaceskip</code> 的情况。.....	15
<code>\l_xeCJK_family_tl</code> : 不将其初始化为 <code>\CJKfamilydefault</code> 。.....	65
<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。.....	66
<code>\xeCJK_FullLeft_and_Default::</code> : 修正 <code>xeCJK</code> 使西文在部分情况下无法断词的问题。.....	33

v3.2.1

General: 调整 <code>Verb</code> 选项: 在命令 <code>\verb</code> 里使用时,不破坏标点禁则,增加值 <code>env+</code> 。.....	71
--	----

v3.2.2

General: 增加小宏包 <code>xeCJK-listings</code> ,用于支持 <code>listings</code> 宏包。.....	91
<code>_xeCJK_ulem_CJK_and_FullRight_glue:N</code> : 修正下划线不能跳过全角右标点的问题。.....	88
<code>\xeCJK@text@composite</code> : 修正某些重音不能正确显示的问题。.....	78

v3.2.3

General: 不再改变 CJK 字符类的 <code>\catcode</code> 。.....	25
完善对 <code>listings</code> 宏包的支持。.....	91
提供四个 <code>TECkit</code> 映射文件用于句号转换和简繁互换。..	1
根据 <code>X_gTeX</code> 的脚本重新整理全角标点符号。.....	20
解决 <code>CheckSingle</code> 选项与 <code>tablists</code> 宏包的冲突。.....	40
<code>_xeCJK_listings_initial_hook::</code> : 解决 <code>listings</code> 环境中代码行号输出不正确的问题,并解决在其中跨页时对页眉和页脚的影响。.....	92
<code>_xeCJK_listings_process_CJK:n</code> : 在 <code>listings</code> 环境中对 <code>\charcode</code> 大于 255 的字符根据其 <code>\catcode</code> 区分 <code>letter</code> 和 <code>other</code> 。.....	93
<code>_xeCJK_restore_output_CJKsymbol::</code> : 解决 <code>\CJKunderdot</code> 跨页使用时影响到页眉页脚的问题。.....	90
<code>_xeCJK_ulem_FullLeft_and_CJK::</code> : 修正全角左标点后下划线与 <code>\CJKunderdot</code> 连用时结果不正常的问题。....	88
<code>\xeCJKOffVerbAddon</code> : 新增 <code>\xeCJKOffVerbAddon</code> 用于局部取消 <code>\xeCJKOffVerbAddon</code> 的影响;并解决跨页使用时影响到页眉页脚的问题。.....	71

v3.2.4

General: 不再使用 <code>CJKnumber</code> 选项,可以在 <code>xeCJK</code> 之后直接使用 <code>CJKnumb</code> 宏包得到中文数字。.....	83
使 <code>listings</code> 的 <code>breaklines</code> 选项对 CJK 字符类可用,并保持标点符号的禁则。.....	94
使用 <code>AllowBreakBetweenPuncts</code> 时,相应标点符号仍能与边界对齐。.....	36
修正 <code>xeCJKfntef</code> 与 <code>natbib</code> 等的冲突。.....	83
内部调整分区字体的设置方法。.....	56
尽量移除用作判断标志的 <code>\kern</code> 。.....	30
改进获取分区字体属性的办法。.....	56
解决使用 <code>CheckSingle</code> 时,某些 <code>\CJKglue</code> 不能被正确加入的问题。.....	40
遵循 <code>L^AT_EX3</code> 变量需要预先声明的原则。.....	1
<code>_xeCJK_Boundary_and_FullLeft_glue:N</code> : 细化边界与全角左标点之间是否压缩空白的判断。.....	34

_xeCJK_set_verb_exspace:: 当计算得出的间距为负时, 缩小 CJK 字体。	72	\xeCJK_fallback_loop:Nn: 使 \CJKfamilydefault 的 FallBack 设置全局可用。	54
\addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。	67	\xeCJK_tl_remove_outer_braces:n: 去掉外层分组括号时, 移除空格, 避免死循环。	14
CJKfilltwosides: 改用 minipage 和 L ^A T _E X 表格(tabular)来实现。	91	\xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。	18

代码索引

斜体的数字表示对应项说明所在的页码, 下划线的数字表示定义所在的代码行号, 而直立体的数字表示对应项使用时所在的行号。

Symbols

\@italiccorr	3221, 3229, 3235, 3241, 3244, 3255	_xeCJK_check_single_space:NN	1162, 1171, <u>1203</u> , 1203
\@empty	3157, 3161, 3164, 4067	_xeCJK_class_csname:n	302, 305, 308, 313, 316, <u>321</u> , 321, 322, 442, 445, 1324, 1342, 3272
\@ifpackagelater	36, 39, 3177, 3193	_xeCJK_clear_Boundary_and_CJK_toks:n	553, 554
\@ifpackageloaded	19	_xeCJK_copy_family:nn	<u>2390</u> , 2390, 2409
\@onefilewithoptions	3407	_xeCJK_copy_family:xx	2452
\@onlypreamble	1300, 1895, 1902, 2579, 2580, 2581, 2582, 2583, 2620	_xeCJK_copy_sub_family:n	2352, 2359
\@pkgextension	64	_xeCJK_dim_max:nn	3187, 3195, 3199
\@text@composite	3155, 3158, 3164, 3167, 3168	_xeCJK_dim_min:nn	3188, 3196, 3200
\[.	3082	_xeCJK_document_left_hook:	82, 88
\\	5, 6, 7, 13, 30, 31, 32, 33, 325, 326, 1336, 1388, 1389, 1878, 1879, 1880, 1892, 1893, 1987, 2539, 2672, 2754, 2755, 2756, 2757, 2762, 2963, 2965, 3059, 3103, 3104, 3171, 3172, 4097	_xeCJK_document_right_hook:	84, 90
_xeCJK_Boundary_and_FullLeft_glue:N	900, <u>908</u> , 908, 3483	_xeCJK_error:n	<u>49</u> , 50, 1394
_xeCJK_CJK_and_Boundary_aux:	786, 789, 3478, 3740	_xeCJK_error:nx <u>49</u> , 51, 303, 314, 1330, 1871, 1900, 3053, 3179	
_xeCJK_CJK_and_FullLeft_glue:N	890, 895, 3481	_xeCJK_fallback_save_CJKsymbol:N	1934, 1941, 1949, 1964, 1975
_xeCJK_CJK_and_FullRight_glue:N	965, <u>971</u> , 971, 981, 3482	_xeCJK_family_csname:n	2203, 2376, 2455, <u>2463</u> , 2463, 2481
_xeCJK_CJK_class_tl:c	442	_xeCJK_family_default_wrap:n	2711, 2725, 3111, 3114
_xeCJK_CJK_class_tl:n	276, 279, 281	_xeCJK_family_nfss_csname:n	2220, 2405, 2406, <u>2463</u> , 2464, 2465, 2469, 2479
_xeCJK_Default_and_FullLeft_glue:N	876, 885, 3479	_xeCJK_family_unknown_warning:n	2524, 2524, 2535
_xeCJK_Default_and_FullRight_glue:N	940, 953, <u>971</u> , 981, 3480, 3635	_xeCJK_family_unknown_warning:x	2500, 2512, 2522
_xeCJK_add_fake_bold:n	<u>2233</u> , 2233, 2282, 2301	_xeCJK_family_use:x	2418, 2427, 2459, <u>2463</u> , 2465, 2498, 2521
_xeCJK_add_fake_slant:n	<u>2233</u> , 2238, 2293	_xeCJK_font_csname:n	2411, 2412, 2424, 2426, 2446, 2449
_xeCJK_add_font:nV	2259, 2261	_xeCJK_fontspec:nnn	2608, 2615
_xeCJK_add_font:nn	<u>2227</u> , 2227, 2231	_xeCJK_fontspec:xnn	2603
_xeCJK_add_font:nv	2249	_xeCJK_get_map_font:n	2309, 2311, <u>2314</u> , 2314, 2318
_xeCJK_add_font:nx	2309	_xeCJK_get_sub_features:Vn	2001
_xeCJK_add_font_if_new:nn	<u>2227</u> , 2229, 2232, 2303	_xeCJK_get_sub_features:nn	2055, <u>2061</u> , 2061, 2091
_xeCJK_add_font_if_new:nv	2253	_xeCJK_get_sub_features:w	<u>2061</u> , 2065, 2076
_xeCJK_add_font_if_new:nx	2311	_xeCJK_gobble_CJKfamily:	2515, 2515, 3332
_xeCJK_add_special_punct:nn	1419, 1422, 1425, 1447	_xeCJK_gobble_CJKfamily:wn	2516, 2517
_xeCJK_add_sub_class_features:n	2642, 2649, <u>2675</u> , 2675	_xeCJK_group_begin:	<u>3295</u> , 3295, 3306, 3334
_xeCJK_after_end_preamble:n	<u>65</u> , 72, 1395, 3258, 3326, 3339	_xeCJK_group_end:	<u>3295</u> , 3296, 3306, 3335
_xeCJK_after_preamble:n	<u>65</u> , 70, 2817, 3297	_xeCJK_gset_family_cs:x	2168, <u>2201</u> , 2201
_xeCJK_at_end_preamble:n	<u>65</u> , 68, 2709, 3189, 3361, 3374	_xeCJK_gset_family_nfss_cs:xx	2210, 2383, <u>2463</u> , 2466
_xeCJK_aux_r:n	3309, 3317, 3318	_xeCJK_gset_punct_dim:nnnx	1492, 1594, 1674, 1794, 1795, 1796, 1801, 1811, 1813, 1826, 1849
_xeCJK_backup_inter_class_toks:nn	3863, 3864, 3865, 3866, 3867, 3872, 3881	_xeCJK_gset_punct_dim:nnx	1490, 1820, 1822
_xeCJK_block_select_font:n	2440, <u>2444</u> , 2444	_xeCJK_if_font_select:n	2267
_xeCJK_calc_kerning_margin:NN	1686, <u>1706</u> , 1706	_xeCJK_if_font_select:nF	2252, 2258, 2260, 2310
_xeCJK_calc_kerning_margin:nNN	1719, 1728, <u>1773</u> , 1773	_xeCJK_if_font_select:nTF	2247, <u>2267</u>
_xeCJK_check_family:V	2167, 2361	_xeCJK_if_font_select:p:n	<u>2267</u> , 2289, 2291, 2299
_xeCJK_check_family:n	<u>2215</u> , 2215, 2224	_xeCJK_if_it_or_sl:n	2272
_xeCJK_check_num_range:nnNN	470, 470, 484, 2799	_xeCJK_if_it_or_sl:nT	2295
_xeCJK_check_single_save:N	1125, 1132, 1153, 1163, 1172, 1196, 1197, 1208, 1209, 1211, 1217, 1220, 1230, 1233	_xeCJK_if_it_or_sl:nTF	2250, <u>2267</u>
		_xeCJK_if_it_or_sl_p:n	<u>2267</u> , 2287, 2298
		_xeCJK_if_map_font_select:n	2316
		_xeCJK_if_map_font_select:nT	2307

