

# xeCJK 宏包

ctex.org

2012/10/27 v3.0.10

## 目录

<b>1</b>	<b>简介</b>	<b>2</b>
<b>2</b>	<b>基本用法</b>	<b>2</b>
<b>3</b>	<b>用户手册</b>	<b>3</b>
3.1	宏包选项	3
3.2	字体设置与选择	5
3.2.1	X <sub>g</sub> T <sub>E</sub> X 的字体名查找	8
3.3	CJK 分区字体设置	9
3.4	设置 CJK 字符范围	10
3.5	设置标点宽度和间距	11
<b>4</b>	<b>已知问题和兼容性</b>	<b>11</b>
<b>5</b>	<b>xeCJK 代码实现</b>	<b>11</b>
5.1	运行环境检查	11
5.2	内部工具	12
5.3	功能开关	15
5.4	字符类别设定	15
5.5	字符输出规则	22
5.6	段末孤字处理	30
5.7	增加 CJK 子分区	33
5.8	标点处理	36
5.9	后备字体	44
5.10	CJK 字体族声明方式	45
5.11	字体切换	54
5.12	数学字体设置	59
5.13	处理等宽字体和抄录环境	60
5.14	xeCJK 其它选项	62
5.15	xeCJK 初始化设置	63
5.16	兼容性修补	64

<b>6</b>	<b>例子</b>	<b>68</b>
6.1	xeCJK-example-autofake.tex	68
6.2	xeCJK-example-fallback.tex	69
6.3	xeCJK-example-subCJKblock.tex	70
6.4	xeCJK-example-CJKecglue.tex	71
6.5	xeCJK-example-checksingle.tex	72

## 1 简介

xeCJK 是一个  $\text{\LaTeX}$  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  $\text{\LaTeX}$  的一些最新特性,需要  $\text{\LaTeX}$  0.9995.0 [2009/06/29] 以后的版本。xeCJK 依赖  $\text{\LaTeX}$  3 项目的 expl3, xparse 和 l3keys2e 宏包。xeCJK 还通过 fontspec<sup>1</sup> 宏包来调用系统字体。个别兼容性功能需要 CJK 宏包的支持。

xeCJK 的原始作者是孙文昌,2009 年 5 月起宏包被收入 ctex-kit<sup>2</sup> 项目进行维护,目前主要维护者是刘海洋<sup>3</sup> 和李清<sup>4</sup>。

## 2 基本用法

与其他  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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<sup>5</sup>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。

<sup>1</sup><http://mirrors.ctan.org/macros/latex/contrib/fontspec/>

<sup>2</sup><http://code.google.com/p/ctex-kit/>

<sup>3</sup>[leoliu.pku@gmail.com](mailto:leoliu.pku@gmail.com)

<sup>4</sup>[sobenlee@gmail.com](mailto:sobenlee@gmail.com)

<sup>5</sup><http://mirrors.ctan.org/language/chinese/ctex/>

## 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。

---

<code>\xeCJKsetup</code>	<code>\xeCJKsetup {<math>\langle key1 \rangle = \langle var1 \rangle</math>, <math>\langle key2 \rangle = \langle var2 \rangle</math>, ...}</code>
--------------------------	--

---

其中  $\langle key1 \rangle$ ,  $\langle key2 \rangle$  是设置选项, 而  $\langle val1 \rangle$ ,  $\langle val2 \rangle$  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。

下列选项使用加粗的形式来表示 xeCJK 的默认设置。

---

<code>xeCJKactive</code>	<code>xeCJKactive = <math>\langle true   false \rangle</math></code>
--------------------------	--

---

打开/关闭对中文的特殊处理。

---

<code>CJKnumber <math>\star</math></code>	<code>CJKnumber = <math>\langle true   false \rangle</math></code>
---	--

---

是否使用 CJKnumb 宏包。请注意, 为了与 CJKnumb 兼容, xeCJK 修改了它的一些内部命令, 所以这个选项需要在调用 xeCJK 时设置, 而不能通过 `\xeCJKsetup` 来设置它。

---

<code>indentfirst <math>\star</math></code>	<code>indentfirst = <math>\langle true   false \rangle</math></code>
---	--

---

跟在章节标题后面的第一段首行是否缩进。这个选项需要在调用 xeCJK 时设置, 不能通过 `\xeCJKsetup` 来设置它。xeCJK 宏包中只有 indentfirst 和 CJKnumber 这两个选项, 不能通过 `\xeCJKsetup` 来设置。

---

<code>CJKspace</code>	<code>CJKspace = <math>\langle true   false \rangle</math></code>
-----------------------	---

---

缺省状态下, xeCJK 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。

---

<code>CJKmath <math>\star</math></code>	<code>CJKmath = <math>\langle true   false \rangle</math></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。

---

<code>CJKglue</code>	<code>CJKglue = {\hspace{0pt plus .08\baselineskip}}</code>
----------------------	---

---

设置 CJK 文字之间插入的 glue, 上边是 xeCJK 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 glue 最好有一定的弹性。

<hr/> CJKe glue <hr/>	CJKe glue = {\glue} <p>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 glue 最好也要用一定的弹性。请注意,这里设置的 glue 只影响 xeCJK 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 xeCJK 可能不能正确地调整间距,需要手动加空格。</p>
<hr/> xCJKe glue <hr/>	xCJKe glue = {\true false glue} <p>缺省状态下,xeCJK 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 CJKe glue 替换源文件中直接输入的 CJK 文字与西文之间的空格。</p>
<hr/> CheckSingle <hr/>	CheckSingle = {\true false} <p>是否避免单个 CJK 文字单独占一个段落的最后一行。</p>
<hr/> InlineEnv InlineEnv+ InlineEnv- <hr/>	InlineEnv = {env1, env2, env3, ...} <p>在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 L<sup>A</sup>T<sub>E</sub>X 环境的开始 \begin... 和结束 \end... 视为断行的地方,如果有某些特殊的 L<sup>A</sup>T<sub>E</sub>X 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。</p>
<hr/> AutoFallBack <hr/>	AutoFallBack = {\true false} <p>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。</p>
<hr/> AutoFakeBold ☆ <hr/>	AutoFakeBold = {\true false 数字} <p>全局设定当没有声明对应的粗体时,是否使用<b>伪粗体</b>;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。</p>
<hr/> AutoFakeSlant ☆ <hr/>	AutoFakeSlant = {\true false 数字} <p>全局设定当没有声明对应的斜体时,是否使用<b>伪斜体</b>;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。</p>
<hr/> EmboldenFactor ☆ <hr/>	EmboldenFactor = {\数字 4} <p>设置伪粗体的默认粗细程度。</p>
<hr/> SlantFactor ☆ <hr/>	SlantFactor = {\数字 0.167} <p>设置伪斜体的粗细程度,范围是 -0.999 ~ 0.999。</p>
<hr/> PunctStyle <hr/>	PunctStyle = {\quanjiao banjiao kaiming hangmoban CCT plain} <p>设置标点处理格式,有效值分别为</p> <p>quanjiao 全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;</p> <p>banjiao 半角式: 所有标点占半个汉字宽度;</p> <p>kaiming 开明式: 句末点号用全角,其他半角;</p> <p>hangmoban 行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</p> <p>CCT CCT 格式;</p> <p>plain 原样(不调整标点间距)。</p>

<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 = {\dimen}
<hr/>		
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 <code>dimen</code> 的单位最好用 <code>em</code> 等相对距离单位,而不建议使用诸如 <code>pt</code> 之类的绝对距离单位。这里的设置对除了 <code>plain</code> 以外的所有标点处理格式都有效。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 节的 <code>\xeCJKsetwidth</code> 。		

<hr/> AllowBreakBetweenPuncts		AllowBreakBetweenPuncts = {true false}
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。		

## 3.2 字体设置与选择

<hr/> \setCJKmainfont	★	\setCJKmainfont [⟨font features⟩] {⟨font name⟩}
设置正文罗马族的 CJK 字体,影响 \rmfamily 和 \textrm 的字体。后面两个参数继承自 fontspec 宏包,⟨font features⟩ 表示字体属性选项,⟨font name⟩ 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 fontspec 宏包的文档。需要说明的是 xeCJK 修改了 AutoFakeBold 和 AutoFakeSlant 选项,以便配合全局伪粗体和伪斜体的设定。		
<hr/>		
AutoFakeBold		AutoFakeBold = {⟨true false 数字⟩}
AutoFakeSlant		AutoFakeSlant = {⟨true false 数字⟩}
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。		
<hr/>		
\setCJKsansfont	★	\setCJKsansfont [⟨font features⟩] {⟨font name⟩}
设置正文无衬线族的 CJK 字体,影响 \sffamily 和 \textsf 的字体。		
<hr/>		
\setCJKmonofont	★	\setCJKmonofont [⟨font features⟩] {⟨font name⟩}
设置正文等宽族的 CJK 字体,影响 \ttfamily 和 \texttt 的字体。为了有利于等宽字体的代码对齐等情形,xeCJK 在 ⟨font features⟩ 里增加了 Mono 这个选项。		

<hr/> <b>Mono</b> <hr/>	<p><code>Mono = &lt;Scale Exspace&gt;</code></p> <p>有效值仅为 <code>Scale</code> 或 <code>Exspace</code>。分别表示使用放缩的方式或改变间距的方式,让 CJK 字符的宽度等于西文字体中两个空格的宽度。</p>
<hr/> <code>\CJKfixedspacing</code> <hr/> <code>\CJKflexiblespacing</code> <hr/>	<p><code>\CJKfixedspacing</code> <code>\CJKflexiblespacing</code></p> <p><code>\CJKfixedspacing</code> 命令设置固定字距和不压缩标点格式,符合上面 <code>Mono=Exspace</code> 选项的意义。<code>\CJKflexiblespacing</code> 命令恢复可伸缩的字距和标点压缩。抄录 (verbatim) 环境将自动使用 <code>\CJKfixedspacing</code> 的效果。</p>
<hr/> <code>\setCJKmonoscale</code> <hr/> <code>\setCJKmonoexspace</code> <hr/>	<p><code>\setCJKmonoscale</code> <code>\setCJKmonoexspace</code></p> <p><code>\setCJKmonoscale</code> 和 <code>\setCJKmonoexspace</code> 命令分别计算并设置当前字号下西文等宽字体与 CJK 字体的放缩比例和字距,以符合上面所说的 <code>Mono=Scale</code> 与 <code>Mono=Exspace</code> 选项的意义。在字号改变时,应该使用两个命令之一,重新计算相关参数;但注意两个命令不能同时使用。</p>
<hr/> <code>\setCJKfamilyfont</code> ★ <hr/>	<p><code>\setCJKfamilyfont &lt;{family}&gt; [&lt;{font features}&gt;] &lt;{font name}&gt;</code></p> <p>声明新的 CJK 字体族 <i>&lt;family&gt;</i> 并指定字体。</p>
<hr/> <code>\CJKfamily</code> <hr/>	<p><code>\CJKfamily &lt;{family}&gt;</code>  <code>\CJKfamily+ &lt;{family}&gt;</code>  <code>\CJKfamily- &lt;{family}&gt;</code></p> <p>用于在文档中切换 CJK 字体族,<i>&lt;family&gt;</i> 必须预先声明。<code>\CJKfamily</code> 仅对 CJK 字符类有效,<code>\CJKfamily+</code> 对所有字符类均有效,<code>\CJKfamily-</code> 对非 CJK 字符类有效。当 <code>\CJKfamily+</code> 和 <code>\CJKfamily-</code> 的参数为空时,则使用当前的 CJK 字体族。</p>
<hr/> <code>\newCJKfontfamily</code> ★ <hr/>	<p><code>\newCJKfontfamily [&lt;{family}&gt;] \&lt;{font-switch}&gt; [&lt;{font features}&gt;] &lt;{font name}&gt;</code></p> <p>声明新的 CJK 字体族 <i>&lt;family&gt;</i> 并指定字体,并定义 <code>\&lt;{font-switch}&gt;</code>,在文档中可以使用它来切换 CJK 字体族。可以不必指定 <i>&lt;family&gt;</i>,这时候 <i>&lt;family&gt;</i> 将等于 <i>&lt;font-switch&gt;</i>。事实上,<code>\newCJKfontfamily</code> 是 <code>\setCJKfamilyfont</code> 和 <code>\CJKfamily</code> 的合并。例如</p> <p style="text-align: center;"><code>\newCJKfontfamily[song]\songti{SimSun}</code></p> <p>等价于</p> <p style="text-align: center;"><code>\setCJKfamilyfont{song}{SimSun}</code>  <code>\newcommand\songti{\CJKfamily{song}}</code></p>
<hr/> <code>\CJKfontspec</code> <hr/>	<p><code>\CJKfontspec [&lt;{font features}&gt;] &lt;{font name}&gt;</code></p> <p>在文档中随机定义新的 CJK 字体族,并马上使用它。</p>