_xeCJK_if_map_font_select:nTF	2316	_xeCJK_punct_hskip:n	829, 835, 1018, 3477
_xeCJK_if_map_font_select_p:n	2316	_xeCJK_punct_if_long:NT	1824
_xeCJK_info:nx	49, 55	_xeCJK_punct_if_long:NFT	973, 1800, 3619
_xeCJK_info:nxx	49, 56, 2437	_xeCJK_punct_if_long_p:N	3998
_xeCJK_int_until_do:wn	200, 201, 202, 203	_xeCJK_punct_if_middle:NT	946, 958, 967, 977, 3623
_xeCJK_kerning_width_or_ratio:nNN		_xeCJK_punct_if_middle:NFT	
.....	1724, 1725, 1743, 1743	841, 851, 1585, 1601, 1625, 3644
_xeCJK_listings_CJK_toks:	3856	_xeCJK_punct_if_mixed_width:NFT	1588, 1637
_xeCJK_listings_CJK_toks_hook:	3878, 3886, 3886	_xeCJK_punct_if_right:N	1466
_xeCJK_listings_append:nN		_xeCJK_punct_if_right:NFT	1699, 1702, 1779, 1782
.....	3926, 3926, 3940, 3952, 3967, 3988, 4008	_xeCJK_punct_kern:NN	836, 836, 987, 996, 1011, 1030
_xeCJK_listings_breaklines_toks: ...	3856, 3886, 3900	_xeCJK_punct_min_bound:NN	1712, 1715, 1759, 1759
_xeCJK_listings_escape:N	4091, 4092, 4093	_xeCJK_punct_nobreak_kern:NN ...	1009, 1009, 1014, 1415
_xeCJK_listings_initial_hook:	3834, 3843, 3843	_xeCJK_punct_rule:NN 820, 820, 843, 853, 863, 869, 882,	
_xeCJK_listings_inline_group:n	4076, 4080	892, 905, 947, 959, 968, 1017, 1024, 1062, 3647, 3671, 3684	
_xeCJK_listings_inline_group:w	4067, 4074, 4079	_xeCJK_punct_width_or_ratio:nN	
_xeCJK_listings_inside_convert:nw ..	4067, 4067, 4073	1586, 1589, 1590, 1648, 1648
_xeCJK_listings_output_IVS:	3840, 4040, 4040	_xeCJK_remove_duplicate_keys:N	2162, 2179, 2179
_xeCJK_listings_peek_active_loop:TF		_xeCJK_reset_char_class:n	
.....	4052, 4052, 4061, 4066	2866, 2867, 2868, 2869, 2870, 2900
_xeCJK_listings_process_CJK:nN		_xeCJK_restore_output_CJKsymbol: 3723, 3751, 3756, 3756	
.....	3889, 3891, 3893, 3897, 3914, 3920	_xeCJK_save_CJK_class:n	439, 444, 1379
_xeCJK_listings_process_Default:N ..	3875, 3914, 3914	_xeCJK_save_FullRight_check:	1042, 1052
_xeCJK_listings_process_FullLeft:nN .	3905, 3954, 3969	_xeCJK_save_FullRight_symbol:N	1043, 1053, 1069
_xeCJK_listings_process_FullRight:nN	3907, 3954, 3990	_xeCJK_save_family_info:	2169, 2337, 2337
_xeCJK_listings_process_IVS:nN	3877, 4033, 4033	_xeCJK_save_under_dot_CJKsymbol:N	
_xeCJK_listings_process_breaklines_CJK:nN	3721, 3727, 3749, 3764, 3785
.....	3903, 3911, 3954, 3954	_xeCJK_set_char_class_aux:Nnw	451, 460, 466, 2792
_xeCJK_listings_process_letter:nN		_xeCJK_set_char_class_eq:nn	
.....	3923, 3931, 3931, 4038	492, 492, 2855, 2856, 2857, 2858, 2859
_xeCJK_listings_process_other:nN ...	3924, 3931, 3942	_xeCJK_set_fake_font:nn	2265, 2279, 2279
_xeCJK_listings_toks_hook:	3835, 3859, 3859	_xeCJK_set_family_initial:	2129, 2129, 2156
_xeCJK_make_under_symbol:n	3720, 3748, 3769, 3769	_xeCJK_set_listings_escape:	4070, 4083, 4086, 4090
_xeCJK_map_it_sl:n	2305	_xeCJK_set_others_toks:n	3258, 3266, 3270
_xeCJK_map_it_sl:nn	2295, 2305	_xeCJK_set_special_punct:nn	1418, 1421, 1424, 1436
_xeCJK_margin_width_or_ratio:n .	1638, 1639, 1661, 1661	_xeCJK_set_sub_block_family:	2170, 2346, 2346
_xeCJK_msg_def_family_map:n	2540, 2542, 2756, 2763	_xeCJK_set_sub_class_toks:nn	1345, 1351, 1351
_xeCJK_msg_family_map:n		_xeCJK_set_verb_exspace:	2843, 2908, 2908
.....	1985, 2226, 2539, 2553, 2761, 2762, 2965	_xeCJK_set_verb_exspace:n	2925, 2931, 2931
_xeCJK_msg_new:nn		_xeCJK_set_verb_scale:nn	2940, 2950, 2950
.....	49, 49, 323, 1334, 1386, 1876, 1890, 1983,	_xeCJK_set_visible_space_size:n	2986, 2990
2225, 2443, 2537, 2670, 2752, 2759, 2961, 3057, 3101, 3169		_xeCJK_space_skip_scale:nnn	145, 149, 162, 166, 176
_xeCJK_original_kerning_margin:NN		_xeCJK_special_punct_seq:n	
.....	1687, 1694, 1694, 1757	1432, 1435, 1438, 1440, 1444, 1451, 1454, 1463
_xeCJK_parse_features:	2165, 2243, 2243	_xeCJK_special_punct_tl:nN	
_xeCJK_parse_features:nn	2243, 2244, 2245	1433, 1439, 1443, 1453, 1462, 1476
_xeCJK_pass_features:	2166, 2321, 2321	_xeCJK_sub_restore_or_cancel:n	1320, 1320, 1333
_xeCJK_peek_after_do:w	238, 249	_xeCJK_sub_restore_or_cancel:x	1307, 1316
_xeCJK_peek_catcode_false:w	210, 229, 234	_xeCJK_sub_special_punct:nn	1420, 1423, 1426, 1458
_xeCJK_peek_catcode_ignore_spaces_branches:w ..		_xeCJK_swap_cs_aux:w	114, 116, 117
.....	213, 215, 220	_xeCJK_switch_font:nn	
_xeCJK_peek_catcode_true:w	209, 226, 233	1360, 1365, 1374, 1376, 1383, 2433, 2433, 3567, 3572
_xeCJK_peek_ignore_spaces_branches:w ..	240, 242, 246	_xeCJK_text_composite:Nnw	3159, 3161
_xeCJK_prop_put_aux:n	2182, 2197	_xeCJK_tl_remove_outer_braces:w	100, 102
_xeCJK_prop_put_aux:nn	2182, 2199	_xeCJK_tmp:w	4087, 4098
_xeCJK_punct_breakable_kern:NN	1015, 1015, 1410	_xeCJK_token_value_charcode:w	255, 258, 270
_xeCJK_punct_dim_csname:nn	1480, 1487, 1491	_xeCJK_ulem_Boundary_and_Default:	
_xeCJK_punct_dim_csname:nnn 1482, 1489, 1493, 1790, 1846		3475, 3517, 3517, 3525
_xeCJK_punct_dim_csname:nnnn	1483, 1484	_xeCJK_ulem_Boundary_and_FullLeft_glue:N	
_xeCJK_punct_glue:NN	3483, 3587, 3587, 3597
.....	827, 827, 845, 855, 864, 871, 886, 896,	_xeCJK_ulem_CJK_and_Boundary_aux:	
935, 978, 1064, 3582, 3594, 3607, 3624, 3649, 3674, 3687		3478, 3527, 3527, 3534, 3740

`_xeCJK_ulem_CJK_and_CJK:N` 3474, 3542, 3542, 3551
`_xeCJK_ulem_CJK_and_FullLeft_glue:N`
. 3481, 3599, 3599, 3611
`_xeCJK_ulem_CJK_and_FullRight_glue:N`
. 3482, 3630, 3630, 3638
`_xeCJK_ulem_Default_and_FullLeft_glue:N`
. 3479, 3576, 3576, 3585
`_xeCJK_ulem_Default_and_FullRight_glue:N`
. 3480, 3613, 3613, 3628
`_xeCJK_ulem_FullLeft_and_CJK:` . . 3471, 3658, 3658, 3665
`_xeCJK_ulem_FullLeft_and_Default:`
. 3470, 3640, 3640, 3656
`_xeCJK_ulem_FullRight_and_CJK:` . . 3473, 3680, 3680, 3693
`_xeCJK_ulem_FullRight_and_Default:`
. 3472, 3667, 3667, 3678
`_xeCJK_ulem_between_CJK_blocks:nnN`
. 3492, 3494, 3498, 3559, 3559
`_xeCJK_ulem_ccglue:`
. 3547, 3564, 3606, 3620, 3688, 3701, 3707
`_xeCJK_ulem_class_group_begin:`
. 3548, 3553, 3553, 3609, 3636, 3663, 3691
`_xeCJK_ulem_fix_penalty:` 3476, 3536, 3536, 3540
`_xeCJK_ulem_glue:n` 3448, 3450, 3701, 3701
`_xeCJK_ulem_hskip:n` 3531, 3698, 3701, 3704, 3709
`_xeCJK_ulem_initial:` 3432, 3467, 3467
`_xeCJK_ulem_leader_type:` 3427, 3465, 3466
`_xeCJK_ulem_punct_hskip:n` 3477, 3695, 3695, 3699
`_xeCJK_ulem_restore_CJK_and_Boundary:` 3718, 3731, 3735
`_xeCJK_ulem_skip_punct_begin:`
. 3429, 3462, 3462, 3581, 3592, 3605, 3618
`_xeCJK_ulem_skip_punct_end:`
. 3430, 3462, 3464, 3652, 3675, 3689
`_xeCJK_ulem_swap_cs:NN` 3469, 3503, 3507
`_xeCJK_under_CJKsymbol:N` 3722, 3750, 3783, 3783
`_xeCJK_update_cs_case_tl:NNnn` . . 1079, 1085, 1090, 1094
`_xeCJK_update_family:n` 2206, 3206, 3218
`_xeCJK_update_inline_env_case_tl:`
. 1265, 1274, 1280, 1284, 1284
`_xeCJK_use_punct_dim:nn`
. 1486, 1604, 1616, 1657, 1732, 1733, 1784, 1785, 1823
`_xeCJK_use_punct_dim:nnn` 823, 831, 832,
837, 1020, 1021, 1022, 1488, 1565, 1568, 1689, 1690,
1698, 1701, 1764, 1765, 1769, 1770, 1778, 1781, 1817, 1818
`_xeCJK_verb_CJKecglue:` 2861, 2878
`_xeCJK_verb_CJKglue:` 2860, 2877
`_xeCJK_verb_addon:` 2844, 2850
`_xeCJK_verb_check_for_glue:` 2862, 2879
`_xeCJK_verb_ignore_spaces:w` 2863, 2880
`_xeCJK_warning:n` 52, 2665
`_xeCJK_warning:nx`
. 49, 53, 1885, 2531, 2644, 2727, 3097, 3098, 3099
`_xeCJK_warning:nxx` 54, 2221, 2745, 2953
`_xeCJK_zero_glue:` . . 857, 859, 912, 2822, 2823, 2828, 2829

`_` 1337, 1388, 3059, 3104, 3105, 3435, 3442, 3511, 4100

A

`\addCJKfontfeature` 2626, 2672
`\addCJKfontfeatures` 6, 2617, 2621, 2626
`\addto@hook` 3419
`\AfterEndPreamble` 78
`\AfterPreamble` 77

`KaiMingPunct` 4
`KaiMingPunct+` 4
`KaiMingPunct-` 4
`FallBack` 7
`\AllowBreakBetweenPuncts` 1404
`Mapping` 5
`\arraystretch` 3802
`\AssignTemplateKeys` 1558
`\AtBeginDocument` 81
`\AtEndOfPackage` 558, 2978, 3359
`\AtEndPreamble` 76
`\AutoFakeBold` 2013, 2108
`\AutoFakeSlant` 2013, 2108
`\AutoFallBack` 1927

B

`\begin` 3083
`\bfdefault` 2784
`\BODY` 3810, 3816, 3823
`\BoldFeatures` 2094
`\BoldFont` 2094
`\BoldItalicFeatures` 2094
`\BoldItalicFont` 2094
`\BoldSlantedFeatures` 2094
`\BoldSlantedFont` 2094
`\bool_gset_false:N` 3013, 3846, 4045
`\bool_gset_true:N` 2019, 2027, 3016, 3021, 4036
`\bool_if:NF` 1116, 1304, 1327, 2841, 2852, 3422
`\bool_if:nF` 916, 3974
`\bool_if:NT` 1313, 1669,
2282, 2749, 3094, 3403, 3404, 3717, 3724, 3726, 3730, 4043
`\bool_if:nT` 659, 778, 2286, 2296, 2646, 2654, 3341, 3995
`\bool_if:NTF` 683, 770, 774, 1107, 1161, 1170,
1178, 1190, 1195, 1573, 1608, 1621, 1685, 1711, 1714,
2284, 2873, 3426, 3934, 3945, 3957, 3972, 3993, 4013, 4024
`\bool_if:nTF` 104, 472, 591,
599, 607, 645, 652, 668, 676, 1145, 1184, 1678, 1828, 3312
`\bool_if:p:n` 2275
`\bool_new:N` 46, 235, 753, 1301, 1429,
2009, 2010, 2149, 2150, 2848, 2906, 3027, 3453, 3460, 4051
`\bool_set_eq:NN` 2140, 2141
`\bool_set_false:N` 211,
740, 1109, 1315, 1414, 2112, 2121, 3459, 3948, 4016, 4027
`\bool_set_true:N` 218, 735, 745, 1104, 1306, 1409,
2115, 2124, 2854, 3424, 3457, 3842, 3938, 3964, 3985, 4005
`\Boundary` 328
`\box_new:N` 44, 3734
`\box_use:N` 3778, 3786
`\box_wd:N` 3777, 3813

C

`\c_xeCJK_CJ_chars_clist` 372, 372, 388
`\c_xeCJK_CJK_chars_clist` 391, 391, 512, 2787
`\c_xeCJK_CL_chars_clist` 356, 356, 384
`\c_xeCJK_config_ext_tl` 3406, 3408
`\c_xeCJK_encoding_tl` 2471, 2782, 2784, 3093, 3093
`\c_xeCJK_EX_chars_clist` 369, 369, 386
`\c_xeCJK_FullLeft_chars_clist` 351, 351, 507, 2787
`\c_xeCJK_FullRight_chars_clist` 382, 382, 508, 2789
`\c_xeCJK_HalfLeft_chars_clist` 337, 337, 505
`\c_xeCJK_HalfRight_chars_clist` 337, 339, 506
`\c_xeCJK_IS_chars_clist` 371, 371, 387
`\c_xeCJK_IVS_chars_clist` 427, 427, 514

<code>\c_xeCJK_left_tl</code>	845, 855, 875, 882, 886, 889, 892, 896, 899, 905, 935, 947, 959, 968, 985, 1003, 1022, 1024, 1402, 1402, 1569, 1690, 1703, 1764, 1769, 1780, 1795, 1811, 1817, 3582, 3594, 3607, 3646, 3649
<code>\c_xeCJK_math_family_tl</code>	2780, 2782, 2784
<code>\c_xeCJK_math_tl</code>	2767, 2768, 2771, 2772, 2781, 2783, 2785
<code>\c_xeCJK_NormalSpace_chars_clist</code>	337, 341, 513
<code>\c_xeCJK_NS_chars_clist</code>	364, 364, 385
<code>\c_xeCJK_OP_chars_clist</code>	342, 342, 353
<code>\c_xeCJK_package_ext_tl</code>	61, 64
<code>\c_xeCJK_PO_chars_clist</code>	381, 381, 389
<code>\c_xeCJK_PR_chars_clist</code>	349, 349, 354
<code>\c_xeCJK_punct_style_plain_tl</code>	1792, 1848, 1866, 1875, 3838
<code>\c_xeCJK_right_tl</code>	843, 853, 863, 864, 869, 871, 939, 952, 964, 978, 994, 1017, 1021, 1028, 1061, 1062, 1064, 1402, 1403, 1569, 1689, 1700, 1765, 1770, 1783, 1796, 1813, 1818, 3624, 3647, 3671, 3674, 3684, 3687
<code>\c_catcode_letter_token</code>	1139, 1159
<code>\c_eleven</code>	594, 662, 914, 1105
<code>\c_four</code>	137, 158, 3263
<code>\c_fourteen</code>	2825, 2832
<code>\c_group_begin_token</code>	556
<code>\c_group_end_token</code>	557
<code>\c_math_toggle_token</code>	768, 1215
<code>\c_max_dim</code>	1498, 1504, 1515, 1518, 1520, 1522, 1599, 1627, 1650, 1654, 1663, 1718, 1745, 1753
<code>\c_minus_one</code>	913
<code>\c_one</code>	295, 329, 912, 927, 1812, 2604, 2811, 3263, 3593, 3928, 3966
<code>\c_one_fp</code>	1499, 1505, 1523
<code>\c_one_thousand</code>	131, 185
<code>\c_seven</code>	152, 167, 2793
<code>\c_space_tl</code>	741, 3523
<code>\c_space_token</code>	217, 244
<code>\c_ten</code>	655, 671
<code>\c_ten_thousand</code>	58
<code>\c_thirteen</code>	922
<code>\c_three</code>	136, 157, 331, 1494, 1495, 1526, 1814, 2813, 3261, 3711, 3977, 4007
<code>\c_true_bool</code>	2956
<code>\c_two</code>	135, 145, 151, 330, 832, 1605, 1632, 2812, 2934, 2941, 2987, 2995, 3002, 3889, 3891, 3893, 3897, 3903, 3905, 3907, 3911, 3959, 3976, 3987, 3997
<code>\c_two_hundred_fifty_five</code>	332
<code>\c_two_hundred_fifty_six</code>	3385
<code>\c_xeCJK_math_fam_int</code>	2785, 2793
<code>\c_xeCJK_space_skip_tl</code>	129, 129, 595, 663
<code>\c_zero</code>	57, 296, 328, 924, 928, 929, 2810, 2839, 3225, 3877
<code>\c_zero_dim</code>	824, 825, 1512, 1524, 1794, 1795, 1796, 1801, 1833, 1849, 2897, 2935, 3066, 3772, 3804, 3805
<code>\c_zero_fp</code>	1519, 1521, 1653, 1721, 1748
<code>\c_zero_skip</code>	133, 142, 147, 164, 860, 3531
<code>\char_set_catcode:nn</code>	3354
<code>\char_set_catcode_active:N</code>	4097
<code>\char_set_catcode_ignore:n</code>	297
<code>\char_set_catcode_letter:n</code>	3352
<code>\char_set_lccode:nn</code>	3396
<code>\char_value_catcode:n</code>	3354
<code>\CheckFullRight</code>	1035
<code>\CheckSingle</code>	1118
<code>\CJK</code>	328
<code>\CJK@@UL</code>	3455, 3455, 3458
<code>\CJK@hundredmillion</code>	3390
<code>\CJK@ifundefined</code>	3345, 3346
<code>\CJK@postUnderdot</code>	3729
<code>\CJK@preUnderdot</code>	3719
<code>\CJK@tenthousand</code>	3389
<code>\CJK@UL</code>	3455, 3455, 3456
<code>\CJK@underdotBox</code>	3747, 3778
<code>\CJK@underdotSkip</code>	3720
<code>\CJK@UnicodeEnc</code>	3381
<code>\CJKaddEncHook</code>	3379, 3379
<code>\CJKecglue</code> 587, 604, 612, 633, 657, 674, 725, 729, 736, 746, 748, 771, 2823, 2829, 2861, 2878, 2893, 2895, 2896, 3445, 3449
<code>xCJKecglue</code>	3
<code>\CJKfamily</code>	6, 2484, 2484, 2516
<code>\CJKfamilydefault</code>	6, 1968, 1978, 2455, 2456, 2685, 2688, 2712, 2714, 2722, 2727, 2729, 2731, 2732, 2736, 2742, 2746, 2748, 2774, 2775, 3107, 3116, 3117, 3121, 3213
<code>\CJKfilltwosides</code>	3790
<code>\CJKfixedspacing</code>	2907
<code>\CJKfontspec</code>	6, 2584, 2592
<code>\CJKglue</code>	647, 650, 684, 716, 720, 792, 865, 896, 974, 2822, 2828, 2860, 2877, 2891, 3438, 3447, 3793, 3801, 4047
<code>\CJKmath</code>	2765
<code>\CJKnospace</code>	3135, 3136
<code>\CJKpunctsymbol</code>	883, 893, 906, 989, 1007, 1058, 3062, 3063
<code>\CJKrmdefault</code>	6, 2546, 2557, 2565, 2718, 2732, 2735, 2736, 3107, 3107, 3118, 3210
<code>\CJKsetecglue</code>	3133, 3133, 3134
<code>\CJKsfdefault</code>	6, 2547, 2558, 2571, 2719, 3107, 3108, 3211
<code>\CJKspace</code>	754, 3135, 3135
<code>\CJKsymbol</code>	579, 641, 792, 794, 796, 1364, 1365, 1932, 1934, 1935, 1940, 1941, 3062, 3062, 3549, 3568, 3573, 3721, 3722, 3727, 3749, 3750, 3763
<code>\CJKttdefault</code>	6, 2548, 2559, 2576, 2720, 3107, 3109, 3212
<code>\CJKkunderanysymbol</code>	3744, 3744
<code>\CJKkunderdot</code>	3715, 3715
<code>\clist_clear:c</code>	2137
<code>\clist_clear:N</code>	2064, 2139, 2183, 2636
<code>\clist_concat:NNN</code>	2160, 2657, 2660, 2696, 2786, 2788
<code>\clist_const:Nn</code>	337, 339, 341, 342, 349, 356, 364, 369, 371, 372, 381, 391, 427
<code>\clist_const:Nx</code>	351, 382
<code>\clist_gconcat:ccN</code>	454
<code>\clist_gconcat:NNN</code>	468
<code>\clist_gset:Nn</code>	1431, 2619
<code>\clist_if_empty:cF</code>	2325
<code>\clist_map_function:nN</code>	444
<code>\clist_map_inline:cn</code>	495, 2903
<code>\clist_map_inline:Nn</code>	456, 1434, 1471, 2790
<code>\clist_map_inline:nn</code>	17, 571, 583, 619, 799, 815, 817, 1269, 1278, 1322, 1380, 1998, 2637
<code>\clist_new:c</code>	306, 317
<code>\clist_new:N</code>	48, 1430, 2090, 2148, 2175, 2617, 2667, 2668
<code>\clist_put_left:cx</code>	2235, 2240
<code>\clist_put_left:Nn</code>	2002
<code>\clist_put_right:Nn</code>	2332
<code>\clist_put_right:No</code>	2188
<code>\clist_put_right:Nx</code>	2190, 2327, 2698
<code>\clist_remove_all:Nn</code>	2050, 2371, 2634
<code>\clist_set:Nn</code>	453, 2085, 2158, 2632
<code>\cs:w</code>	122, 3343
<code>\cs_end:</code>	122, 276, 1476, 3343