<hr/> <code>\defaultCJKfontfeatures</code> ☆ <hr/>	<code>\defaultCJKfontfeatures {&lt;font features&gt;}</code> 全局设置 CJK 字体族的默认选项。例如, 使用 <code>\defaultCJKfontfeatures{Scale=0.962216}</code> 可以将全部 CJK 字体缩小为 0.962216。xeCJK 宏包的初始化设置是 <code>\defaultCJKfontfeatures{Script=CJK}</code>
<hr/> <code>\addCJKfontfeatures</code> <hr/>	<code>\addCJKfontfeatures {&lt;font features&gt;}</code> 临时增加当前使用的 CJK 字体的选项。例如, 使用 <code>\addCJKfontfeatures{Scale=1.1}</code> 可以将文档中当前使用的 CJK 字体放大为 1.1。
<hr/> <code>\CJKrmdefault</code> <hr/>	保存 <code>\rmfamily</code> 所使用的 CJK 字体族, 默认值是 <code>rm</code> 。类似西文字体的 <code>\rmdefault</code> 。
<hr/> <code>\CJKsfdefault</code> <hr/>	保存 <code>\sffamily</code> 所使用的 CJK 字体族, 默认值是 <code>sf</code> 。类似西文字体的 <code>\sfdefault</code> 。
<hr/> <code>\CJKttdefault</code> <hr/>	保存 <code>\ttfamily</code> 所使用的 CJK 字体族, 默认值是 <code>tt</code> 。类似西文字体的 <code>\ttdefault</code> 。
<hr/> <code>\CJKfamilydefault</code> <hr/>	保存 <code>\normalfont</code> 所使用的 CJK 字体族, 默认值是 <code>\CJKrmdefault</code> 。类似西文字体的 <code>\familydefault</code> 。可以通过更改它来改变文档的默认 CJK 字体族。例如, 使用 <code>\renewcommand\familydefault{\sfdefault}</code> <code>\renewcommand\CJKfamilydefault{\CJKsfdefault}</code> 可以将全文的 CJK 和西文默认字体改为无衬线字体族。
<hr/> <code>\setCJKmathfont</code> ★ <hr/>	<code>\setCJKmathfont [&lt;font features&gt;] {&lt;font name&gt;}</code> 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 <code>\setCJKmathfont</code> 设置数学公式中的 CJK 字体, 那么将使用 <code>\CJKfamilydefault</code> 作为数学公式中的 CJK 字体。
<hr/> <code>\setCJKfallbackfamilyfont</code> ★ <hr/>	<code>\setCJKfallbackfamilyfont {&lt;family&gt;} [&lt;font features&gt;] {&lt;font name&gt;}</code> 设置 CJK 字体族 <code>&lt;family&gt;</code> 的备用字体。例如, 使用 <code>\setCJKmainfont{SimSun}</code> <code>\setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}</code> 可以将 SimSun-ExtB 作为 SimSun 的备用字体。

---

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

---

xeCJK 在 <font features> 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 features 1>]} {<font name 1>}} ,
  {[<font features 2>]} {<font name 2>}} ,
  .....
}
```

---

\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。

### 3.2.1 X<sub>Y</sub>TeX 的字体名查找

由于在 fontspec 宏包文档中缺少关于如何查看 X<sub>Y</sub>TeX 可用字体名的说明,这里略作说明。

X<sub>Y</sub>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系列表存入 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,
Ítalik,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,Negrito Itálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın Ítalik,
Krepmo 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,Normálne,Navadno,
    thường,Arrunta
宋体,SimSun:style=Regular
黑体,SimHei:style=Normal,obyčejné,Standard,Κανονικά,Regular,Normaali,
    Normál,Normale,Standaard,Normalny,Обычный,Normálne,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<sup>6</sup>。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。

<sup>6</sup>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。

---

`\xeCJKDeclareSubCJKBlock` ★ `\xeCJKDeclareSubCJKBlock {<block>} {<block range>}`  
`\xeCJKDeclareSubCJKBlock* {<block>} {<block range>}`

其中 `<block range>` 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

```
{ `中 -> `文 , "3400 -> "4DBF , "5000 -> "7000 , `汉 , `字 , "3500 }
```

的形式。需要注意的是, 这里设置的 `<block range>` 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

```
\xeCJKDeclareSubCJKBlock{SPUA}{ "E400 -> "E4DA , "E500 -> "E5E8 , "E600 -> "E6CE }
\xeCJKDeclareSubCJKBlock{Ext-B}{ "20000 -> "2A6DF }
```

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 `<font features>` 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。

例如, 可以使用

```
\setCJKmainfont[SPUA=SunmanPUA,Ext-B=SimSun-ExtB]{SimSun}
```

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。

`\xeCJKDeclareSubCJKBlock` 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 `<block>` 选项, 将使用 `\CJKfamilydefault` 的 `<block>` 选项作为该 CJK 字体族的 `<block>` 选项。如果希望在使用某 CJK 字体族时, 不在 CJK 主分区与 `<block>` 之间切换字体, 可以使用 `<block>=*` 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。

---

`\xeCJKCancelSubCJKBlock` `\xeCJKCancelSubCJKBlock {<block1,block2,...>}`  
`\xeCJKCancelSubCJKBlock* {<block1,block2,...>}`

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。

---

`\xeCJKRestoreSubCJKBlock` `\xeCJKRestoreSubCJKBlock {<block1,block2,...>}`  
`\xeCJKRestoreSubCJKBlock* {<block1,block2,...>}`

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。

### 3.4 设置 CJK 字符范围

---

`\xeCJKDeclareCharClass` ★ `\xeCJKDeclareCharClass {<class>} {<class range>}`  
`\xeCJKDeclareCharClass* {<class>} {<class range>}`

`<class range>` 的格式和 3.3 节的 `<block range>` 相同。`<class>` 的有效值见源代码(第 5.4 节)。`\xeCJK` 已经支持 Unicode 中所有 CJK 文字和标点。一般来说, 不要輕易改变字符类别。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。

---

`\xeCJKResetPunctClass` ★ `\xeCJKResetPunctClass`

用于重置标点符号所属的字符类。

---

`\normalspacedchars` `\normalspacedchars {<char list>}`

在 `<char list>` 中出现的字符两端不自动添加空格, 默认设置是 /。

### 3.5 设置标点宽度和间距

这里的设置对除了 `plain` 以外的所有标点处理格式都有效。

---

```
\xeCJKsetwidth ★ \xeCJKsetwidth {<标点列表>} {<dimen>}
```

---

<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 `0.7em`。

---

```
\xeCJKsetkern ★ \xeCJKsetkern {<标点 1>} {<标点 2>} {<dimen>}
```

---

`xeCJK` 会根据选定的标点处理格式自动调整相邻两个 CJK 标点的距离。如果对缺省配置不满意,可以使用此命令设置两个标点之间的距离。例如,

```
\xeCJKsetkern{:}{“}{0.3em}
```

## 4 已知问题和兼容性

为了与 `CJKnumb` 和 `CJKulem` 宏包兼容,`xeCJK` 使用并重新定义了 `CJK` 宏包的部分宏命令,如 `\CJKfamily`、`\CJKsymbol` 和 `\CJKpunctsymbol` 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 `CJKfntef` 宏包,以实现汉字加点等。

`xeCJK` 依赖  $\text{XeTeX}$  的 `\XeTeXinterchartoks` 机制,与使用相同机制的宏包(例如 `polyglossia`)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  $\text{XeTeX}$  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 } }
    应该使用较新版本的 expl3 宏包。
11 \msg_new:nnn { xeCJK } { l3-too-old }
12 {
13   Support~package~'expl3'~too~old. \\
14   Please~update~an~up~to~date~version~of~the~bundles\\
15   'l3kernel'~and~'l3packages'\\
```

```

16     using~your~TeX~package~manager~or~from~CTAN.\\
17     Loading~xeCJK~will~abort!
18 }
19 \@ifpackagelater { expl3 } { 2012/09/10 } { }
20 { \msg_critical:nn { xecjk } { l3-too-old } }
21 \RequirePackage{xparse}
22 \RequirePackage{l3keys2e}

    \XeTeXglyphbouds 可以得到一个字符的左右边距,用于标点压缩。
23 \msg_new:nnn { xecjk } { XeTeX-too-old }
24 {
25     \token_to_str:N \XeTeXglyphbouds \ is~not~defined.\\\
26     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
27 }
28 \cs_if_free:NT \XeTeXglyphbouds
29 {
30     \msg_error:nn { xecjk } { XeTeX-too-old }
31     \AtEndOfPackage { \__xecjk_AfterPreamble:n { \punctstyle { plain } } }
32 }

```

## 5.2 内部工具

分配临时变量。

```

\l__xecjk_tmpa_int
\l__xecjk_tmpp_int 33 \int_new:N \l__xecjk_tmpa_int
\l__xecjk_tmppc_int 34 \int_new:N \l__xecjk_tmpp_int
35 \int_new:N \l__xecjk_tmppc_int
(End definition for \l__xecjk_tmpa_int, \l__xecjk_tmpp_int, and \l__xecjk_tmppc_int)

\l__xecjk_tmpa_dim
\l__xecjk_tmpp_dim 36 \dim_new:N \l__xecjk_tmpa_dim
\l__xecjk_tmppc_dim 37 \dim_new:N \l__xecjk_tmpp_dim
38 \dim_new:N \l__xecjk_tmppc_dim
(End definition for \l__xecjk_tmpa_dim, \l__xecjk_tmpp_dim, and \l__xecjk_tmppc_dim)

\l__xecjk_tmpa_tl
\l__xecjk_tmpp_tl 39 \tl_new:N \l__xecjk_tmpa_tl
40 \tl_new:N \l__xecjk_tmpp_tl
(End definition for \l__xecjk_tmpa_tl and \l__xecjk_tmpp_tl)

\l__xecjk_tmpa_clist
\l__xecjk_tmpp_clist 41 \clist_new:N \l__xecjk_tmpa_clist
\l__xecjk_tmppc_clist 42 \clist_new:N \l__xecjk_tmpp_clist
43 \clist_new:N \l__xecjk_tmppc_clist
(End definition for \l__xecjk_tmpa_clist, \l__xecjk_tmpp_clist, and \l__xecjk_tmppc_clist)

\xecjk_no_break:
44 \cs_new_protected_nopar:Npn \xecjk_no_break: { \tex_penalty:D \c_ten_thousand }
(End definition for \xecjk_no_break:)

```

`\__xeCJK_AtEndPreamble:n` 在 `\document` 前后加上各种钩子。

`\__xeCJK_AfterPreamble:n`

`\__xeCJK_AfterEndPreamble:n`

```

45 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
46 \tl_new:N \g__xeCJK_after_preamble_hook_tl
47 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl
48 \cs_new_protected:Npn \__xeCJK_AtEndPreamble:n #1
49 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
50 \cs_new_protected:Npn \__xeCJK_AfterPreamble:n #1
51 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
52 \cs_new_protected:Npn \__xeCJK_AfterEndPreamble:n #1
53 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
54 \AtBeginDocument { \g__xeCJK_after_preamble_hook_tl }
55 \tl_put_left:Nn \document { \group_end: \g__xeCJK_at_end_preamble_hook_tl \group_begin: }
56 \tl_put_right:Nn \document { \g__xeCJK_after_end_preamble_hook_tl \tex_ignorespaces:D }
(End definition for \__xeCJK_AtEndPreamble:n, \__xeCJK_AfterPreamble:n, and \__xeCJK_AfterEndPreamble:n)

```

`\xeCJK_if_outer_macro:p:c` 判断是否是 `\outer` 宏,输入的是控制序列的名字。

`\xeCJK_if_outer_macro:cTF`

```

57 \group_begin:
58 \char_set_lccode:nn { \Z } { \t }
59 \char_set_lccode:nn { \Y } { \c }
60 \char_set_lccode:nn { \; } { \: }
61 \char_set_lccode:nn { \ / } { \ \ }
62 \tl_map_inline:nn { \T \F \C \J \K } { \char_set_lccode:nn { \#1 } { \#1 } }
63 \tl_map_function:nN { \O \U \Z \E \R \M \A \Y \R \; \ / } \char_set_catcode_other:N
64 \tl_to_lowercase:n
65 {
66   \group_end:
67   \prg_new_conditional:Npnn \xeCJK_if_outer_macro:c #1 { p , T , F , TF }
68   {
69     \exp_last_unbraced:Nf \__xeCJK_if_outer_macro_aux:w
70     { \cs_meaning:c {#1} } OUZER ~ MAYRO ; \q_stop
71   }
72   \cs_new_nopar:Npn \__xeCJK_if_outer_macro_aux:w #1 OUZER ~ MAYRO ; #2 \q_stop
73   {
74     \if_cs_exist:w c__xeCJK_prefix_#1_aux_tl \cs_end:
75     \prg_return_true: \else: \prg_return_false: \fi:
76   }
77   \tl_new:c { c__xeCJK_prefix_/_aux_tl }
78   \tl_new:c { c__xeCJK_prefix_/long/_aux_tl }
79   \tl_new:c { c__xeCJK_prefix_/protected/_aux_tl }
80   \tl_new:c { c__xeCJK_prefix_/protected/long/_aux_tl }
81 }
(End definition for \xeCJK_if_outer_macro:c)

```

`\xeCJK_peek_catcode_ignore_spaces:NTF` 对 `l3token` 打的补丁,使 `\xeCJK_peek_catcode_ignore_spaces:NTF` 等可用于 `\outer` 宏。

```

82 \token_new:Nn \l__xeCJK_peek_search_token { ? }
83 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
84 {
85   \cs_set_eq:NN \l__xeCJK_peek_search_token #1 \prg_do_nothing:
86   \tl_set:Nn \__xeCJK_peek_true:w { \group_align_safe_end: #2 }
87   \tl_set:Nn \__xeCJK_peek_false:w { \group_align_safe_end: #3 }
88   \group_align_safe_begin:
89   \peek_after:Nw \__xeCJK_peek_ignore_spaces_execute_branches:

```

```

90 }
91 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NT #1#2
92 { \xeCJK_peek_catcode_ignore_spaces:NTF #1 {#2} { } }
93 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NF #1#2
94 { \xeCJK_peek_catcode_ignore_spaces:NTF #1 { } {#2} }
95 \cs_new_nopar:Npn \__xeCJK_peek_ignore_spaces_execute_branches:
96 {
97   \if_meaning:w \l_peek_token \c_space_token
98     \tex_afterassignment:D \__xeCJK_peek_ignore_spaces_execute_branches_aux:
99     \exp_after:wN \cs_set_eq:NN \exp_after:wN \__xeCJK_peek_tmp:w
100   \else:
101     \exp_after:wN \__xeCJK_peek_execute_branches_catcode:
102   \fi:
103 }
104 \cs_new_protected_nopar:Npn \__xeCJK_peek_ignore_spaces_execute_branches_aux:
105 { \peek_after:Nw \__xeCJK_peek_ignore_spaces_execute_branches: }
106 \cs_new_nopar:Npn \__xeCJK_peek_execute_branches_catcode:
107 {
108   \if_catcode:w \exp_not:N \l_peek_token \exp_not:N \l__xeCJK_peek_search_token
109     \exp_after:wN \__xeCJK_peek_true:w
110   \else:
111     \exp_after:wN \__xeCJK_peek_false:w
112   \fi:
113 }