\cs_generate_variant:Nn	281, 468, 469, 520, 530, 536, 1067, 1245, 1333, 1350, 1399, 1838, 1856, 2091, 2092, 2176, 2196, 2224, 2231, 2232, 2278, 2409, 2476, 2535, 2614, 2615, 2669, 2707, 2708
\cs_gset_eq:cc	2404
\cs_gset_eq:NN	111, 121, 2981, 3195, 3196, 3721, 3749
\cs_gset_nopar:Npx	86, 3305
\cs_gset_protected_nopar:cpx	2203, 2375, 2469
\cs_if_eq:NNF	1040, 1123, 1932
\cs_if_eq:NNT	1050, 1131, 1940
\cs_if_eq:NNTF	3358, 4093
\cs_if_exist:cT	3150
\cs_if_exist:cTF	2454, 2479
\cs_if_exist:NF	790, 1392, 3345
\cs_if_exist_p:N	3342
\cs_if_exist_use:cF	2415, 2424, 2446, 2974, 3165
\cs_if_exist_use:cTF	1575, 2481
\cs_if_free:NFT	3346
\cs_new:Npn	97, 102
\cs_new_eq:cN	322
\cs_new_eq:NN	267, 272, 556, 557, 835, 981, 1014, 2432, 2568, 2626, 2907, 3111, 3134, 3167, 3187, 3188, 3218, 3220, 3295, 3296, 3309
\cs_new_nopar:Npn	93, 176, 252, 254, 258, 270, 279, 321, 445, 521, 1058, 1432, 1433, 1480, 1482, 1484, 1486, 1488, 1648, 1661, 1694, 1706, 1743, 1759, 1773, 1839, 2314, 2412, 2463, 2464, 2465, 2542, 2553, 3062, 3063, 3155, 3161, 3509, 3881
\cs_new_protected:Npn	68, 70, 72, 108, 110, 112, 200, 202, 206, 236, 1070, 1094, 1284, 3379, 3756, 3769, 4052, 4067, 4080, 4090, 4092
\cs_new_protected_nopar:Npn	49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 82, 84, 95, 119, 186, 215, 242, 300, 311, 439, 451, 470, 482, 492, 518, 523, 525, 531, 537, 543, 552, 554, 589, 643, 698, 704, 709, 766, 789, 792, 813, 820, 827, 836, 839, 849, 859, 861, 867, 873, 885, 887, 895, 897, 908, 937, 950, 962, 971, 982, 991, 1001, 1009, 1015, 1026, 1059, 1068, 1137, 1157, 1203, 1213, 1222, 1236, 1320, 1340, 1351, 1436, 1447, 1458, 1490, 1492, 1563, 1672, 1788, 1809, 1844, 1946, 1957, 1991, 2039, 2061, 2076, 2129, 2153, 2177, 2179, 2197, 2199, 2201, 2215, 2227, 2229, 2233, 2238, 2243, 2245, 2279, 2305, 2321, 2337, 2346, 2359, 2390, 2413, 2422, 2433, 2444, 2466, 2505, 2515, 2518, 2524, 2597, 2608, 2627, 2675, 2769, 2797, 2850, 2900, 2908, 2931, 2950, 2972, 2983, 2990, 3223, 3270, 3393, 3420, 3462, 3464, 3467, 3503, 3517, 3527, 3536, 3542, 3553, 3559, 3576, 3587, 3599, 3613, 3630, 3640, 3658, 3667, 3680, 3695, 3701, 3707, 3709, 3735, 3783, 3843, 3859, 3886, 3900, 3914, 3920, 3926, 3931, 3942, 3954, 3969, 3990, 4033, 4040, 4074
\cs_new_protected_nopar:Npx	2589
\cs_set:Npn	4087
\cs_set_eq:NN	109, 114, 115, 116, 208, 736, 741, 748, 1042, 1043, 1044, 1045, 1052, 1053, 1125, 1126, 1132, 1410, 1415, 1934, 1935, 1941, 2206, 2516, 2711, 2822, 2823, 2828, 2829, 2860, 2861, 2862, 2863, 2877, 2878, 2879, 2880, 2895, 2896, 3168, 3346, 3386, 3427, 3435, 3436, 3437, 3442, 3443, 3444, 3455, 3465, 3722, 3727, 3750, 3762, 3793, 3801, 3856, 3951, 4047, 4066, 4073, 4079
\cs_set_nopar:Npn	3310, 3384
\cs_set_protected_nopar:Npn	720, 729, 746, 2891, 2893, 3447, 3449, 4011, 4022
\cs_set_protected_nopar:Npx	2864
\cs_to_str:N	2588
\cs_undefine:c	1439, 1462, 2220
\cs_undefine:N	117, 2725, 3199, 3200
\curr@fontshape	2910, 2913, 2916, 2946, 2974, 2985
\CurrentOption	3413, 3830

D

\DeclareExpandableDocumentCommand	2517
\DeclareInstance	1887
\DeclareObjectType	1494
\DeclareOption	3413, 3830
\DeclareSymbolFont	2781
\DeclareTemplateCode	1526
\DeclareTemplateInterface	1495
\def	3386
\Default	328
\defaultCJKfontfeatures	6, 2617, 2618, 2620, 3090
\dim_compare:nNnF	2995
\dim_compare:nNnTF	1599, 1627, 1650, 1663, 1718, 1745, 1753, 2935, 3805, 3813
\dim_const:cn	701
\dim_eval:n	178, 832, 1676, 1696, 1775, 1823, 2999, 3792, 3815
\dim_gset:cn	701
\dim_if_exist:cTF	700
\dim_max:nn	1596, 1610, 1708, 1715, 1761, 3187, 3195
\dim_min:nn	1611, 1622, 1670, 1763, 1768, 3188, 3196
\dim_new:N	45, 1646, 1647
\dim_ratio:nn	3002
\dim_set:Nn	1565, 1566, 1571, 1815
\dim_set_eq:NN	3804
\dim_to_fp:n	2941, 2942
\dim_use:N	1821, 1834, 1841, 2926, 2987
\document	86, 89

E

xeCJKactive	3
\EditInstance	1899
\else:	127, 195, 197, 222, 227, 248, 277, 286, 696, 1469, 1477, 2276, 3513, 3937, 3963, 3984, 4004, 4060
\EmboldenFactor	2013
\end	3083
\endminipage	3796
\endtabular	3811, 3817, 3824
\EnvCS	1253
Verb	5
\etex_currentgrouplevel:D	2839
\etex_currentgrouptype:D	2825, 2832
\etex_dimexpr:D	1631, 1657, 1667, 1731, 1757
\etex_fontcharwd:D	1821, 2926
\etex_glueshrink:D	172
\etex_gluestretch:D	171
\etex_iffontchar:D	126
\etex_lastnodetype:D	594, 655, 662, 671, 910, 927, 1105, 3593
NewLineCS	3
NewLineCS+	3
NewLineCS-	3
\exp_after:wN	121, 122, 188, 189, 203, 219, 220, 225, 226, 228, 229, 245, 246, 249, 255, 1149, 1150, 1152, 2065, 2883, 2885, 2886, 2887, 2924, 2925, 2926, 2986, 2987, 3247, 3248, 3249, 3250, 3343, 3512, 3514, 3728, 3759, 3760, 3762, 3763, 3764, 3848, 3850, 3851, 3852, 4076, 4077
\exp_args:Nc	305, 1473

\exp_args:NNv	3152	\g__xeCJK_family_font_name_prop	1986, 1995, 2217, 2334, 2335, 2340, 2362, 2365, 2397, 2526, 2629, 2677, 2684, 2726, 2739
\exp_args:No	1148	\g__xeCJK_family_font_options_prop	2334, 2336, 2342, 2368, 2372, 2398, 2651, 2680, 2687, 2691
\exp_args:NV	96	\g__xeCJK_family_int	2131, 2145, 2604
\exp_last_unbraced:Nf	99	\g__xeCJK_family_name_prop	2219, 2334, 2334, 2380, 2396, 2468, 2778
\exp_not:c	802, 2872	\g__xeCJK_features_id_prop	2094, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2134, 2244, 2323
\exp_not:N	224, 790, 2210, 2380, 2381, 2385, 2471, 2472, 2473, 2718, 2719, 2720, 3114, 3118, 3792	\g__xeCJK_fontspec_prop	2599, 2610, 2616
\exp_not:n	528, 534, 2206, 2207, 2711, 2875, 3759	\g__xeCJK_indent_bool	3031, 3094
\exp_not:o	89, 2191, 3306	\g__xeCJK_last_punct_tl	838, 838, 841, 843, 845, 851, 853, 855, 863, 864, 869, 871, 881, 891, 904, 945, 957, 966, 986, 987, 988, 995, 996, 997, 1004, 1005, 1006, 1029, 1030, 1031, 1061, 1062, 1064, 3644, 3646, 3647, 3649, 3671, 3674, 3684, 3687
\exp_not:V	541, 549, 2072, 2073, 2191, 2208, 2209, 2702, 2703, 3117	\g__xeCJK_listings_IVS_bool	3846, 4036, 4043, 4045, 4051
\exp_not:v	2315, 2328	\g__xeCJK_math_bool	2749, 2765
\exp_stop_f:	126, 193, 196, 285, 1842	\g__xeCJK_new_class_seq	298, 299, 308, 3261, 3265
\extrarowheight	3803, 3804	\g__xeCJK_non_CJK_class_seq	433, 435, 436, 797
F		\g__xeCJK_number_bool	3030, 3403
\f@baselineskip	3005	\g__xeCJK_punct_style_seq	1880, 1886, 1889
\f@encoding	2471, 2992, 3314	\g__xeCJK_punct_width_tl	1427, 1578, 1580
\f@family	2472, 2967, 2968, 2993, 3205	\g__xeCJK_scale_family_prop	2917, 2958, 2971
\f@series	2412	\g__xeCJK_slant_factor_fp	2012, 2028, 2032, 2127, 2143
\f@shape	2412	\g__xeCJK_spacefactor_int	131, 144, 161, 179, 182, 184, 185, 623
\f@size	2412, 2910, 2913, 2946, 2974, 2985, 3001	\g__xeCJK_special_punct_clist	1430, 1431, 1434, 1471
\FallBack	2093	\g__xeCJK_sub_key_seq	2038, 2038, 2041, 2633, 2639, 2649
\familydefault	2716, 3213	\g__xeCJK_under_symbol_box	3734, 3772, 3786
\fi:	127, 197, 198, 203, 230, 231, 250, 277, 286, 696, 1469, 1477, 2276, 2834, 3515, 3857, 3937, 3950, 3963, 3984, 4004, 4018, 4019, 4029, 4030, 4037, 4049, 4060, 4077	\g__xeCJK_unknown_family_seq	2528, 2530, 2536
\file_input:n	3353	\g_fontspec_encoding_tl	2992, 3093
\fix@penalty	3220, 3222, 3539	\group_align_safe_begin:	212, 239, 1166, 1224, 1238
\fontfamily	3203, 3203	\group_align_safe_end:	209, 210, 238, 1169, 1177, 1183, 1227, 1241, 1248, 1255
\fontsize	2997	\group_begin:	83, 1993, 2155, 2205, 2339, 2924, 3295, 3383, 3395, 3746, 4086
\fontspec_set_family:Nnn	2207	\group_end:	83, 2006, 2171, 2211, 2344, 2924, 3296, 3388, 3399, 3753, 4089
\fontspec_visible_space:	2980	H	
\fontspec_visible_space_fallback:	2981	\HalfLeft	333
\footnote	3084	\HalfRight	333
\footnotemark	3084	\hbox_overlap_left:n	3786
\fp_compare:nNnTF	256, 1653, 1721, 1748	\hbox_set:Nn	188, 3747, 3771, 3807
\fp_eval:n	179, 2954, 2955	\hbox_to_zero:n	3775
\fp_gset:Nn	2020, 2028	\hbox_unpack:N	3819
\fp_new:N	2011, 2012, 2151, 2152, 2970	CheckFullRight	5
\fp_set:Nn	2116, 2125, 2952	CheckSingle	3
\fp_set_eq:NN	2142, 2143	\hskip	3437, 3444
\fp_use:c	1656, 1666, 1750	I	
\fp_use:N	1630, 1730, 1755, 2236, 2241, 2957	\icprotect	3342, 3343
\FullLeft	328	MiddlePunct	4
\FullRight	328	MiddlePunct+	4
G		MiddlePunct-	4
\g__xeCJK_after_end_preamble_hook_tl	67, 73, 78, 85	\if_case:w	193, 196, 2821
\g__xeCJK_after_preamble_hook_tl	66, 71, 77, 81	\if_catcode:w	223
\g__xeCJK_at_end_preamble_hook_tl	65, 69, 76, 83	\if_cs_exist:w	276, 1476
\g__xeCJK_auto_fake_bold_bool	2009, 2016, 2019, 2140	\if_dim:w	695
\g__xeCJK_auto_fake_slant_bool	2010, 2024, 2027, 2141	\if_false:	4077
\g__xeCJK_base_class_seq	433, 433, 434, 1353	\if_int_compare:w	201, 284, 1468
\g__xeCJK_CJK_class_seq	433, 438, 441, 553, 3273		
\g__xeCJK_CJK_sub_class_seq	1293, 1293, 1369, 1378, 3485, 3487, 3869, 3894, 3908		
\g__xeCJK_class_seq	298, 298, 307, 318, 434, 560		
\g__xeCJK_config_bool	3013, 3016, 3021, 3027, 3404		
\g__xeCJK_config_name_tl	3017, 3022, 3026, 3408		
\g__xeCJK_default_features_clist	2161, 2617, 2619		
\g__xeCJK_embolden_factor_fp	2011, 2020, 2031, 2118, 2142		