```

(End definition for \xeCJK\_peek\_catcode\_ignore\_spaces:NTF)

\xeCJK\_if\_blank\_x:p:n 判断是否为空或者仅含一个空格。

```

\xeCJK_if_blank_x:n:NTF 114 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
115 {
116   \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
117     \prg_return_true:
118   \else:
119     \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
120     \prg_return_true: \else: \prg_return_false: \fi:
121   \fi:
122 }

```

(End definition for \xeCJK\_if\_blank\_x:n)

\xeCJK\_if\_package\_loaded\_p:n 判断宏包是否载入,可用于文档中。

```

\xeCJK_if_package_loaded:n:NTF 123 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
124 {
125   \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_pkg_extension_tl }
126     { \prg_return_true: } { \prg_return_false: }
127 }
128 \tl_const:Nx \c__xeCJK_pkg_extension_tl { \@pkgextension }

```

(End definition for \xeCJK\_if\_package\_loaded:n)

\\_\_xeCJK\_msg\_new:nn 各种信息函数的缩略形式。

```

\__xeCJK_error:n 129 \cs_new_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:nx 130 \cs_new_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_warning:nx 131 \cs_new_nopar:Npn \__xeCJK_error:nx { \msg_error:nnx { xeCJK } }
\__xeCJK_info:nx 132 \cs_new_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\__xeCJK_info:nxx

```

```

133 \cs_new_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
134 \cs_new_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
135 \cs_new_nopar:Npn \__xeCJK_info:nx { \msg_info:nnx { xeCJK } }
136 \cs_new_nopar:Npn \__xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }
(End definition for \__xeCJK_msg_new:nn and others.)

```

### 5.3 功能开关

**xeCJKactive**

```

137 \keys_define:nn { xeCJK / options }
138 {
139   xeCJKactive .choice:,
140   xeCJKactive / true .code:n = { \makexeCJKactive },
141   xeCJKactive / false .code:n = { \makexeCJKinactive },
142   xeCJKactive .default:n = { true },
143 }
(End definition for xeCJKactive This function is documented on page 3.)

```

```

\makexeCJKactive
\makexeCJKinactive
144 \NewDocumentCommand \makexeCJKactive { } { \XeTeXinterchartokenstate = \c_one }
145 \NewDocumentCommand \makexeCJKinactive { } { \XeTeXinterchartokenstate = \c_zero }
(End definition for \makexeCJKactive and \makexeCJKinactive)
抑制 BOM。
146 \char_set_catcode_ignore:n { "FEFF }

```

### 5.4 字符类别设定

xeCJK 需要以下字符类别用于字符输出。其中 Default、CJK、FullLeft、FullRight、Boundary 为 Xe<sub>La</sub>TeX 中预定义的类别,xeCJK 新增加了 HalfLeft、HalfRight、NormalSpace。

类别	说明	例子
Default	西文一般符号	abc123
CJK	CJK 表意符号	汉字あいう
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

这五类是 Xe<sub>La</sub>TeX 预定义的类别。

```

\c__xeCJK_Default_class_int 这五类是 XeLaTeX 预定义的类别。
\c__xeCJK_CJK_class_int
\c__xeCJK_FullLeft_class_int
\c__xeCJK_FullRight_class_int
\c__xeCJK_Boundary_class_int
147 \int_const:Nn \c__xeCJK_Default_class_int { 0 }
148 \int_const:Nn \c__xeCJK_CJK_class_int { 1 }
149 \int_const:Nn \c__xeCJK_FullLeft_class_int { 2 }
150 \int_const:Nn \c__xeCJK_FullRight_class_int { 3 }
151 \int_const:Nn \c__xeCJK_Boundary_class_int { 255 }
(End definition for \c__xeCJK_Default_class_int and others.)

```

`\__xeCJK_new_class:n` 新建一个字符类别。  
`\g__xeCJK_class_seq`

```

152 \cs_new_protected_nopar:Npn \__xeCJK_new_class:n #1
153 {
154   \exp_args:Nc \newXeTeXintercharclass { c__xeCJK_#1_class_int }
155   \clist_gclear_new:c { g__xeCJK_#1_range_clist }
156   \seq_gput_right:Nv \g__xeCJK_class_seq { c__xeCJK_#1_class_int }
157 }
158 \seq_new:N \g__xeCJK_class_seq
(End definition for \__xeCJK_new_class:n and \g__xeCJK_class_seq)

```

`\c__xeCJK_HalfLeft_class_int` 新增西文半角左/右标点和前后原始间距的符号类。  
`\c__xeCJK_HalfRight_class_int`  
`\c__xeCJK_NormalSpace_class_int`

```

159 \__xeCJK_new_class:n { HalfLeft }
160 \__xeCJK_new_class:n { HalfRight }
161 \__xeCJK_new_class:n { NormalSpace }
(End definition for \c__xeCJK_HalfLeft_class_int, \c__xeCJK_HalfRight_class_int, and \c__xeCJK_NormalSpace_class_int)

```

`\g__xeCJK_base_class_clist`  
`\g__xeCJK_CJK_class_clist`  
`\g__xeCJK_CJK_class_prop`

```

162 \clist_set:Nn \g__xeCJK_base_class_clist
163 {
164   Default, CJK, FullLeft, FullRight, Boundary,
165   HalfLeft, HalfRight, NormalSpace
166 }
167 \clist_map_inline:Nn \g__xeCJK_base_class_clist
168 { \clist_gclear_new:c { g__xeCJK_#1_range_clist } }
169 \clist_new:N \g__xeCJK_CJK_class_clist
170 \prop_new:N \g__xeCJK_CJK_class_prop
171 \cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
172 {
173   \clist_gput_right:Nx \g__xeCJK_CJK_class_clist {#1}
174   \prop_gput:Nvx \g__xeCJK_CJK_class_prop { c__xeCJK_#1_class_int } {#1}
175 }
176 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
177 \clist_map_function:nN { CJK, FullLeft, FullRight } \__xeCJK_save_CJK_class:n
(End definition for \g__xeCJK_base_class_clist, \g__xeCJK_CJK_class_clist, and \g__xeCJK_CJK_class_prop)

```

`\__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。

```

178 \cs_new_nopar:Npn \__xeCJK_class_num:n #1 { \tl_use:c { c__xeCJK_#1_class_int } }
(End definition for \__xeCJK_class_num:n)

```

`\xeCJKDeclareCharClass` 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:Nn`

```

179 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
180 {
181   \__xeCJK_declare_char_class:nn {#2} {#3}
182   \IfBooleanT {#1} { \xeCJKResetPunctClass }
183 }
184 \cs_new_protected_nopar:Npn \__xeCJK_declare_char_class:nn #1#2
185 {
186   \clist_gput_right:cx { g__xeCJK_#1_range_clist } {#2}
187   \clist_map_inline:xn {#2}
188   {
189     \str_if_eq:x:nnF {#1} { -> }

```

```

190      {
191        \__xeCJK_set_char_class_aux:Nn \__xeCJK_set_char_class:nnn {##1}
192        { \__xeCJK_class_num:n {#1} }
193      }
194    }
195  }
196  \NewDocumentCommand \__xeCJK_set_char_class_aux:Nn
197    { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
198  \cs_generate_variant:Nn \clist_map_inline:nn { x }
  (End definition for \xeCJKDeclareCharClass This function is documented on page 10.)

```

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

```

199  \NewDocumentCommand \xeCJKsetcharclass { s m m m }
200  {
201    \__xeCJK_set_char_class:nnn {#2} {#3} {#4}
202    \IfBooleanF {#1} { \xeCJKResetPunctClass }
203  }
204  \cs_new_protected_nopar:Npn \__xeCJK_set_char_class:nnn #1#2#3
205  {
206    \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
207    \int_set:Nn \l__xeCJK_tmpc_int {#3}
208    \prop_if_in:NVTF \g__xeCJK_CJK_class_prop \l__xeCJK_tmpc_int
209    { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \char_set_catcode_other:n }
210    { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \use_none:n }
211    \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
212    {
213      \__xeCJK_set_char_catcode:n { \l__xeCJK_tmpa_int }
214      \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
215      \int_incr:N \l__xeCJK_tmpa_int
216    }
217  }
  (End definition for \xeCJKsetcharclass)

```

```

\__xeCJK_int_until_do:nn
\__xeCJK_int_until_do_aux:wn
218  \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
219  {
220    \__xeCJK_int_until_do_aux:wn \use_none:n
221    { \reverse_if:N \if_int_compare:w #1#2 }
222  }
223  \cs_new_protected:Npn \__xeCJK_int_until_do_aux:wn \use_none:n #1
224  {
225    #1 \exp_after:wN \__xeCJK_int_until_do_aux:wn \fi:
226    \use_none:n {#1}
227  }
  (End definition for \xeCJK_int_until_do:nn This function is documented on page ??.)

```

```

\__xeCJK_check_num_range:nnNN
228  \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
229  {
230    \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
231    {
232      \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }

```

```

233     \int_set_eq:NN #3 #4
234   }
235   {
236     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
237     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
238   }
239 }

```

(End definition for \\_\_xeCJK\_check\_num\_range:nnNN)

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

```

240 \NewDocumentCommand \xeCJKResetPunctClass { }
241 {
242   \__xeCJK_declare_char_class:nn { HalfLeft }
243   { "28 , "2D , "5B , "60 , "7B }
244   \__xeCJK_declare_char_class:nn { HalfRight }
245   { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , }
246   \__xeCJK_declare_char_class:nn { FullLeft }
247   {
248     "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
249     "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
250     "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
251     "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6 ,
252   }
253   \__xeCJK_declare_char_class:nn { FullRight }
254   {
255     "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2030 , "2500 ,
256     "3001 , "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 ,
257     "3015 , "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 ,
258     "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C ,
259     "309D , "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 ,
260     "30E5 , "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE ,
261     "FE50 , "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C ,
262     "FE5E , "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B ,
263     "FF1F , "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 ,
264     "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E ,
265     "FF9F ,
266   }
267 }

```

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

设置 CJK 文字的范围。

```

268 \__xeCJK_declare_char_class:nn { CJK }
269 {

```

- Hangul Jamo (谚文字母)

```

270     "1100 -> "11FF ,

```

- CJK Radicals Supplement (中日韩部首补充)

```

271     "2E80 -> "2EFF ,

```

- Kangxi Radicals (康熙字典部首)

```

272     "2F00 -> "2FDF ,

```

- Ideographic Description Characters (表意文字描述符)  
273 "2FF0 -> "2FFF ,
- CJK Symbols and Punctuation (中日韩符号和标点)  
274 "3000 -> "303F ,
- Hiragana (日文平假名)  
275 "3040 -> "309F ,
- Katakana (日文片假名)  
276 "30A0 -> "30FF ,
- Bopomofo (注音字母)  
277 "3100 -> "312F ,
- Hangul Compatibility Jamo (谚文兼容字母)  
278 "3130 -> "318F ,
- Kanbun (象形字注释标志)  
279 "3190 -> "319F ,
- Bopomofo Extended (注音字母扩展)  
280 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)  
281 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)  
282 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)  
283 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)  
284 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)  
285 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)  
286 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)  
287 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)  
288 "A000 -> "A48F ,

- Yi Radicals (彝文字根)  
289 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)  
290 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)  
291 "AC00 -> "D7AF ,
- Kana Supplement (日文假名增补)  
292 "B000 -> "B0FF ,
- Hangul Jamo Extended-B (谚文扩展 B)  
293 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)  
294 "F900 -> "FAFF ,
- CJK Compatibility Forms (中日韩兼容形式)  
295 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)  
296 "FF00 -> "FFEF ,
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)  
297 "20000 -> "2A6DF ,
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)  
298 "2A700 -> "2B73F ,
- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)  
299 "2B740 -> "2B81F ,
- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)  
300 "2F800 -> "2FA1F ,
- 301 }

重置标点符号的字符类。

302 \xeCJKResetPunctClass

**\normalspacedchars** 声明前后不加间距的字符。

```

303 \NewDocumentCommand \normalspacedchars { m }
304 {
305   \tl_map_inline:nn {#1}
306     { \XeTeXcharclass `##1 = \__xeCJK_class_num:n { NormalSpace } }
307 }
308 \normalspacedchars{/}

```

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

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

```

309 \cs_new_protected_nopar:Npn \__xeCJK_inter_class_toks:nnn #1#2#3
310 { \XeTeXinterchartoks \__xeCJK_class_num:n {#1} \__xeCJK_class_num:n {#2} = {#3} }
311 \cs_generate_variant:Nn \__xeCJK_inter_class_toks:nnn { nnc , nnx }
(End definition for \__xeCJK_inter_class_toks:nnn, \__xeCJK_inter_class_toks:nnc, and \__xeCJK_inter_class_toks:nnx)

```

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

```

312 \cs_new_nopar:Npn \__xeCJK_get_inter_class_toks:nn #1#2
313 { \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。

```

314 \cs_new_protected_nopar:Npn \__xeCJK_clear_inter_class_toks:nn #1#2
315 { \__xeCJK_inter_class_toks:nnn {#1} {#2} { \c_empty_tl } }
(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。

```

316 \cs_new_protected_nopar:Npn \__xeCJK_pre_inter_class_toks:nnn #1#2#3
317 {
318   \__xeCJK_inter_class_toks:nnx {#1} {#2}
319   { \exp_not:n {#3} \__xeCJK_get_inter_class_toks:nn {#1} {#2} }
320 }
(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。