\if_meaning:w	217, 244, 3511	\l__xeCJK_add_block_features_clist	2636, 2661, 2668, 2698
\if_predicate:w	2274	\l__xeCJK_add_font_features_clist	2632, 2634, 2658, 2667, 2697
\IfBooleanF	2488	\l__xeCJK_add_font_prop	2132, 2146, 2228, 2230, 2331
\IfBooleanT	449, 1298, 1308, 1317, 2498	\l__xeCJK_add_min_bound_to_margin_bool	1541, 1669
\IfBooleanTF	2492	\l__xeCJK_auto_fake_bold_bool	2112, 2115, 2140, 2149, 2282, 2297
\ifCTEX@fntef	3358	\l__xeCJK_auto_fake_slant_bool	2121, 2124, 2141, 2150, 2284
\IfInstanceExistTF	1869, 1884, 1898	\l__xeCJK_begin_int	204, 484, 486, 488, 489, 2799, 2800, 2803, 2804
\IfNoValueF	3349	\l__xeCJK_bound_dim	1565, 1611, 1622, 1632, 1646, 1667, 1670
\IfNoValueTF	478, 479, 2588	\l__xeCJK_ccglue_skip	721, 724, 3439, 3448, 3571, 3708
\ignorespacesafterend	3797, 3827	\l__xeCJK_check_single_cs_case_tl	1226, 1235, 1250, 1257
\indentfirst	3028	\l__xeCJK_current_coor_tl	2417, 2419, 2426, 2428, 2431, 2448, 2460, 2916, 2918, 2959, 2985, 2988
\InlineEnv	1260	\l__xeCJK_different_align_margin_dim	1551
\int_add:Nn	3928	\l__xeCJK_different_align_ratio_fp	1552
\int_case:nnn	910	\l__xeCJK_ecglue_skip	664, 730, 747, 752, 3446, 3450, 3522
\int_compare:nNnF	2839, 3260, 3593, 3959	\l__xeCJK_embolden_factor_fp	2116, 2142, 2151, 2236
\int_compare:nNnT	2832	\l__xeCJK_enabled_global_setting_bool	1528, 1573, 1680
\int_compare:nNnTF	131, 144, 161, 924, 927, 2825, 3225, 3711, 4060	\l__xeCJK_enabled_kerning_bool	1544, 1685
\int_compare_p:nNn	594, 655, 662, 671, 3976, 3977, 3997	\l__xeCJK_end_int	205, 484, 486, 2799, 2800
\int_const:cn	316	\l__xeCJK_env_cs_case_tl	1251, 1254, 1258
\int_const:Nn	2785	\l__xeCJK_env_cs_seq	1254
\int_div_truncate:nn	181	\l__xeCJK_fallback_family_tl	1952, 1953, 1962, 1965, 1973, 1994, 1996, 2000, 2001, 2003, 2008
\int_eval:n	260, 271, 280, 2604	\l__xeCJK_family_default_init_tl	2712, 3110, 3112, 3121
\int_gincr:N	2131	\l__xeCJK_family_name_tl	2049, 2157, 2167, 2168, 2173, 2341, 2343, 2350, 2363, 2369, 2378, 2381
\int_gset_eq:NN	185, 623	\l__xeCJK_fixed_margin_ratio_fp	1536
\int_if_exist:cF	1342	\l__xeCJK_fixed_margin_width_dim	1535
\int_if_exist:cTF	302, 313, 1324	\l__xeCJK_fixed_punct_ratio_fp	1530
\int_incr:N	489, 2804	\l__xeCJK_fixed_punct_width_dim	1529
\int_max:nn	479	\l__xeCJK_font_name_bf_tl	2259, 2261
\int_min:nn	478	\l__xeCJK_font_name_tl	1996, 1997, 2068, 2069, 2159, 2174, 2209, 2341, 2630, 2663, 2693
\int_new:N	43, 184, 204, 205, 2145, 2816, 4010	\l__xeCJK_font_options_clist	2050, 2158, 2160, 2161, 2162, 2164, 2175, 2343, 2652, 2657, 2658, 2660, 2661, 2663
\int_set:cn	3272	\l__xeCJK_font_options_prop	2181, 2184, 2195, 2198, 2200
\int_set:Nn	474, 478, 479, 485, 494, 2902	\l__xeCJK_fontspec_family_tl	2207, 2210, 2214, 2381, 2385
\int_set_eq:NN	475, 2810, 2811, 2812, 2813, 3966, 3987, 4007	\l__xeCJK_fontspec_options_clist	2139, 2148, 2164, 2208, 2327, 2332
\int_step_inline:nnnn	3263	\l__xeCJK_inline_env_case_tl	1240, 1286, 1288, 1290
\int_to_hexadecimal:n	1973	\l__xeCJK_inline_env_seq	1264, 1271, 1272, 1279, 1283, 1287
\int_use:N	256	\l__xeCJK_kerning_margin_minimum_dim	1555, 1709
\int_while_do:nNnn	1105	\l__xeCJK_kerning_margin_ratio_fp	1554, 1755
\int_zero:N	4019, 4030	\l__xeCJK_kerning_margin_width_dim	1553, 1753, 1754
\ItalicFeatures	2094	\l__xeCJK_kerning_total_ratio_fp	1547, 1721, 1730
\ItalicFont	2094	\l__xeCJK_kerning_total_width_dim	1546, 1718, 1719
\IVS	333	\l__xeCJK_listings_env_bool	2841, 2848, 3842
		\l__xeCJK_listings_flag_int	3959, 3966, 3976, 3977, 3987, 3997, 4007, 4010, 4019, 4030
J			
CJKecglue	3	\l__xeCJK_listings_letter_bool	3934, 3938, 3945, 3948, 3957, 3964, 3972, 3985, 3993, 4005, 4013, 4016, 4024, 4027
CJKglue	3	\l__xeCJK_margin_minimum_dim	1543, 1597
CJKmath	3	\l__xeCJK_middle_margin_ratio_fp	1540, 1630
CJKspace	3	\l__xeCJK_middle_margin_width_dim	1539, 1627, 1628
K			
\KaiMingPunct	1404	\l__xeCJK_middle_punct_ratio_fp	1534
\keys_define:nn	288, 716, 725, 754, 1035, 1074, 1118, 1260, 1291, 1404, 1857, 1927, 2013, 2042, 2102, 2108, 2765, 2807, 3009, 3028, 3034, 3095	\l__xeCJK_middle_punct_width_dim	1533
\keys_set:nn	3064, 3124	\l__xeCJK_min_bound_to_kerning_bool	1545, 1711
\keys_set_known:nVN	2163	\l__xeCJK_mixed_margin_ratio_fp	1538
\keyval_parse:NNn	2196	\l__xeCJK_mixed_margin_width_dim	1537
\keyval_parse:NNV	2182		
\KeyValue	1500, 1501, 1502, 1503, 1506, 1507, 1508, 1509		
L			
\l__keys_module_tl	3059		
\l__peek_search_token	208, 224		

\l__xeCJK_mixed_punct_ratio_fp	1532	2509, 2510, 2514, 2514, 2600, 2601, 2630, 2652, 2678,
\l__xeCJK_mixed_punct_width_dim	1531	2681, 2692, 2910, 2913, 2916, 2918, 2920, 2946, 2959, 2965
\l__xeCJK_new_line_cs_case_tl	1247, 1251, 1258	\l__xeCJK_punct_style_tl
\l__xeCJK_new_line_cs_seq	1247	1481, 1483, 1792, 1804, 1848, 1852, 1866, 1870, 1874, 3838
\l__xeCJK_no_break_cs_case_tl	1063, 1101	\LA@space
\l__xeCJK_no_break_cs_seq	1101	3511
\l__xeCJK_optimize_kerning_bool	1548, 1714	PlainEquation
\l__xeCJK_optimize_margin_bool	1542, 1608, 1621	3
\l__xeCJK_peek_ignore_spaces_bool		SlantFactor
...	211, 218, 235, 770, 774, 1161, 1170, 1178, 1190, 1195	4
\l__xeCJK_plain_equation_bool	1186, 1292	AllowBreakBetweenPuncts
\l__xeCJK_punct_breakable_bool	1409, 1414, 1429, 3978	5
\l__xeCJK_reserve_space_bool	683, 756, 781	\LocalConfig
\l__xeCJK_restore_listings_toks_tl		3009
...	3839, 3847, 3851, 3861, 3871, 3880	\LongPunct
\l__xeCJK_reverse_bound_dim		1404
...	1566, 1611, 1615, 1632, 1647, 1670	\lst@AddToHook
\l__xeCJK_same_align_margin_dim	1549	3834, 3835, 3836, 3842
\l__xeCJK_same_align_ratio_fp	1550	\lst@Append
\l__xeCJK_scale_factor_fp	2952, 2954, 2957, 2970	3929, 4020, 4031
\l__xeCJK_slant_factor_fp	2125, 2143, 2152, 2241	\lst@AppendLetter
\l__xeCJK_sub_cancel_bool	1301, 1304, 1306, 1313, 1315, 1327	4011, 4011
\l__xeCJK_sub_family_name_tl		\lst@AppendOther
...	2088, 2350, 2354, 2361, 2366, 2373, 2376, 2384	4011, 4022
\l__xeCJK_sub_font_name_tl		\lst@carg
...	2004, 2067, 2068, 2069, 2073, 2079,	4071, 4082
...	2081, 2083, 2084, 2089, 2363, 2366, 2678, 2685, 2693, 2703	\lst@FillOutputBox
\l__xeCJK_sub_font_options_clist		4046
...	2002, 2004, 2064, 2072, 2085,	\lst@ifbreaklines
...	2090, 2369, 2371, 2373, 2681, 2688, 2692, 2696, 2697, 2702	3855, 4019, 4030
\l__xeCJK_sub_key_prop	2048, 2054, 2070, 2133, 2147, 2348	\lst@ifec
\l__xeCJK_sub_key_seq	2635, 2641, 2646, 2655	4060
\l__xeCJK_tmp_bool	46, 1104, 1107, 1109, 1116	\lst@ifflexible
\l__xeCJK_tmp_box	44, 188, 3771, 3777, 3807, 3813, 3819	4035, 4042
\l__xeCJK_tmp_clist	48, 453, 455, 456, 2786, 2788, 2789, 2790	\lst@ifletter
\l__xeCJK_tmp_dim	45, 1571, 1599, 1604, 1615, 1815, 1823, 1834	3937, 3950, 3963, 3984, 4004, 4018, 4029
\l__xeCJK_tmp_int	43, 322, 485, 488, 494, 496, 2902, 2904	\lst@ifNextCharActive
\l__xeCJK_tmp_skip	47, 1110, 1112, 1116	4066
\l__xeCJK_tmp_tl	42, 539,	\lst@InlineGJ
...	540, 541, 545, 546, 548, 549, 2063, 2065, 2079, 2084,	4079
...	2186, 2187, 2191, 2217, 2219, 2221, 2401, 2402, 2588,	\lst@InlineGJEnd
...	2589, 2590, 2731, 2746, 2772, 2775, 2778, 2780, 4069, 4071	4084
\l__xeCJK_ulem_hook_used_bool		\lst@InsideConvert@
...	3422, 3424, 3453, 3717, 3724, 3726, 3730	4073
\l__xeCJK_ulem_skip_punct_bool	3426, 3457, 3459, 3460	\lst@lastother
\l__xeCJK_underdot_output_tl	3728, 3758, 3765, 3768	3951, 4031
\l__xeCJK_verb_addon_bool	2852, 2854, 2906	\lst@length
\l__xeCJK_verb_case_int	2810, 2811, 2812, 2813, 2816, 2821	3928
\l__xeCJK_verb_exspace_skip		\lst@letterfalse
...	2892, 2894, 2912, 2921, 2930, 2933, 2935, 2937, 2947	3947, 3950, 4026, 4029
\l__xeCJK_xecglue_bool		\lst@lettertrue
...	593, 661, 735, 740, 745, 753, 2873, 2874	3935, 3937, 3960, 3963, 3981, 3984, 4001, 4004, 4015, 4018
\l_keys_choice_tl	2016, 2024	\lst@numberstyle
\l_keys_key_tl	3053, 3054, 3097, 3098, 3099	3847
\l_keys_value_tl	1869, 1870, 1871, 2020, 2028, 2116, 2125, 3022	\lst@Output
\l_peek_token	217, 224, 244, 627, 780, 782, 790, 1063,	3937, 3947,
...	1142, 1147, 1148, 1149, 1150, 1167, 1175, 1187, 1207, 1225	3950, 3959, 3963, 3980, 3984, 4000, 4004, 4015, 4026, 4029
\l__xeCJK_current_font_tl		\lst@OutputOther
...	1481, 1485, 2410, 2410, 2411, 2415, 2417	3937, 3963, 3984, 4004, 4018
\l__xeCJK_family_tl	1952, 1968, 1978, 1979, 2411, 2418,	\lst@ProcessLetter
...	2446, 2449, 2450, 2452, 2456, 2459, 2489, 2496, 2497,	3917
		\lst@ProcessOther
		3918
		\lst@whitespacefalse
		3933, 3944, 3956, 3971, 3992
		M
		\makexeCJKactive
		291, 295, 295
		\makexeCJKinactive
		292, 295, 296, 3306, 3308, 3317, 3323, 3333
		EmboldenFactor
		4
		\mddefault
		2782
		\MiddlePunct
		1404
		\minipage
		3792
		\msg_critical:nn
		10
		\msg_critical:nnn
		37
		\msg_error:nn
		50
		\msg_error:nnn
		20, 40
		\msg_error:nnx
		51
		\msg_info:nnx
		55
		\msg_info:nnxx
		56
		\msg_new:nnn
		3, 11, 28, 49
		\msg_redirect_module:nnn
		3038, 3039, 3045, 3046
		\msg_warning:nn
		52
		\msg_warning:nnnnn
		1972
		\msg_warning:nnx
		53
		\msg_warning:nnxx
		54
		N
		indentfirst
		3
		\newCJKfontfamily
		6, 2584, 2586
		\NewDocumentCommand
		295, 296, 446, 466,
		498, 503, 510, 1102, 1294, 1302, 1311, 1397, 1400, 1882,

<code>\seq_map_inline:Nn</code>	560, 797, 1097, 1287, 1353, 1369, 2633, 3273, 3485, 3487, 3869, 3894, 3908
<code>\seq_new:c</code>	1435
<code>\seq_new:N</code>	298, 299, 433, 435, 438, 1073, 1283, 1293, 1889, 2038, 2536
<code>\seq_put_right:Nn</code>	1084, 1272, 2641
<code>\seq_remove_all:Nn</code>	1089, 1279
<code>\seq_set_from_clist:Nn</code>	1264
<code>\seq_set_split:Nnn</code>	1078
<code>\seq_use:Nnnn</code>	1880
<code>\setCJKfallbackfamilyfont</code>	7, 7, 1989, 1989
<code>\setCJKfamilyfont</code>	5, 2550, 2584, 2584
<code>\setCJKmainfont</code>	5, 2546, 2563, 2563, 2568, 2579
<code>\setCJKmathfont</code>	7, 2580, 2766, 2766
<code>\setCJKmonofont</code>	5, 2548, 2563, 2574, 2582
<code>\setCJKromanfont</code>	2568, 2583
<code>\setCJKsansfont</code>	5, 2547, 2563, 2569, 2581
<code>\SetSymbolFont</code>	2783
<code>\sfdefault</code>	2719, 3211
<code>\shapedefault</code>	2782, 2784
<code>\silent</code>	3034
<code>\skip_add:Nn</code>	1112
<code>\skip_horizontal:N</code>	860, 1116, 2892, 3437, 3444, 3522, 3571
<code>\skip_horizontal:n</code>	835, 2894, 3066, 3705, 3712
<code>\skip_if_eq:nnTF</code>	133, 142, 147, 164
<code>\skip_if_eq_p:nn</code>	595, 663, 664, 919
<code>\skip_if_finite_p:n</code>	918
<code>\skip_new:N</code>	47, 724, 752, 2930
<code>\skip_set:Nn</code>	189, 2871, 2897, 2912, 2933, 3713
<code>\skip_set_eq:NN</code>	1110, 3708
<code>\skip_use:N</code>	189, 2871, 2947
<code>\skip_vertical:n</code>	3774
<code>\skip_zero:N</code>	2921, 2937
<code>\SlantedFeatures</code>	2094
<code>\SlantedFont</code>	2094
<code>\SlantFactor</code>	2013
<code>\SplitArgument</code>	467
<code>\str_case:nnn</code>	3208
<code>\str_case_x:nnn</code>	1245, 2544, 2555, 2716
<code>\str_case_x:non</code>	1239
<code>\str_if_eq:nnF</code>	458, 566, 586, 632, 2435
<code>\str_if_eq:nnTF</code>	34, 2256, 3381
<code>\str_if_eq:nnTF</code>	14, 562, 1357, 2053, 2281, 2438, 3157, 3489
<code>\str_if_eq_p:nn</code>	1830, 1831, 2275, 2288, 2290
<code>\str_if_eq_x:nnTF</code>	94, 1968, 2732, 2967
<code>\str_if_eq_x_p:nn</code>	3314, 3315
<code>\sw@slant</code>	3222
T	
<code>\tabular</code>	3809, 3815, 3822
<code>\tex_baselineskip:D</code>	3066
<code>\tex_font:D</code>	122, 126, 135, 136, 137, 145, 151, 152, 157, 158, 167, 1821, 2926, 2934, 2941, 2987, 2995, 3002
<code>\tex_fontdimen:D</code>	135, 136, 137, 145, 151, 152, 157, 158, 167, 2934, 2941, 2987, 2995, 3002
<code>\tex_futurelet:D</code>	4031
<code>\tex_global:D</code>	2802
<code>\tex_hfill:D</code>	3793, 3801
<code>\tex_hss:D</code>	3778, 4047
<code>\tex_iftrue:D</code>	3358
<code>\tex_ignorespaces:D</code>	85, 810, 814, 2503, 2595, 2624, 3125, 3732, 3754
<code>\tex_italiccorrection:D</code>	624, 625
<code>\tex_kern:D</code>	706, 707, 837, 3777
<code>\tex_lastkern:D</code>	695, 3711
<code>\tex_lastpenalty:D</code>	924
<code>\tex_lastskip:D</code>	189, 595, 663, 664, 918, 919, 1110, 1112
<code>\tex_let:D</code>	3343
<code>\tex_noindent:D</code>	3845
<code>\tex_number:D</code>	182
<code>\tex_output:D</code>	2883, 2888, 3728, 3759, 3760, 3848, 3853
<code>\tex_penalty:D</code>	57, 58, 928, 929, 3436, 3443
<code>\tex_rightskip:D</code>	2871, 2897
<code>\tex_romannumeral:D</code>	221, 247
<code>\tex_space:D</code>	3435, 3442
<code>\tex_spacefactor:D</code>	623
<code>\tex_spaceskip:D</code>	133, 139, 142, 162, 167, 171, 172
<code>\tex_the:D</code>	122, 522, 2888, 3759, 3853
<code>\tex_unkern:D</code>	710
<code>\tex_unpenalty:D</code>	926
<code>\tex_unskip:D</code>	598, 667, 1113
<code>\tex_vrule:D</code>	822
<code>\tex_vss:D</code>	3780
<code>\tex_xspaceskip:D</code>	147, 155, 164, 169
<code>\textcentereddot</code>	3301
<code>\textellipsis</code>	3301
<code>\textemdash</code>	3301
<code>\textperiodcentered</code>	3301
<code>\textquotedblleft</code>	3302
<code>\textquotedblright</code>	3302
<code>\textquoteleft</code>	3302
<code>\textquoteright</code>	3302
<code>\textvisiblespace</code>	2976
<code>\tipaencoding</code>	3308
<code>\tl_case:Nnn</code>	1067
<code>\tl_case:Non</code>	1063, 1225
<code>\tl_clear:c</code>	2136
<code>\tl_clear:N</code>	1096, 1286, 1997
<code>\tl_concat:NNN</code>	1250, 1257
<code>\tl_const:cn</code>	442
<code>\tl_const:cx</code>	1491, 1493, 2946
<code>\tl_const:Nn</code>	129, 1402, 1403, 1875, 2768, 3400, 3406
<code>\tl_const:Nx</code>	64, 2780, 3093
<code>\tl_gput_right:Nn</code>	69, 71, 73
<code>\tl_gset:cn</code>	23, 1398, 1401
<code>\tl_gset:Nn</code>	2736, 3017, 3107, 3108, 3109, 3389, 3390
<code>\tl_gset:Nx</code>	881, 891, 904, 945, 957, 966, 988, 997, 1006, 1031, 2714, 3022, 3112
<code>\tl_gset_eq:cc</code>	24, 25, 26, 27
<code>\tl_gset_eq:NN</code>	3121
<code>\tl_gset_rescan:Nnn</code>	2742
<code>\tl_if_blank:nTF</code>	1559, 2046, 2278
<code>\tl_if_blank:VTF</code>	2187
<code>\tl_if_blank:vTF</code>	2269
<code>\tl_if_empty:NF</code>	540, 546
<code>\tl_if_empty:NTF</code>	2067, 2083
<code>\tl_if_empty:nTF</code>	262, 271, 3163
<code>\tl_if_eq:NNT</code>	2712
<code>\tl_if_eq:NNTF</code>	1792, 1848
<code>\tl_if_exist:cF</code>	1790, 1846
<code>\tl_if_exist:cTF</code>	61, 2910
<code>\tl_if_exist:NF</code>	3107, 3108, 3109
<code>\tl_if_exist:NTF</code>	3116
<code>\tl_if_exist_p:c</code>	1681
<code>\tl_if_head_is_N_type_p:n</code>	104