```

321 \cs_new_protected_nopar:Npn \__xeCJK_app_inter_class_toks:nnn #1#2#3
322 {
323   \__xeCJK_inter_class_toks:nnx {#1} {#2}
324   { \__xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
325 }
326 \cs_generate_variant:Nn \__xeCJK_app_inter_class_toks:nnn { nnc }
(End definition for \__xeCJK_app_inter_class_toks:nnn and \__xeCJK_app_inter_class_toks:nnc)

```

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

```

327 \cs_new_protected_nopar:Npn \__xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
328 {
329   \__xeCJK_inter_class_toks:nnx {#1} {#2}
330   { \__xeCJK_get_inter_class_toks:nn {#3} {#4} }
331 }
(End definition for \__xeCJK_copy_inter_class_toks:nnnn)

```

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

```

332 \cs_new_protected_nopar:Npn \__xeCJK_clear_CJK_toks:
333 {
334   \clist_map_inline:Nn \g__xeCJK_CJK_class_clist
335   { \__xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
336 }
(End definition for \__xeCJK_clear_CJK_toks:)

```

## 5.5 字符输出规则

	Default	CJK	FullLeft	FullRight	HalfLeft	HalfRight	NormalSpace	Boundary
Default		✓	✓	✓				✓
CJK	✓	✓	✓	✓	✓	✓	✓	✓
FullLeft	✓	✓	✓	✓	✓	✓	✓	✓
FullRight	✓	✓	✓	✓	✓	✓	✓	✓
HalfLeft		✓	✓	✓				
HalfRight		✓	✓	✓				✓
NormalSpace		✓	✓	✓				✓
Boundary	✓	✓	✓	✓	✓		✓	

```

337 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
338 {
339   \clist_map_inline:nn { CJK, FullLeft, FullRight }
340   {
341     \__xeCJK_inter_class_toks:nnn {#1} {##1}
342     {
343       \c_group_begin_token
344       \xeCJK_select_font:
345       \__xeCJK_clear_inter_class_toks:nn {#1} {##1}
346       \__xeCJK_clear_CJK_toks:
347     }
348     \__xeCJK_inter_class_toks:nnn {##1} {#1} { \c_group_end_token }
349   }
350   \__xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
351   \clist_map_inline:nn { FullLeft, FullRight }
352   { \__xeCJK_app_inter_class_toks:nnc {#1} {##1} { xeCJK_CJK_and_##1:N } }
353 }
354 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
355 {
356   \__xeCJK_inter_class_toks:nnn { Boundary } {#1}
357   {
358     \bool_if:nTF
359     {
360       \l__xeCJK_xecglue_bool &&
361       \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
362       \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_space_glue_tl
363     }
364     {
365       \tex_unskip:D
366       \bool_if:nTF
367       { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK_space } }
368       { \CJKecglue } { \c_space_token }
369     }
370     {
371       \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
372       { \xeCJK_if_last_node:nT { CJK_space } { \__xeCJK_space_or_xecglue: } }
373     }
374   }
375   \str_if_eq_x:nnF {#1} { NormalSpace }
376   { \__xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }

```

```

377 }
378 \clist_map_inline:nn { Default, HalfRight, NormalSpace }
379 {
380   \__xeCJK_inter_class_toks:nnn {#1} { Boundary }
381   {
382     \peek_catcode:NTF \c_space_token
383     { { \xeCJK_make_node:n { default_space } } }
384     { { \xeCJK_make_node:n { default } } }
385   }
386   \str_if_eq_x:nnF {#1} { NormalSpace }
387   { \__xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
388 }
389 \__xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }

```

\xeCJK\_Boundary\_and\_CJK:

\\_\_xeCJK\_check\_for\_glue:

```

390 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_CJK:
391 {
392   { \__xeCJK_check_for_glue: }
393   \c_group_begin_token
394   \__xeCJK_clear_CJK_toks:
395   \xeCJK_select_font:
396   \CJKsymbol
397 }
398 \cs_new_protected_nopar:Npn \__xeCJK_check_for_glue:
399 {
400   \bool_if:nTF
401   { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK_space } }
402   { \CJKglue }
403   {
404     \bool_if:nTF
405     {
406       \xeCJK_if_last_node_p:n { default } ||
407       \int_compare_p:nNn \tex_lastnodetype:D = \c_ten ||
408       \xeCJK_if_last_node_p:n { default_itcorr }
409     }
410     { \CJKecglue }
411     {
412       \bool_if:nT
413       {
414         \l__xeCJK_xecglue_bool &&
415         \int_compare_p:nNn \tex_lastnodetype:D = \c_eleven &&
416         ( \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_space_glue_tl ||
417           \dim_compare_p:nNn \tex_lastskip:D = \l__xeCJK_ecglue_dim )
418       }
419       {
420         \tex_unskip:D
421         \bool_if:nTF
422         {
423           \xeCJK_if_last_node_p:n { default_space } ||
424           \int_compare_p:nNn \tex_lastnodetype:D = \c_ten ||
425           \xeCJK_if_last_node_p:n { default } ||
426           \xeCJK_if_last_node_p:n { default_itcorr }

```

```

427         }
428         { \CJKecglue }
429         {
430             \bool_if:nTF
431             {
432                 \xeCJK_if_last_node_p:n { CJK } ||
433                 \xeCJK_if_last_node_p:n { CJK_space }
434             }
435             { \bool_if:nTF \l__xeCJK_space_bool \c_space_token \CJKglue }
436             { \c_space_token }
437         }
438     }
439 }
440 }
441 }
442 \tl_set:Nn \l__xeCJK_space_glue_tl
443 {
444     \tex_fontdimen:D \c_two \tex_font:D
445     \@plus \tex_fontdimen:D \c_three \tex_font:D
446     \@minus \tex_fontdimen:D \c_four \tex_font:D
447 }
448 (End definition for \xeCJK_Boundary_and_CJK: and \__xeCJK_check_for_glue:)
449 \__xeCJK_inter_class_toks:nnn { CJK } { Boundary }
450 {
451     \c_group_end_token
452     { \xeCJK_make_node:n { CJK } }
453     \xeCJK_ignorespaces:
454 }
455 \__xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }
456 \xeCJK_CJK_and_CJK:N
457 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }
458 (End definition for \xeCJK_CJK_and_CJK:N)
459 \clist_map_inline:nn { FullLeft, FullRight }
460 {
461     \__xeCJK_inter_class_toks:nnn { Boundary } {#1}
462     {
463         \c_group_begin_token
464         \__xeCJK_clear_CJK_toks:
465         \xeCJK_select_font:
466         \use:c { xeCJK_CJK_and_#1:N }
467     }
468     \__xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
469     \__xeCJK_inter_class_toks:nnn {#1} { Boundary } { \c_group_end_token \tex_ignorespaces:D }
470 }
471 \clist_map_inline:Nn \g__xeCJK_base_class_clist
472 {
473     \bool_if:nF
474     { \str_if_eq_x_p:nn {#1} { FullLeft } || \str_if_eq_x_p:nn {#1} { FullRight } }
475     {
476         \__xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \xeCJK_no_break: }

```

```

474     \__xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \xeCJK_after_FullRight: }
475     }
476 }
477 \clist_map_inline:nn { CJK, FullLeft, FullRight }
478 {
479     \clist_map_inline:nn { FullLeft, FullRight }
480     { \__xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
481 }

```

\\_\_xeCJK\_punct\_rule:n 用于抹去标点符号的左/右空白。

```

482 \cs_new_protected_nopar:Npn \__xeCJK_punct_rule:n #1
483 {
484     \tex_vrule:D
485     \@width \dim_eval:c { \l__xeCJK_punct_coors_tl/rule/#1 }
486     \@depth \c_zero_dim
487     \@height \c_zero_dim \scan_stop:
488 }
(End definition for \__xeCJK_punct_rule:n)

```

\\_\_xeCJK\_punct\_glue:n 根据所选的标点处理方式在标点符号左/右增加的空白。

```

489 \cs_new_protected_nopar:Npn \__xeCJK_punct_glue:n #1
490 {
491     \skip_horizontal:n
492     {
493         \dim_eval:c { \l__xeCJK_punct_coors_tl/glue/#1 }
494         \@plus \dim_eval:n { ( \tl_use:c { \l__xeCJK_punct_coors_tl/glue/#1 } ) /4 }
495         \@minus \dim_eval:n { ( \tl_use:c { \l__xeCJK_punct_coors_tl/glue/#1 } ) /2 }
496     }
497 }
(End definition for \__xeCJK_punct_glue:n)

```

\\_\_xeCJK\_punct\_kern:n 相邻两个标点之间的间距。

```

498 \cs_new_protected_nopar:Npn \__xeCJK_punct_kern:n #1
499 { \tex_kern:D \dim_eval:c { \l__xeCJK_punct_coors_tl/kern/#1 } }
(End definition for \__xeCJK_punct_kern:n)

```

\dim\_eval:c

```

500 \cs_generate_variant:Nn \dim_eval:n { c }
(End definition for \dim_eval:c)

```

\xeCJK\_before\_FullLeft:N

```

501 \cs_new_protected_nopar:Npn \xeCJK_before_FullLeft:N #1
502 {
503     \tl_set:Nx \l__xeCJK_lastpunct_tl {#1}
504     \__xeCJK_punct_rule:n { l/#1 }
505     \CJKpunctsymbol {#1}
506 }
(End definition for \xeCJK_before_FullLeft:N)

```

\xeCJK\_after\_FullRight:

```
507 \cs_new_protected_nopar:Npn \xeCJK_after_FullRight:
508 {
509   \xeCJK_get_punct_bounds:nV { r } \l__xeCJK_lastpunct_tl
510   \__xeCJK_punct_rule:n { r/\l__xeCJK_lastpunct_tl }
511   \__xeCJK_punct_glue:n { r/\l__xeCJK_lastpunct_tl }
512 }
(End definition for \xeCJK_after_FullRight:)
```

\xeCJK\_CJK\_and\_FullLeft:N

```
513 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
514 {
515   \xeCJK_get_punct_bounds:nN { l } {#1}
516   \int_compare:nNf \etex_lastnodetype:D = \c_one
517   { \__xeCJK_punct_glue:n { l/#1 } }
518   \xeCJK_before_FullLeft:N {#1}
519   \__xeCJK_punct_if_middle:NT {#1}
520   {
521     \__xeCJK_punct_rule:n { lr/#1 }
522     \__xeCJK_punct_glue:n { l/#1 }
523   }
524 }
(End definition for \xeCJK_CJK_and_FullLeft:N)
```

\xeCJK\_CJK\_and\_FullRight:N

```
525 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
526 {
527   \xeCJK_get_punct_bounds:nN { r } {#1}
528   \__xeCJK_punct_if_long:NTF {#1} \CJKglue
529   {
530     \xeCJK_no_break:
531     \__xeCJK_punct_if_middle:NT {#1}
532     {
533       \__xeCJK_punct_glue:n { r/#1 }
534       \__xeCJK_punct_rule:n { lr/#1 }
535     }
536   }
537   \tl_gset:Nx \l__xeCJK_lastpunct_tl {#1}
538   \CJKpunctsymbol {#1}
539 }
(End definition for \xeCJK_CJK_and_FullRight:N)
```

\xeCJK\_FullLeft\_and\_FullLeft:N

```
540 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
541 {
542   \xeCJK_no_break:
543   \xeCJK_get_punct_bounds:nN { l } {#1}
544   \xeCJK_calc_kern:VN \l__xeCJK_lastpunct_tl {#1}
545   \__xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
546   \xeCJK_before_FullLeft:N {#1}
547 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N)
```

\xeCJK\_FullLeft\_and\_FullRight:N

```
548 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
549 {
550   \xeCJK_no_break:
551   \xeCJK_get_punct_bounds:nN { r } {#1}
552   \xeCJK_calc_kern:VN \l__xeCJK_lastpunct_tl {#1}
553   \__xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
554   \xeCJK_no_break:
555   \tl_set:Nx \l__xeCJK_lastpunct_tl {#1}
556   \CJKpunctsymbol {#1}
557 }
(End definition for \xeCJK_FullLeft_and_FullRight:N)
```

\xeCJK\_FullRight\_and\_FullLeft:N

```
558 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
559 {
560   \__xeCJK_punct_rule:n { r/\l__xeCJK_lastpunct_tl }
561   \xeCJK_get_punct_bounds:nN { l } {#1}
562   \xeCJK_calc_kern:VN \l__xeCJK_lastpunct_tl {#1}
563   \__xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
564   \__xeCJK_punct_nobreak:
565   \xeCJK_before_FullLeft:N {#1}
566 }
(End definition for \xeCJK_FullRight_and_FullLeft:N)
```

\xeCJK\_FullRight\_and\_FullRight:N

```
567 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
568 {
569   \__xeCJK_punct_rule:n { r/\l__xeCJK_lastpunct_tl }
570   \xeCJK_get_punct_bounds:nN { r } {#1}
571   \xeCJK_calc_kern:VN \l__xeCJK_lastpunct_tl {#1}
572   \__xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
573   \xeCJK_no_break:
574   \tl_set:Nx \l__xeCJK_lastpunct_tl {#1}
575   \CJKpunctsymbol {#1}
576 }
(End definition for \xeCJK_FullRight_and_FullRight:N)
```

\xeCJK\_if\_last\_node\_p:n

\xeCJK\_if\_last\_node:nTF

```
577 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
578 {
579   \if_dim:w \dim_use:c { c__xeCJK_#1_node_dim } = \tex_lastkern:D
580   \prg_return_true: \else: \prg_return_false: \fi:
581 }
(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

```
582 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
583 {
584   \dim_if_exist:CTF { c__xeCJK_#1_node_dim }
585   { \dim_gset:cn } { \dim_const:cn }
586   { c__xeCJK_#1_node_dim } {#2}
```

```

587 }
588 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
589 {
590   \tex_kern:D - \dim_use:c { c__xeCJK_#1_node_dim }
591   \tex_kern:D \dim_use:c { c__xeCJK_#1_node_dim }
592 }
593 \xeCJK_def_node:nn { CJK } { 11 sp }
594 \xeCJK_def_node:nn { CJK_space } { 12 sp }
595 \xeCJK_def_node:nn { default } { 13 sp }
596 \xeCJK_def_node:nn { default_space } { 14 sp }
597 \xeCJK_def_node:nn { default_itcorr } { 15 sp }
(End definition for \xeCJK_def_node:nn and \xeCJK_make_node:n)

```

**CJKglue** CJK 文字之间插入的 glue。

```

598 \keys_define:nn { xeCJK / options }
599 { CJKglue .code:n = { \cs_set_protected:Npn \CJKglue {#1} } }
(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**

```

600 \keys_define:nn { xeCJK / options }
601 {
602   CJKecglue .code:n =
603   {
604     \cs_set_protected:Npn \CJKecglue {#1}
605     \settowidth \l__xeCJK_ecglue_dim \CJKecglue
606   },
607   xCJKecglue .choice:,
608   xCJKecglue / true .code:n =
609   {
610     \bool_set_true:N \l__xeCJK_xecglue_bool
611     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
612   },
613   xCJKecglue / false .code:n =
614   {
615     \bool_set_false:N \l__xeCJK_xecglue_bool
616     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \c_space_token
617   },
618   xCJKecglue / unknown .code:n =
619   {
620     \bool_set_true:N \l__xeCJK_xecglue_bool
621     \cs_set_protected:Npn \CJKecglue {#1}
622     \settowidth \l__xeCJK_ecglue_dim \CJKecglue
623     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
624   },
625   xCJKecglue .default:n = { true } ,
626 }
627 \dim_new:N \l__xeCJK_ecglue_dim
628 \bool_new:N \l__xeCJK_xecglue_bool
(End definition for CJKecglue and xCJKecglue These functions are documented on page 4.)

```

**CJKspace** 是否保留 CJK 文字间的空白,默认不保留。

```

629 \keys_define:nn { xeCJK / options }

```

```

630 {
631   CJKspace .choice:,
632   CJKspace / true .code:n = { \CJKspace },
633   CJKspace / false .code:n = { \CJKnospace },
634   CJKspace .default:n = { true },
635   space .meta:n = { CJKspace = true },
636   nospace .meta:n = { CJKspace = false },
637 }

```

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

\CJKspace 是否保留 CJK 文字间的空白,默认不保留。

```

\CJKnospace 638 \NewDocumentCommand \CJKspace { }
639 {
640   \bool_set_true:N \l__xeCJK_space_bool
641   \cs_set_eq:NN \xeCJK_ignorespaces: \__xeCJK_peek_math:
642 }
643 \NewDocumentCommand \CJKnospace { }
644 {
645   \bool_set_false:N \l__xeCJK_space_bool
646   \cs_set_eq:NN \xeCJK_ignorespaces: \__xeCJK_ignore_spaces:
647 }

```

(End definition for \CJKspace and \CJKnospace)

```

\__xeCJK_ignore_spaces:
\__xeCJK_peek_math: 648 \cs_new_protected_nopar:Npn \__xeCJK_ignore_spaces:
649 {
650   \peek_catcode:NTF \c_space_token
651   {
652     \xeCJK_if_last_node:nT { CJK }
653     { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK_space } } }
654     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
655     { \__xeCJK_space_or_xecglue: }
656     {
657       \bool_if:nT
658       {
659         \xeCJK_if_outer_macro_p:c { l_peek_token } ||
660         \token_if_cs_p:c { l_peek_token }
661       }
662       { \__xeCJK_space_or_xecglue: }
663     }
664   }
665   {
666     \xeCJK_if_outer_macro:cF { l_peek_token }
667     {
668       \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
669       { \cs_if_exist:cF { l_peek_token } \exp_not:N }
670     }
671   }
672 }
673 \cs_new_protected_nopar:Npn \__xeCJK_peek_math:
674 {
675   \peek_catcode:NTF \c_space_token

```

```

676 {
677   \xeCJK_if_last_node:nT { CJK }
678   { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK_space } } }
679   \xeCJK_peek_catcode_ignore_spaces:NF \c_space_token
680   { \__xeCJK_space_or_xecglue: }
681 }
682 {
683   \xeCJK_if_outer_macro:cF { l_peek_token }
684   {
685     \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
686     { \cs_if_exist:cF { l_peek_token } \exp_not:N }
687   }
688 }
689 }

```

(End definition for \\_\_xeCJK\_ignore\_spaces: and \\_\_xeCJK\_peek\_math:)

## 5.6 段末孤字处理

**CheckSingle** 孤字处理功能选项。

```

690 \keys_define:nn { xeCJK / options }
691 {
692   CheckSingle .choice:,
693   CheckSingle / true .code:n = { \xeCJKenablechecksingle },
694   CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
695   CheckSingle .default:n = { true },
696   CJKchecksingle .meta:n = { CheckSingle = true } ,
697 }

```

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

**\xeCJKenablechecksingle** 孤字处理功能开关。

```

\exeCJKdisablechecksingle
698 \bool_new:N \l__xeCJK_checksingl_bool
699 \NewDocumentCommand \xeCJKenablechecksingle { }
700 {
701   \bool_if:NF \l__xeCJK_checksingl_bool
702   {
703     \bool_set_true:N \l__xeCJK_checksingl_bool
704     \cs_set_eq:NN \__xeCJK_checksingl_save_CJKsymbol:N \xeCJK_CJK_and_CJK:N
705     \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingl:N
706   }
707 }
708 \NewDocumentCommand \xeCJKdisablechecksingle { }
709 {
710   \bool_if:NT \l__xeCJK_checksingl_bool
711   {
712     \bool_set_false:N \l__xeCJK_checksingl_bool
713     \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_checksingl_save_CJKsymbol:N
714   }
715 }

```

(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle)

\xeCJK\_checksingl:N  
 \xeCJK\_checksingl:NN  
 \xeCJK\_checksingl:NNN

```

716 \cs_new_protected_nopar:Npn \xeCJK_checksingl:N #1

```

```

717 {
718   \peek_catcode:NTF \c_catcode_other_token
719   { \xeCJK_checksinglingle:NN {#1} }
720   {
721     \bool_if:nTF
722     {
723       ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } )      &&
724       \xeCJK_if_blank_x_p:n { \token_get_arg_spec:c { l_peek_token } } &&
725       \exp_args:NNc \exp_args:No \tl_if_single_token_p:n { l_peek_token } &&
726       \exp_args:NNc \exp_after:wN \token_if_other_p:N { l_peek_token }
727     }
728     { \xeCJK_checksinglingle:NN {#1} }
729     { \__xeCJK_checksinglingle_save_CJKsymbol:N {#1} }
730   }
731 }
732 \cs_new_protected_nopar:Npn \xeCJK_checksinglingle:NN #1#2
733 {
734   \peek_catcode:NTF \c_catcode_other_token
735   { \__xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2 }
736   {
737     \bool_if:nTF
738     {
739       ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
740       ( \token_if_space_p:c { l_peek_token } ||
741         \token_if_math_toggle_p:c { l_peek_token } ||
742         \token_if_cs_p:c { l_peek_token } )
743     }
744     {
745       \bool_if:nTF { \token_if_space_p:c { l_peek_token } }
746       {
747         \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_other_token
748         { \__xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2 \c_space_token }
749         {
750           \bool_if:nTF
751           {
752             ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
753             ( \token_if_math_toggle_p:c { l_peek_token } ||
754               \token_if_cs_p:c { l_peek_token } )
755           }
756           { \xeCJK_checksinglingle:NNN {#1} {#2} \c_space_token }
757           { \__xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2 \c_space_token }
758         }
759       }
760       { \xeCJK_checksinglingle:NNN {#1} {#2} { } }
761     }
762     { \__xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2 }
763   }
764 }
765 \cs_new_protected_nopar:Npn \xeCJK_checksinglingle:NNN #1#2#3
766 {
767   \bool_if:nTF
768   {

```

```

769     \token_if_eq_meaning_p:NN \l_peek_token \par ||
770     \token_if_eq_meaning_p:NN \l_peek_token \[ % \]
771   }
772   { \CJKsymbol {#1} \xeCJK_no_break: #2 }
773   { \xeCJK_checksingle_env:NNN {#1} {#2} {#3} }
774 }
(End definition for \xeCJK_checksingle:N, \xeCJK_checksingle:NN, and \xeCJK_checksingle:NNN)

\xeCJK_checksingle_env:NNN
\xeCJK_checksingle_env:NNNNn 775 \cs_new_protected_nopar:Npn \xeCJK_checksingle_env:NNN #1#2#3
776 {
777   \peek_catcode_remove:NTF \c_math_toggle_token
778   {
779     \peek_catcode:NTF \c_math_toggle_token
780     { \CJKsymbol {#1} \xeCJK_no_break: #2 \c_math_toggle_token }
781     { \__xeCJK_checksingle_save_CJKsymbol:N {#1} #2#3 \c_math_toggle_token }
782   }
783   {
784     \peek_meaning_remove:NTF \begin
785     { \xeCJK_checksingle_env:NNNNn {#1} {#2} {#3} \begin }
786     {
787       \peek_meaning_remove:NTF \end
788       { \xeCJK_checksingle_env:NNNNn {#1} {#2} {#3} \end }
789       { \__xeCJK_checksingle_save_CJKsymbol:N {#1} #2#3 }
790     }
791   }
792 }
793 \cs_new_protected_nopar:Npn \xeCJK_checksingle_env:NNNNn #1#2#3#4#5
794 {
795   \clist_if_in:NnTF \g__xeCJK_inline_env_clist {#5}
796   { \__xeCJK_checksingle_save_CJKsymbol:N {#1} #2#3 }
797   { \CJKsymbol {#1} \xeCJK_no_break: #2 }
798   \scan_stop: #4 {#5}
799 }
(End definition for \xeCJK_checksingle_env:NNN and \xeCJK_checksingle_env:NNNNn)

```

InlineEnv

InlineEnv+

InlineEnv-

```

800 \keys_define:nn { xeCJK / options }
801 {
802   InlineEnv .clist_gset:N = \g__xeCJK_inline_env_clist ,
803   InlineEnv+ .code:n =
804   {
805     \clist_gput_right:Nx \g__xeCJK_inline_env_clist {#1}
806     \clist_gremove_duplicates:N \g__xeCJK_inline_env_clist
807   },
808   InlineEnv- .code:n =
809   {
810     \clist_map_inline:xn {#1}
811     { \clist_gremove_all:Nn \g__xeCJK_inline_env_clist {##1} }
812   },
813 }

```

(End definition for InlineEnv, InlineEnv+, and InlineEnv- These functions are documented on page 4.)

```
814 \cs_generate_variant:Nn \token_if_cs_p:N { c }
815 \cs_generate_variant:Nn \token_if_space_p:N { c }
816 \cs_generate_variant:Nn \token_get_arg_spec:N { c }
817 \cs_generate_variant:Nn \token_if_math_toggle_p:N { c }
818 \cs_generate_variant:Nn \token_if_math_toggle:NTF { c }
```

## 5.7 增加 CJK 子分区

\\_xeCJK\_UL\_subclass\_patch\_tl  
\g\_\_xeCJK\_CJK\_subclass\_clist

```
819 \tl_new:N \_xeCJK_UL_subclass_patch_tl
820 \clist_new:N \g__xeCJK_CJK_subclass_clist
```

(End definition for \\_xeCJK\_UL\_subclass\_patch\_tl and \g\_\_xeCJK\_CJK\_subclass\_clist)

\\_\_xeCJK\_erase\_CJKsymbol:  
\\_\_xeCJK\_restore\_CJKsymbol:

```
821 \cs_new_protected_nopar:Npn \__xeCJK_erase_CJKsymbol:
822 {
823   \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
824   \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
825 }
826 \cs_new_protected_nopar:Npn \__xeCJK_restore_CJKsymbol:
827 {
828   \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block
829   \CJKsymbol
830 }
```

(End definition for \\_\_xeCJK\_erase\_CJKsymbol: and \\_\_xeCJK\_restore\_CJKsymbol:)

\xeCJKDeclareSubCJKBlock

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

```
831 \NewDocumentCommand \xeCJKDeclareSubCJKBlock { s m m }
832 {
833   \xeCJKDeclareSubCharClass { CJK } {#2} {#3}
834   \IfBooleanT {#1} { \xeCJKResetPunctClass }
835 }
836 \@onlypreamble \xeCJKDeclareSubCJKBlock
```

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

\xeCJKCancelSubCJKBlock

取消和恢复对 CJK 子区的声明。

\xeCJKRestoreSubCJKBlock

```
837 \bool_new:N \l__xeCJK_sub_cancel_bool
838 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
839 {
840   \bool_if:NF \l__xeCJK_sub_cancel_bool
841   {
842     \bool_set_true:N \l__xeCJK_sub_cancel_bool
843     \__xeCJK_sub_restore_or_cancel:n {#2}
844     \IfBooleanT {#1} { \xeCJKResetPunctClass }
845   }
846 }
847 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
848 {
849   \bool_if:NT \l__xeCJK_sub_cancel_bool
850   {
```