\xeCJK_cs_case_keys_define:nNNnn	\xeCJK_if_last_node_p:n
..... 1070 , 1070 , 1100 , 1246 , 1253 601 , 602 , 609 , 610 , 646 , 654 , 670 , 672 , 678 , 679 , 693
\xeCJK_cs_clear:N	\xeCJK_if_package_loaded:n
. 108 , 108 , 3333 , 3334 , 3335 , 3429 , 3430 , 3463 , 3466 , 4046 59
\xeCJK_cs_gclear:N	\xeCJK_if_package_loaded:nF
108 , 110 3040 , 3047 , 3363
\xeCJK_declare_char_class:nn	\xeCJK_if_package_loaded:nT
451 , 451 , 469 , 1348 3191 , 3320 , 3328 , 3368 , 3376
\xeCJK_declare_char_class:nV	\xeCJK_if_package_loaded:nTF
..... 505 , 506 , 507 , 508 , 512 , 513 , 514 59 , 74 , 3052 , 3175 , 3181 , 3365
\xeCJK_declare_char_class:nx	\xeCJK_if_package_loaded_p:n
448 , 1326 59 , 3342
\xeCJK_declare_sub_char_class:nnn	\xeCJK_if_same_class:NN
1340 , 1340 , 1350 282
\xeCJK_declare_sub_char_class:nxx	\xeCJK_if_same_class:NNTF
1297 282 , 1723
\xeCJK_def_node:nn	\xeCJK_if_same_class_p:NN
698 , 698 , 711 , 712 , 713 , 714 , 715 282
\xeCJK_Default_and_FullLeft:nN	\xeCJK_if_ulem_patch:TF
873 , 873 3509 , 3509 , 3519 , 3529 , 3538 , 3544 , 3561 , 3578 , 3589 , 3601 , 3615 , 3632 , 3642 , 3660 , 3669 , 3682 , 3697 , 3703 , 3737
\xeCJK_Default_and_FullRight:nN	\xeCJK_ignore_spaces:w
937 , 937 764 , 766 , 766 , 2863 , 2880 , 2896 , 3251 , 3788
\xeCJK_fallback_loop:Nn	\xeCJK_int_until_do:nn
1953 , 1957 , 1957 , 1965 , 1979 200 , 200 , 486 , 2800
\xeCJK_fallback_test_glyph:N	\xeCJK_inter_class_toks:nnc
1932 , 1935 , 1940 , 1946 , 1946 803 , 818
\xeCJK_family_if_exist:x	\xeCJK_inter_class_toks:nnn
2477 518 , 518 , 520 , 524 , 573 , 581 , 585 , 621 , 635 , 760 , 791 , 793 , 795 , 806 , 808 , 811 , 3491 , 3493 , 3497 , 3874 , 3876 , 3883 , 3888 , 3890 , 3892 , 3896 , 3902 , 3904 , 3906 , 3910
\xeCJK_family_if_exist:xF	\xeCJK_inter_class_toks:nnx
2450 , 2729 527 , 533 , 541 , 549 , 801
\xeCJK_family_if_exist:xF	\xeCJK_make_node:n
2378 , 2392 625 , 628 , 629 , 698 , 704 , 763 , 777 , 1218 , 1231 , 3230 , 3236 , 3242 , 3532 , 3787
\xeCJK_family_if_exist:xF	\xeCJK_new_class:n
..... 1959 , 2477 , 2494 , 2507 , 2520 , 2735 , 2771 , 2774 300 , 300 , 333 , 334 , 335 , 336 , 1344
\xeCJK_family_if_exist_use:x	\xeCJK_new_sub_key:n
2489 , 2492 , 2518 , 2518 1346 , 2038 , 2039 , 2093
\xeCJK_font_gset_to_current:c	\xeCJK_no_break:
..... 119 , 119 , 2419 , 2428 , 2460 , 2988 57 , 58 , 650 , 844 , 847 , 854 , 857 , 976 , 984 , 993 , 998 , 1012 , 1032 , 1101 , 1115 , 1217 , 1230 , 3622 , 3648 , 3653
\xeCJK_fontspec:nn	\xeCJK_peek_after_ignore_spaces:nw
2594 , 2597 , 2597 , 2614 236 , 236 , 1069
\xeCJK_fontspec:VV	\xeCJK_peek_catcode_ignore_spaces:NNTF
2663 206 , 206 , 768 , 1159
\xeCJK_FullLeft_and_CJK:	\xeCJK_pre_inter_class_toks:nnn
794 , 839 , 839 , 3471 525 , 525 , 530 , 633 , 1359 , 1373 , 1375 , 1382
\xeCJK_FullLeft_and_Default:	\xeCJK_pre_inter_class_toks:nnx
849 , 849 , 3470 , 3662 3279
\xeCJK_FullLeft_and_FullLeft:N	\xeCJK_punct_kern:NN
982 , 982 1005 , 1014 , 1410 , 1415
\xeCJK_FullLeft_and_FullRight:N	\xeCJK_punct_kerning_process:NN
991 , 991 1561 , 1672 , 1672
\xeCJK_FullRight_and_Boundary:	\xeCJK_punct_margin_process:NN
..... 810 , 812 , 813 , 813 , 1040 , 1042 , 1044 , 1050 , 1052 1560 , 1563 , 1563
\xeCJK_FullRight_and_CJK:	\xeCJK_remove_node:
796 , 861 , 861 , 3473 604 , 612 , 615 , 647 , 650 , 657 , 674 , 682 , 709 , 776 , 3229 , 3235 , 3241 , 3522 , 3523
\xeCJK_FullRight_and_Default:	\xeCJK_replace_inter_class_toks:nnnn
814 , 867 , 867 , 3472 543 , 543 , 1363
\xeCJK_FullRight_and_FullLeft:N	\xeCJK_reverse:nnn
1001 , 1001 93 , 93 , 1569 , 2315
\xeCJK_FullRight_and_FullRight:N	\xeCJK_save_class:nn
1026 , 1026 311 , 311 , 328 , 329 , 330 , 331 , 332
\xeCJK_FullRight_symbol:N	\xeCJK_select_font:
... 948 , 960 , 969 , 999 , 1033 , 1043 , 1045 , 1053 , 1058 , 1058 576 , 640 , 878 , 902 , 942 , 955 , 1799 , 2410 , 2413 , 2432 , 2439 , 2924 , 3557
\xeCJK_get_inter_class_toks:nn	\xeCJK_select_font:x
521 , 521 , 528 , 534 , 539 , 545 , 3278 , 3280 , 3282 , 3288 , 3884 1961 , 2410 , 2422
\xeCJK_get_punct_bounds:NN	\xeCJK_set_char_class:nnn
875 , 889 , 899 , 939 , 952 , 964 , 985 , 994 , 1003 , 1028 , 1061 , 1788 , 1788 , 3646 460 , 464 , 482 , 482 , 3147
\xeCJK_get_punct_kerning:NN	\xeCJK_set_family:nnn
1844 , 1844 , 1856 2153 , 2153 , 2176 , 2178 , 2611
\xeCJK_get_punct_kerning:oN	\xeCJK_set_family:Voo
986 , 995 , 1004 , 1029 2354
\xeCJK_glue_to_skip:nN	\xeCJK_set_family:VVV
186 , 186 , 721 , 730 , 747 , 3433 , 3440 2003
\xeCJK_glyph_bounds:NN	\xeCJK_set_family:xxx
1812 , 1814 , 1839 , 1839 2177 , 2565 , 2571 , 2576 , 2585 , 2590 , 2767
\xeCJK_glyph_if_exist:N	\xeCJK_set_family_fallback:nnn
124 1990 , 1991 , 1991
\xeCJK_glyph_if_exist:NNTF	\xeCJK_set_mathfont:
124 , 1948 , 1963 2749 , 2769 , 2769
\xeCJK_glyph_if_exist_p:N	\xeCJK_set_visible_space_font:
124 2975 , 2983 , 2983
\xeCJK_gset_mathcode:nnnn	\xeCJK_space_or_xecglue:
2792 , 2797 , 2797 615 , 736 , 741 , 748 , 771 , 784
\xeCJK_hook_for_ulem:	\xeCJK_swap_cs:NN
3419 , 3420 , 3420 112 , 112 , 3506 , 3739 , 4091
\xeCJK_if_blank_x:n	\xeCJK_switch_family:n
191 2505 , 2589 , 2612 , 2748 , 3210 , 3211 , 3212 , 3213
\xeCJK_if_blank_x:nT	\xeCJK_tl_remove_outer_braces:N
3281 , 3287 95 , 95
\xeCJK_if_blank_x:nTF	\xeCJK_tl_remove_outer_braces:n
191 , 474 , 1578 , 2318 , 2486 95 , 96 , 97 , 105 , 2063 , 2082
\xeCJK_if_blank_x_p:n	\xeCJK_token_value_charcode:N
191 , 472 , 1147 253 , 254 , 254 , 1821 , 1842
\xeCJK_if_CJK_class:N	\xeCJK_token_value_class:N
274 252 , 252 , 276 , 285 , 1468
\xeCJK_if_CJK_class:NNTF	
274 , 1205 , 1207	
\xeCJK_if_CJK_class_p:N	
274	
\xeCJK_if_last_node:n	
693	
\xeCJK_if_last_node:nT	
614 , 3523	
\xeCJK_if_last_node:nTF	
649 , 693 , 3227 , 3233 , 3239 , 3521	

\xeCJK_visible_space_fallback:	2972 , 2972 , 2981	\xeCJKRestoreSubCJKBlock	9 , 1301 , 1311
\xeCJK_xetex_mathcode:w	267 , 272 , 2802	\xeCJKsetcharclass	3145 , 3145
\xeCJKactive	288	\xeCJKsetecglue	3134
\xeCJKallowbreakbetweenpuncts	3137 , 3137	\xeCJKsetemboldenfactor	3127 , 3127
\xeCJKCancelSubCJKBlock	9 , 1301 , 1302	\xeCJKsetkern	10 , 1400 , 1400
\xeCJKcaption	3345 , 3347	\xeCJKsetslantfactor	3127 , 3129
\xeCJKDeclareCharClass	9 , 446 , 446	\xeCJKsetup .	2 , 1395 , 2890 , 3105 , 3122 , 3122 , 3128 , 3130 , 3131 , 3132 , 3133 , 3135 , 3136 , 3138 , 3140 , 3142 , 3144 , 3425
\xeCJKDeclarePunctStyle	10 , 1882 , 1882 , 1895 , 1903 , 1904 , 1905 , 1912 , 1920	\xeCJKsetwidth	10 , 1397 , 1397
\xeCJKDeclareSubCJKBlock	9 , 1294 , 1294 , 1300 , 1337	\xeCJKVerbAddon	12 , 2826 , 2833 , 2837 , 2837 , 2907 , 2963
\xeCJKdisablefallback	3141 , 3143	\xetex_if_engine:F	10
\xeCJKEditPunctStyle	10 , 1896 , 1896 , 1902	\xetex_XeTeXversion:D	256
\xeCJKenablefallback	3141 , 3141	\XeTeXcharclass	253 , 488 , 496 , 501 , 2904
\xeCJKnobreak	12 , 1102 , 1102	\XeTeXcharglyph	1842
\xeCJKnobreakbetweenpuncts	3137 , 3139	\XeTeXdefaultencoding	3349 , 3356
\xeCJKOffVerbAddon	12 , 2837 , 2849 , 2864 , 2886	\XeTeXglyphbounds	1388 , 1392 , 1841
\xeCJKplainchr	3131 , 3132	\XeTeXinterchartokenstate	295 , 296 , 3225
\xeCJKResetCharClass	9 , 510 , 510 , 517	\XeTeXinterchartoks	519 , 522
\xeCJKResetPunctClass	9 , 449 , 503 , 503 , 515 , 1298 , 1308 , 1317 , 3148	\XeTeXmathcode	272
		\XeTeXrevision	256