```

851     \bool_set_false:N \l__xeCJK_sub_cancel_bool
852     \__xeCJK_sub_restore_or_cancel:n {#2}
853     \IfBooleanT {#1} { \xeCJKResetPunctClass }
854   }
855 }
856 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
857 {
858   \clist_map_inline:xn {#1}
859   {
860     \int_if_exist:cTF { c__xeCJK_ CJK/##1 _class_int }
861     {
862       \__xeCJK_declare_char_class:nn
863       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
864       { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
865     }
866     { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
867   }
868 }
869 \__xeCJK_msg_new:nn { SubBlock-undefined }
870 {
871   The~CJK~sub~block~'#1'~is~undefined.\\
872   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
873   to~declare~it.
874 }

```

(End definition for \xeCJKCancelSubCJKBlock and \xeCJKRestoreSubCJKBlock These functions are documented on page 10.)

#### \xeCJKDeclareSubCharClass

```

875 \NewDocumentCommand \xeCJKDeclareSubCharClass
876 { > { \TrimSpaces } m > { \TrimSpaces } m m }
877 {
878   \int_if_exist:cF { c__xeCJK_ #1/#2 _class_int }
879   {
880     \__xeCJK_new_class:n { #1/#2 }
881     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
882     \__xeCJK_new_sub_key:n {#2}
883     \__xeCJK_UL_subclass_patch:nn {#1} {#2}
884   }
885   \__xeCJK_declare_char_class:nn { #1/#2 } {#3}
886 }
887 \@onlypreamble \xeCJKDeclareSubCharClass

```

(End definition for \xeCJKDeclareSubCharClass)

#### \\_\_xeCJK\_set\_sub\_class\_toks:nn

```

888 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
889 {
890   \clist_map_inline:Nn \g__xeCJK_base_class_clist
891   {
892     \__xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
893     \__xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
894     \str_if_eq_x:nnTF {##1} { CJK }
895     {

```

```

896         \__xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
897         { \__xeCJK_switch_font:nn {#1} {#2} }
898     }
899     {
900         \__xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
901         { \__xeCJK_erase_CJKsymbol: }
902         \__xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
903         { \__xeCJK_switch_font:nn {#1} {#2} \__xeCJK_restore_CJKsymbol: }
904     }
905 }
906 \__xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
907 \clist_if_empty:NF \g__xeCJK_CJK_subclass_clist
908 {
909     \clist_map_inline:Nn \g__xeCJK_CJK_subclass_clist
910     {
911         \__xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
912         \__xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
913         \__xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
914         { \__xeCJK_switch_font:nn {#2} {##1} }
915         \__xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
916         { \__xeCJK_switch_font:nn {##1} {#2} }
917     }
918 }
919 \clist_gput_right:Nn \g__xeCJK_CJK_subclass_clist {#2}
920 \__xeCJK_save_CJK_class:n { #1/#2 }
921 \clist_map_inline:nn { CJK, FullLeft, FullRight }
922 {
923     \__xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
924     { \__xeCJK_switch_font:nn {#2} {#1} }
925 }
926 }
(End definition for \__xeCJK_set_sub_class_toks:nn)

```

\\_\_xeCJK\_UL\_subclass\_patch:nn

```

927 \cs_new_protected_nopar:Npn \__xeCJK_UL_subclass_patch:nn #1#2
928 {
929     \tl_put_right:Nn \__xeCJK_UL_subclass_patch_tl
930     {
931         \clist_map_inline:nn { Default, HalfLeft, HalfRight }
932         {
933             \str_if_eq_x:nnTF {##1} { HalfLeft }
934             { \__xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKecglue \CJKsymbol } }
935             { \__xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
936             \__xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
937             {
938                 \__xeCJK_erase_CJKsymbol:
939                 \xeCJK_Boundary_and_CJK:
940                 \__xeCJK_switch_font:nn {#1} {#2}
941                 \__xeCJK_restore_CJKsymbol:
942             }
943         }
944     }
945 }

```

(End definition for \\_\_xeCJK\_UL\_subclass\_patch:nn)

## 5.8 标点处理

\g\_\_xeCJK\_punctstyle\_clist 标点处理方式。

```
946 \clist_new:N \g__xeCJK_punctstyle_clist
947 \clist_set:Nn \g__xeCJK_punctstyle_clist
948   { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
949 \clist_map_inline:Nn \g__xeCJK_punctstyle_clist
950   { \tl_const:cn { c__xeCJK_ps_#1_tl } {#1} }
(End definition for \g__xeCJK_punctstyle_clist)
```

\keys\_define:nx

```
951 \cs_generate_variant:Nn \keys_define:nn { nx }
(End definition for \keys_define:nx)
```

### PunctStyle

```
952 \keys_define:nn { xeCJK / options }
953   {
954     PunctStyle .choice_code:n =
955       {
956         \tl_set:Nx \l_xeCJK_punctstyle_tl { \l_keys_choice_tl }
957         \tl_if_eq:NNT \l_xeCJK_punctstyle_tl \c__xeCJK_ps_plain_tl
958           { \xeCJKallowbreakbetweenpuncts }
959       },
960   }
961 \keys_define:nx { xeCJK / options }
962   { PunctStyle .generate_choices:n = { \exp_not:V \g__xeCJK_punctstyle_clist } }
963 \keys_define:nn { xeCJK / options }
964   {
965     PunctStyle / banjiao .meta:n = { PunctStyle = halfwidth },
966     PunctStyle / quanjiao .meta:n = { PunctStyle = fullwidth },
967     PunctStyle / kaiming .meta:n = { PunctStyle = mixedwidth },
968     PunctStyle / hangmobanjiao .meta:n = { PunctStyle = marginkerning },
969     PunctStyle / unknown .code:n =
970       { \__xeCJK_error:nx { PunctStyle-undefined } \l_keys_value_tl },
971   }
972 \__xeCJK_msg_new:nn { PunctStyle-undefined }
973   {
974     Punctstyle~"#1"~is~not~available. \\\
975     You~can~only~use~one~of\\\
976     "\g__xeCJK_punctstyle_clist".\\
977   }
```

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

AllowBreakBetweenPuncts 相关选项声明。

```
KaiMingPunct 978 \keys_define:nn { xeCJK / options }
LongPunct    979   {
MiddlePunct  980     AllowBreakBetweenPuncts .choice:,
PunctWidth   981     AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
              982     AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
              983     AllowBreakBetweenPuncts .default:n = { true },
```

```

984 KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixedwidth } {#1} },
985 KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixedwidth } {#1} },
986 KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
987 LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} },
988 LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} },
989 LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} },
990 MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} },
991 MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} },
992 MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} },
993 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
994 }

```

(End definition for AllowBreakBetweenPuncts and others. These functions are documented on page 5.)

```

\__xeCJK_allowbreakbetweenpuncts
\__xeCJK_nobreakbetweenpuncts
995 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
996 { \cs_set_eq:NN \__xeCJK_punct_nobreak: \__xeCJK_zero_glue: }
997 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
998 { \cs_set_eq:NN \__xeCJK_punct_nobreak: \xeCJK_no_break: }
999 \cs_new_protected_nopar:Npn \__xeCJK_zero_glue: { \skip_horizontal:N \c_zero_skip }
(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts)

```

```

\g__xeCJK_special_ps_clist
\__xeCJK_set_special_punct:nn 1000 \clist_set:Nn \g__xeCJK_special_ps_clist { mixedwidth , long , middle }
\__xeCJK_add_special_punct:nn 1001 \clist_map_inline:Nn \g__xeCJK_special_ps_clist
\__xeCJK_sub_special_punct:nn 1002 {
1003   \tl_new:c { l__xeCJK_#1_punct_tl }
1004   \prop_new:c { l__xeCJK_#1_punct_prop }
1005 }
1006 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1007 {
1008   \tl_set:cx { l__xeCJK_#1_punct_tl } {#2}
1009   \prop_clear:c { l__xeCJK_#1_punct_prop }
1010   \tl_map_inline:cn { l__xeCJK_#1_punct_tl }
1011     { \prop_put:cnn { l__xeCJK_#1_punct_prop } {##1} { } }
1012 }
1013 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1014 {
1015   \tl_put_right:cx { l__xeCJK_#1_punct_tl } {#2}
1016   \tl_map_inline:cn { l__xeCJK_#1_punct_tl }
1017     { \prop_put:cnn { l__xeCJK_#1_punct_prop } {##1} { } }
1018 }
1019 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1020 {
1021   \tl_map_inline:xn {#2}
1022   {
1023     \tl_remove_all:cn { l__xeCJK_#1_punct_tl } {##1}
1024     \prop_remove:cn { l__xeCJK_#1_punct_prop } {##1}
1025   }
1026 }
1027 \cs_generate_variant:Nn \tl_map_inline:nn { x }
(End definition for \g__xeCJK_special_ps_clist and others.)

```

```

\dim_set:Nc
\dim_add:Nc 1028 \cs_generate_variant:Nn \dim_set:Nn { Nc }
\dim_min:cc 1029 \cs_generate_variant:Nn \dim_add:Nn { Nc }
1030 \cs_generate_variant:Nn \dim_min:nn { cc }
(End definition for \dim_set:Nc, \dim_add:Nc, and \dim_min:cc)

\XeCJK_dim_set_max:Nn
\XeCJK_dim_set_min:Nn 1031 \cs_new_protected_nopar:Npn \XeCJK_dim_set_max:Nn #1#2
1032 { \dim_set:Nn #1 { \dim_max:nn {#1} {#2} } }
1033 \cs_new_protected_nopar:Npn \XeCJK_dim_set_min:Nn #1#2
1034 { \dim_set:Nn #1 { \dim_min:nn {#1} {#2} } }
(End definition for \XeCJK_dim_set_max:Nn and \XeCJK_dim_set_min:Nn)

\l__XeCJK_punct_coortl 记录当前字体状态下的标点处理方式。
1035 \tl_set:Nn \l__XeCJK_punct_coortl { \l__XeCJK_font_coortl/\l__XeCJK_punctstyle_tl }
(End definition for \l__XeCJK_punct_coortl)

\XeCJK_get_punct_bounds:nN #1 为 l/r, #2 为标点字符, 返回标点的实际左/右空白的负值和根据标点处理方式决定添加的左/右空白。全角方式直接添加标点的实际左/右空白; 半角方式添加标点左右空白中的最小值; 开明方式与半角相同, 只在开明标点再增加实际左/右空白的一半。
1036 \cs_new_protected_nopar:Npn \XeCJK_get_punct_bounds:nN #1#2
1037 {
1038   \tl_if_exist:cF { \l__XeCJK_punct_coortl/rule/#1/#2 }
1039   {
1040     \tl_if_eq:NNTF \l__XeCJK_punctstyle_tl \c__XeCJK_ps_plain_tl
1041     {
1042       \tl_set:NV \l__XeCJK_tmpa_tl \c_zero_dim
1043       \tl_gset_eq:cN { \l__XeCJK_punct_coortl/glue/#1/#2 } \l__XeCJK_tmpa_tl
1044       \tl_gset_eq:cN { \l__XeCJK_punct_coortl/rule/#1/#2 } \l__XeCJK_tmpa_tl
1045       \tl_gset_eq:cN { \l__XeCJK_punct_coortl/rule/lr/#2 } \l__XeCJK_tmpa_tl
1046       \tl_gset_eq:cN { \l__XeCJK_punct_coortl/bound/#1/#2 } \l__XeCJK_tmpa_tl
1047       \tl_gset_eq:cN { \l__XeCJK_punct_coortl/bound/lr/#2 } \l__XeCJK_tmpa_tl
1048     }
1049     {
1050       \tl_if_exist:cF { \l__XeCJK_punct_coortl/bound/#1/#2 }
1051       { { \XeCJK_select_font: \XeCJK_calc_punct_dimen:N {#2} } }
1052       \__XeCJK_punct_if_long:NTF {#2}
1053       {
1054         \dim_zero:N \l__XeCJK_tmpa_dim
1055         \dim_zero:N \l__XeCJK_tmppb_dim
1056         \dim_zero:N \l__XeCJK_tmppc_dim
1057       }
1058       {
1059         \dim_set:Nc \l__XeCJK_tmpa_dim { \l__XeCJK_punct_coortl/bound/#1/#2 }
1060         \dim_set:Nc \l__XeCJK_tmppb_dim
1061         { \l__XeCJK_punct_coortl/bound/\str_if_eq_x:nnTF {#1} lrl/#2 }
1062         \dim_set:Nn \l__XeCJK_tmppc_dim
1063         {
1064           \cs_if_exist_use:cF { g__XeCJK_punct_width/#2 }
1065           {
1066             \XeCJK_if_blank_x:nTF \g__XeCJK_punct_width_tl
1067             { \c_zero_dim } { \g__XeCJK_punct_width_tl }

```

```

1068     }
1069   }
1070   \__xeCJK_punct_if_middle:NTF {#2}
1071   {
1072     \dim_set:Nn \l__xeCJK_tmpc_dim
1073     {
1074       \dim_compare:nNnTF \l__xeCJK_tmpc_dim > \c_zero_dim
1075       { \l__xeCJK_tmpc_dim }
1076       {
1077         \tl_case:Nnn \l_xeCJK_punctstyle_tl
1078         {
1079           \c__xeCJK_ps_halfwidth_tl { .5 em }
1080           \c__xeCJK_ps_mixedwidth_tl { .5 em }
1081           \c__xeCJK_ps_CCT_tl { .7 em }
1082           \c__xeCJK_ps_fullwidth_tl { 1 em }
1083         }
1084         { 1 em }
1085       }
1086       - ( \tl_use:c { \l__xeCJK_punct_coors_tl/dimen/#2 } )
1087     }
1088     \dim_compare:nNnTF \l__xeCJK_tmpc_dim > \c_zero_dim
1089     { \dim_set:Nn \l__xeCJK_tmpc_dim { .5 \l__xeCJK_tmpc_dim } }
1090     {
1091       \dim_set:Nn \l__xeCJK_tmpc_dim
1092       { \dim_min:nn \l__xeCJK_tmpa_dim \l__xeCJK_tmpb_dim }
1093     }
1094   }
1095   {
1096     \dim_compare:nNnTF \l__xeCJK_tmpc_dim > \c_zero_dim
1097     {
1098       \dim_set:Nn \l__xeCJK_tmpc_dim
1099       {
1100         \l__xeCJK_tmpc_dim - \l__xeCJK_tmpb_dim
1101         - ( \tl_use:c { \l__xeCJK_punct_coors_tl/dimen/#2 } )
1102       }
1103     }
1104     {
1105       \dim_set:Nn \l__xeCJK_tmpc_dim
1106       { \dim_min:nn \l__xeCJK_tmpa_dim \l__xeCJK_tmpb_dim }
1107       \tl_case:Nnn \l_xeCJK_punctstyle_tl
1108       {
1109         \c__xeCJK_ps_halfwidth_tl { \prg_do_nothing: }
1110         \c__xeCJK_ps_mixedwidth_tl
1111         {
1112           \__xeCJK_punct_if_mixedwidth:NT {#2}
1113           {
1114             \dim_add:Nn \l__xeCJK_tmpc_dim { .5 \l__xeCJK_tmpa_dim }
1115             \xeCJK_dim_set_min:Nn \l__xeCJK_tmpc_dim \l__xeCJK_tmpa_dim
1116           }
1117         }
1118         \c__xeCJK_ps_CCT_tl
1119         {

```

```

1120         \__xeCJK_punct_if_mixedwidth:NTF {#2}
1121         { \dim_add:Nn \l__xeCJK_tmpc_dim { .5 \l__xeCJK_tmpa_dim } }
1122         { \dim_add:Nn \l__xeCJK_tmpc_dim { .3 \l__xeCJK_tmpa_dim } }
1123         \xeCJK_dim_set_min:Nn \l__xeCJK_tmpc_dim \l__xeCJK_tmpa_dim
1124         }
1125     }
1126     { \dim_set_eq:NN \l__xeCJK_tmpc_dim \l__xeCJK_tmpa_dim }
1127 }
1128 }
1129 \xeCJK_dim_set_max:Nn \l__xeCJK_tmpc_dim \c_zero_dim
1130 }
1131 \tl_gset:cx { \l__xeCJK_punct_coors_tl/rule/#1/#2 }
1132 { - \dim_use:N \l__xeCJK_tmpa_dim }
1133 \tl_gset:cx { \l__xeCJK_punct_coors_tl/rule/lr/#2 }
1134 { - \dim_use:N \l__xeCJK_tmppb_dim }
1135 \tl_gset:cV { \l__xeCJK_punct_coors_tl/glue/#1/#2 } \l__xeCJK_tmpc_dim
1136 }
1137 }
1138 }
1139 \cs_generate_variant:Nn \xeCJK_get_punct_bounds:nN { nV }
(End definition for \xeCJK_get_punct_bounds:nN)

```

#### \xeCJKsetwidth

```

1140 \NewDocumentCommand \xeCJKsetwidth { m m }
1141 { \tl_map_inline:xn {#1} { \tl_gset:cn { g__xeCJK_punct_width/##1 } {#2} } }
(End definition for \xeCJKsetwidth This function is documented on page 11.)

```

```

\__xeCJK_punct_if_right_p:N
\__xeCJK_punct_if_both_left_right_p:NN
\__xeCJK_punct_if_long_p:N
\__xeCJK_punct_if_middle_p:N
\__xeCJK_punct_if_mixedwidth_p:N
\__xeCJK_punct_if_right:NTF
\__xeCJK_punct_if_both_left_right:NTF
\__xeCJK_punct_if_long:NTF
\__xeCJK_punct_if_middle:NTF
\__xeCJK_punct_if_mixedwidth:NTF
1142 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
1143 {
1144     \tl_if_exist:cTF { \l__xeCJK_punct_coors_tl/glue/r/#1 }
1145     \prg_return_true: \prg_return_false:
1146 }
1147 \prg_new_conditional:Npnn \__xeCJK_punct_if_both_left_right:NN #1#2 { p , T , F , TF }
1148 {
1149     \bool_if:nTF
1150     {
1151         ( \tl_if_exist_p:c { \l__xeCJK_punct_coors_tl/glue/r/#1 } &&
1152         \tl_if_exist_p:c { \l__xeCJK_punct_coors_tl/glue/r/#2 } ) ||
1153         ( \tl_if_exist_p:c { \l__xeCJK_punct_coors_tl/glue/l/#1 } &&
1154         \tl_if_exist_p:c { \l__xeCJK_punct_coors_tl/glue/l/#2 } )
1155     }
1156     \prg_return_true: \prg_return_false:
1157 }
1158 \cs_generate_variant:Nn \prg_new_conditional:Npnn { c }
1159 \clist_map_inline:Nn \g__xeCJK_special_ps_clist
1160 {
1161     \prg_new_conditional:cpnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1162     {
1163         \prop_if_in:cnTF { l__xeCJK_#1_punct_prop } {##1}
1164         \prg_return_true: \prg_return_false:
1165     }
1166 }

```

(End definition for \\_xeCJK\_punct\_if\_right:N and others.)

\xeCJK\_calc\_kern:NN 标点挤压。行末半角方式,相邻标点不挤压;全角方式,相邻两个标点占 1.5 汉字宽度,半角、开明和 CCT 方式相邻标点占一个汉字宽度。

```
1167 \cs_new_protected_nopar:Npn \xeCJK_calc_kern:NN #1#2
1168 {
1169   \tl_if_exist:cF { \l__xeCJK_punct_coor_tl/kern/#1-#2 }
1170   {
1171     \dim_zero:N \l__xeCJK_tmpa_dim
1172     \tl_if_eq:NNF \l_xeCJK_punctstyle_tl \c__xeCJK_ps_plain_tl
1173     {
1174       \tl_if_exist:cTF { g__xeCJK_punct/kern/#1-#2 }
1175       { \dim_set:Nc \l__xeCJK_tmpa_dim { g__xeCJK_punct/kern/#1-#2 } }
1176       { \_xeCJK_calc_kern:NNN \l__xeCJK_tmpa_dim {#1} {#2} }
1177     }
1178     \tl_gset:cV { \l__xeCJK_punct_coor_tl/kern/#1-#2 } \l__xeCJK_tmpa_dim
1179   }
1180 }
1181 \cs_new_protected_nopar:Npn \_xeCJK_calc_kern:NNN #1#2#3
1182 {
1183   \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c__xeCJK_ps_marginkerning_tl
1184   {
1185     \dim_zero:N #1
1186     \_xeCJK_punct_if_right:NT {#2}
1187     { \dim_add:Nc #1 { \l__xeCJK_punct_coor_tl/bound/r/#2 } }
1188     \_xeCJK_punct_if_right:NF {#3}
1189     { \dim_add:Nc #1 { \l__xeCJK_punct_coor_tl/bound/l/#3 } }
1190   }
1191   {
1192     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c__xeCJK_ps_fullwidth_tl
1193     { \dim_set:Nn #1 { \xeCJK_calc_kern:nnn { 1.5 em } {#2} {#3} } }
1194     {
1195       \_xeCJK_punct_if_both_left_right:NNTF {#2} {#3}
1196       {
1197         \_xeCJK_punct_if_mixedwidth:NT {#2}
1198         {
1199           \dim_set:Nn #1 { \xeCJK_calc_kern:nnn { 1 em } {#2} {#3} }
1200           \dim_compare:nNnT #1 < { .1 em }
1201           { \dim_set:Nc #1 { \l__xeCJK_punct_coor_tl/bound/l/#2 } }
1202         }
1203       }
1204       {
1205         \bool_if:nTF
1206         {
1207           \_xeCJK_punct_if_mixedwidth_p:N {#2} &&
1208           ! ( \tl_if_eq_p:NN \l_xeCJK_punctstyle_tl \c__xeCJK_ps_halfwidth_tl )
1209         }
1210         {
1211           \dim_set:Nn #1 { \tl_use:c { \l__xeCJK_punct_coor_tl/glue/r/#2 } }
1212           \dim_set:Nn #1 { .7 #1 }
1213         }
1214       }
1215     }
1216   }
```

```

1215 \dim_set:Nn #1 { \xeCJK_calc_kern:nNN { 1 em } {#2} {#3} }
1216 \dim_compare:nNnT #1 < { .1 em }
1217 {
1218   \dim_set:Nn #1
1219   {
1220     \dim_max:nn
1221     {
1222       \dim_min:cc
1223       { \l__xeCJK_punct_coord_tl/bound/l/#2 }
1224       { \l__xeCJK_punct_coord_tl/bound/r/#2 }
1225     }
1226     {
1227       \dim_min:cc
1228       { \l__xeCJK_punct_coord_tl/bound/l/#3 }
1229       { \l__xeCJK_punct_coord_tl/bound/r/#3 }
1230     }
1231   }
1232 }
1233 }
1234 }
1235 }
1236 \bool_if:nT
1237 { \__xeCJK_punct_if_long_p:N {#2} || \__xeCJK_punct_if_long_p:N {#3} }
1238 { \xeCJK_dim_set_max:Nn #1 { .1 em } }
1239 }
1240 \xeCJK_dim_set_max:Nn #1 \c_zero_dim
1241 }
1242 \cs_generate_variant:Nn \xeCJK_calc_kern:NN { V }
(End definition for \xeCJK_calc_kern:NN and \__xeCJK_calc_kern:NNN)

```

`\xeCJK_calc_kern: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。

```

1243 \cs_new_protected_nopar:Npn \xeCJK_calc_kern:nNN #1#2#3
1244 {
1245   \dim_eval:n
1246   {
1247     #1
1248     - \tl_use:c { \l__xeCJK_punct_coord_tl/
1249       \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } /l/#2 }
1250     - \tl_use:c { \l__xeCJK_punct_coord_tl/
1251       \__xeCJK_punct_if_right:NTF {#3} { glue } { bound } /r/#3 }
1252     - \tl_use:c { \l__xeCJK_punct_coord_tl/dimen/#2 }
1253     - \tl_use:c { \l__xeCJK_punct_coord_tl/dimen/#3 }
1254     \__xeCJK_punct_if_both_left_right:NNT {#2} {#3}
1255     { + \tl_use:c { \l__xeCJK_punct_coord_tl/rule/lr/#3 } }
1256   }
1257 }
(End definition for \xeCJK_calc_kern:nNN)

```

`\xeCJK_calc_punct_dimen:N` 返回标点的左右空白和实际尺寸。对于破折号, 计算两标点之间的空白, 保证它中间不被断开。

```

1258 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
1259 {

```

```

1260 \tl_gset:cx { \l__xeCJK_punct_coors_tl/bound/l/#1 }
1261           { \xeCJK_glyph_bounds:nN { left } {#1} }
1262 \tl_gset:cx { \l__xeCJK_punct_coors_tl/bound/r/#1 }
1263           { \xeCJK_glyph_bounds:nN { right } {#1} }
1264 \tl_gset:cx { \l__xeCJK_punct_coors_tl/bound/lr/#1 }
1265     {
1266       \dim_eval:n
1267       {
1268         ( \tl_use:c { \l__xeCJK_punct_coors_tl/bound/l/#1 } ) +
1269         ( \tl_use:c { \l__xeCJK_punct_coors_tl/bound/r/#1 } )
1270       }
1271     }
1272 \tl_gset:cx { \l__xeCJK_punct_coors_tl/dimen/#1 }
1273     {
1274       \dim_eval:n
1275       {
1276         ( \etex_fontcharwd:D \tex_font:D `#1 ) -
1277         ( \tl_use:c { \l__xeCJK_punct_coors_tl/bound/lr/#1 } )
1278       }
1279     }
1280 \bool_if:nT
1281   {
1282     \__xeCJK_punct_if_long_p:N {#1} &&
1283     ! ( \str_if_eq_x_p:nn {#1} { ^^^^2025 } || \str_if_eq_x_p:nn {#1} { ^^^^2026 } )
1284   }
1285   {
1286     \clist_map_inline:Nn \g__xeCJK_punctstyle_clist
1287     {
1288       \str_if_eq_x:nnF \c__xeCJK_ps_plain_tl {##1}
1289       {
1290         \tl_gset:cx { \l__xeCJK_font_coors_tl/##1/kern/#1-#1 }
1291         { - \tl_use:c { \l__xeCJK_punct_coors_tl/bound/lr/#1 } }
1292       }
1293     }
1294   }
1295 }

```

(End definition for \xeCJK\_calc\_punct\_dimen:N)

\xeCJK\_glyph\_bounds:nN 用 \XeTeXglyphbounds 计算标点的上下左右空白。

```

1296 \cs_new_protected_nopar:Npn \xeCJK_glyph_bounds:nN #1#2
1297 {
1298   \dim_eval:n
1299   {
1300     \XeTeXglyphbounds
1301     \str_case:x:nnn {#1}
1302     {
1303       { left } { \c_one } { top } { \c_two }
1304       { right } { \c_three } { bottom } { \c_four }
1305     }
1306     { \c_three }
1307     \int_eval:n { \XeTeXcharglyph `#2 }
1308   }
1309 }

```

(End definition for \xeCJK\_glyph\_bounds:nN)

**\xeCJKsetkern** 手动设置相邻标点的距离, 仅在当前字体状态下有效。

```
1310 \NewDocumentCommand \xeCJKsetkern { m m m }
1311 { \tl_gset:cn { g__xeCJK_punct/kern/#1-#2 } {#3} }
(End definition for \xeCJKsetkern This function is documented on page 11.)
```

## 5.9 后备字体

**AutoFallback** 后备字体的宏包选项声明。

```
1312 \keys_define:nn { xeCJK / options }
1313 {
1314   AutoFallback .choice:,
1315   AutoFallback / true .code:n = { \xeCJKenablefallback },
1316   AutoFallback / false .code:n = { \xeCJKdisablefallback },
1317   AutoFallback .default:n = { true },
1318   fallback .meta:n = { AutoFallback = true },
1319 }
(End definition for AutoFallback This function is documented on page 4.)
```

**\xeCJKenablefallback** 后备字体的启用开关, 默认关闭。

```
\xeCJKdisablefallback 1320 \bool_new:N \l__xeCJK_fallback_bool
1321 \NewDocumentCommand \xeCJKenablefallback { }
1322 {
1323   \bool_if:NF \l__xeCJK_fallback_bool
1324   {
1325     \bool_set_true:N \l__xeCJK_fallback_bool
1326     \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1327     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_testsymbol:N
1328   }
1329 }
1330 \NewDocumentCommand \xeCJKdisablefallback { }
1331 {
1332   \bool_if:NT \l__xeCJK_fallback_bool
1333   {
1334     \bool_set_false:N \l__xeCJK_fallback_bool
1335     \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N
1336   }
1337 }
(End definition for \xeCJKenablefallback and \xeCJKdisablefallback)
```

**\xeCJK\_fallback\_testsymbol: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。

```
1338 \cs_new_protected_nopar:Npn \xeCJK_fallback_testsymbol:N #1
1339 {
1340   \font_glyph_if_exist:NnTF \tex_font:D {`#1}
1341   { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1342   {
1343     \xeCJK_family_if_exist:xTF { \l__xeCJK_family_tl/FallBack }
1344     { {
1345       \tl_put_right:Nn \l__xeCJK_family_tl { /FallBack }
1346       \xeCJK_select_font:
1347       \xeCJK_fallback_testsymbol:N {#1}

```

```

1348         } }
1349     {
1350         \__xeCJK_warning:nx { fallback } {#1}
1351         \__xeCJK_fallback_save_CJKsymbol:N {#1}
1352     }
1353 }
1354 }
1355 \__xeCJK_msg_new:nn { fallback }
1356 {
1357     CJKfamily~'\l__xeCJK_family_tl'~
1358     ( \prop_get:Nv \g__xeCJK_family_prop \l__xeCJK_family_tl )~
1359     does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {'#1}).\\
1360 }

```

(End definition for \xeCJK\_fallback\_testsymbol:N)

### \setCJKfallbackfamilyfont

```

\__xeCJK_set_family_fallback:xnn 1361 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1362 {
1363     \exp_args:Nx \tl_if_in:nnTF {#3} { , }
1364     { \xeCJK_set_family_fallback:xnn {#1} {#2} {#3} }
1365     { \xeCJK_set_family:xnn { #1/FallBack } {#2} {#3} }
1366 }
1367 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:xnn #1#2#3
1368 {
1369     \group_begin:
1370     \tl_set:Nx \l__xeCJK_family_fb_tl {#1}
1371     \prop_get:NvNF \g__xeCJK_family_prop \l__xeCJK_family_fb_tl \l__xeCJK_fontname_tl
1372     { \tl_clear:N \l__xeCJK_fontname_tl }
1373     \clist_map_inline:xn {#3}
1374     {
1375         \tl_put_right:Nn \l__xeCJK_family_fb_tl { /FallBack }
1376         \__xeCJK_get_sub_features:nn \l__xeCJK_family_fb_tl {##1}
1377         \clist_put_left:cx { \l__xeCJK_ \l__xeCJK_family_fb_tl _fontfeat_clist } {#2}
1378         \xeCJK_set_family:xcc \l__xeCJK_family_fb_tl
1379         { \l__xeCJK_ \l__xeCJK_family_fb_tl _fontfeat_clist }
1380         { \l__xeCJK_ \l__xeCJK_family_fb_tl _fontname_tl }
1381     }
1382     \group_end:
1383 }

```

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

## 5.10 CJK 字体族声明方式

```

1384 \bool_new:N \g__xeCJK_AutoFakeBold_bool
1385 \bool_new:N \g__xeCJK_AutoFakeSlant_bool
1386 \fp_new:N \g__xeCJK_EmboldenFactor_fp
1387 \fp_new:N \g__xeCJK_SlantFactor_fp

```

AutoFakeBold 伪粗体和伪斜体的宏包选项声明。  
 AutoFakeSlant  
 EmboldenFactor  
 SlantFactor

```

1388 \keys_define:nn { xeCJK / options }
1389 {
1390     AutoFakeBold .choice:,

```

```

1391 AutoFakeBold / true .code:n =
1392 { \bool_set_true:N \g__xeCJK_AutoFakeBold_bool },
1393 AutoFakeBold / false .code:n =
1394 { \bool_set_false:N \g__xeCJK_AutoFakeBold_bool },
1395 AutoFakeBold / unknown .code:n =
1396 {
1397 \bool_set_true:N \g__xeCJK_AutoFakeBold_bool
1398 \fp_set:Nn \g__xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1399 },
1400 AutoFakeBold .default:n = { true },
1401 AutoFakeSlant .choice:,
1402 AutoFakeSlant / true .code:n =
1403 { \bool_set_true:N \g__xeCJK_AutoFakeSlant_bool },
1404 AutoFakeSlant / false .code:n =
1405 { \bool_set_false:N \g__xeCJK_AutoFakeSlant_bool },
1406 AutoFakeSlant / unknown .code:n =
1407 {
1408 \bool_set_true:N \g__xeCJK_AutoFakeSlant_bool
1409 \fp_set:Nn \g__xeCJK_SlantFactor_fp { \l_keys_value_tl }
1410 },
1411 AutoFakeSlant .default:n = { true },
1412 EmboldenFactor .fp_set:N = \g__xeCJK_EmboldenFactor_fp,
1413 SlantFactor .fp_set:N = \g__xeCJK_SlantFactor_fp,
1414 BoldFont .meta:n = { AutoFakeBold = true },
1415 boldfont .meta:n = { AutoFakeBold = true },
1416 SlantFont .meta:n = { AutoFakeSlant = true },
1417 slantfont .meta:n = { AutoFakeSlant = true },
1418 }

```

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

\keys\_set\_known:nxN

```

1419 \cs_generate_variant:Nn \keys_set_known:nnN { nx }

```

(End definition for \keys\_set\_known:nxN)

\\_\_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_clist 1420 \clist_new:N \g__xeCJK_sub_key_clist
1421 \cs_new_protected_nopar:Npn \__xeCJK_new_sub_key:n #1
1422 {
1423 \clist_gput_right:Nx \g__xeCJK_sub_key_clist {#1}
1424 \keys_define:nn { xeCJK / features }
1425 {
1426 #1 .code:n =
1427 {
1428 \xeCJK_if_blank_x:nTF {##1}
1429 {
1430 \bool_set_false:c { l__xeCJK_copy_#1_bool }
1431 \bool_set_false:c { l__xeCJK_add_#1_bool }
1432 \tl_put_right:Nx \l__xeCJK_family_aux_tl { /#1 }
1433 }
1434 {
1435 \clist_put_right:Nx \l__xeCJK_sub_key_clist {#1}
1436 \str_if_eq_x:nnTF {##1} *

```

```

1437         {
1438             \bool_set_true:c { l__xeCJK_copy_#1_bool }
1439             \bool_set_false:c { l__xeCJK_add_#1_bool }
1440         }
1441         {
1442             \bool_set_false:c { l__xeCJK_copy_#1_bool }
1443             \bool_set_true:c { l__xeCJK_add_#1_bool }
1444             \__xeCJK_get_sub_features:nn {#1} {##1}
1445         }
1446     }
1447 },
1448 #1 .default:n = \c_empty_tl,
1449 }
1450 }
1451 \cs_generate_variant:Nn \__xeCJK_new_sub_key:n { x }
1452 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
1453 {
1454     \tl_set:Nx \l__xeCJK_tmpa_tl {#2}
1455     \clist_clear:N \l__xeCJK_sub_fontfeat_clist
1456     \__xeCJK_gobble_brace:N \l__xeCJK_tmpa_tl
1457     \exp_args:No \tl_if_head_eq_charcode:nNTF \l__xeCJK_tmpa_tl [ % ]
1458     {
1459         \exp_after:wN \__xeCJK_get_sub_features_aux:n \l__xeCJK_tmpa_tl \c_empty_tl
1460         \xeCJK_if_blank_x:nT \l__xeCJK_sub_fontname_tl
1461         {
1462             \tl_set:Nx \l__xeCJK_sub_fontname_tl \l__xeCJK_tmpa_tl
1463             \clist_clear:N \l__xeCJK_sub_fontfeat_clist
1464         }
1465     }
1466     { \tl_set:Nx \l__xeCJK_sub_fontname_tl \l__xeCJK_tmpa_tl }
1467     \xeCJK_if_blank_x:nTF \l__xeCJK_sub_fontname_tl
1468     { \tl_set:Nx \l__xeCJK_sub_fontname_tl \l__xeCJK_fontname_tl }
1469     { \tl_replace_all:Nnx \l__xeCJK_sub_fontname_tl * \l__xeCJK_fontname_tl }
1470     \clist_set:cx { l__xeCJK_#1_fontfeat_clist } \l__xeCJK_sub_fontfeat_clist
1471     \tl_set:cx { l__xeCJK_#1_fontname_tl } \l__xeCJK_sub_fontname_tl
1472 }

```

(End definition for \\_\_xeCJK\_new\_sub\_key:n and \g\_\_xeCJK\_sub\_key\_clist)

```

\__xeCJK_gobble_brace:N
\__xeCJK_get_sub_features:nn 1473 \cs_new_protected_nopar:Npn \__xeCJK_gobble_brace:N #1
\__xeCJK_get_sub_features_aux:n 1474 {
1475     \exp_args:No \tl_if_head_is_group:nT #1
1476     {
1477         \exp_last_unbraced:NNo \tl_set:Nn #1 #1
1478         \__xeCJK_gobble_brace:N #1
1479     }
1480 }
1481 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features_aux:n [#1] #2
1482 {
1483     \clist_set:Nx \l__xeCJK_sub_fontfeat_clist {#1}
1484     \tl_set:Nx \l__xeCJK_sub_fontname_tl {#2}
1485 }

```

(End definition for \\_\_xeCJK\_gobble\_brace:N, \\_\_xeCJK\_get\_sub\_features:nn, and \\_\_xeCJK\_get\_sub\_features\_aux:n)

### FallBack

```
1486 \__xeCJK_new_sub_key:n { FallBack }
```

(End definition for FallBack This function is documented on page 8.)

\g\_\_xeCJK\_features\_id\_prop 调用字体的属性声明, 同 fontspec 宏包。

```
      BoldFont 1487 \cs_new_nopar:Npn \__xeCJK_map_features_id:n #1
      ItalicFont 1488 { \prop_get:Nn \g__xeCJK_features_id_prop {#1} }
      BoldItalicFont 1489 \prop_new:N \g__xeCJK_features_id_prop
      SlantedFont 1490 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
      BoldSlantedFont 1491 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
      BoldFeatures 1492 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
      ItalicFeatures 1493 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
      BoldItalicFeatures 1494 \prop_put:Nnn \g__xeCJK_features_id_prop { bfsl } { BoldSlanted }
      SlantedFeatures 1495 \prop_map_inline:Nn \g__xeCJK_features_id_prop
      BoldSlantedFeatures 1496 {
1497   \keys_define:nn { xeCJK / features }
1498   {
1499     #2Font .tl_set_x:c = { l__xeCJK_fontname_#1_tl },
1500     #2Features .tl_set_x:c = { l__xeCJK_fontfeat_#1_clist } ,
1501   }
1502 }
```

(End definition for \g\_\_xeCJK\_features\_id\_prop and others.)

### AutoFakeBold

### AutoFakeSlant

### Mono

```
1503 \keys_define:nn { xeCJK / features }
1504 {
1505   AutoFakeBold .choice:,
1506   AutoFakeBold / false .code:n =
1507     { \bool_set_false:N \l__xeCJK_AutoFakeBold_bool },
1508   AutoFakeBold / unknown .code:n =
1509     {
1510       \bool_set_true:N \l__xeCJK_AutoFakeBold_bool
1511       \fp_set:Nn \l__xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1512     },
1513   AutoFakeBold .default:n = { \g__xeCJK_EmboldenFactor_fp },
1514   AutoFakeSlant .choice:,
1515   AutoFakeSlant / false .code:n =
1516     { \bool_set_false:N \l__xeCJK_AutoFakeSlant_bool },
1517   AutoFakeSlant / unknown .code:n =
1518     {
1519       \bool_set_true:N \l__xeCJK_AutoFakeSlant_bool
1520       \fp_set:Nn \l__xeCJK_SlantFactor_fp { \l_keys_value_tl }
1521     },
1522   AutoFakeSlant .default:n = { \g__xeCJK_SlantFactor_fp },
1523   Mono .choice:,
1524   Mono / Exspace .code:n =
1525     {
1526       \xeCJK_set_monoexspace:
1527       \tl_clear:N \l__xeCJK_monoscale_tl
1528     },

```

```

1529 Mono / Scale .code:n =
1530 {
1531     \xeCJK_set_monoscale:
1532     \tl_set:Nx \l__xeCJK_monoscale_tl { Scale = { \fp_use:N \g__xeCJK_monoscale_fp } }
1533 }
1534 }

```

(End definition for AutoFakeBold, AutoFakeSlant, and Mono)

\\_\_xeCJK\_set\_init:

```

1535 \cs_new_protected_nopar:Npn \__xeCJK_set_init:
1536 {
1537     \cs_set_eq:NN \CJKfamily \use_none:n
1538     \tl_clear:N \l__xeCJK_monoscale_tl
1539     \int_gincr:N \g__xeCJK_fontspec_int
1540     \prop_clear:N \l__xeCJK_add_font_prop
1541     \prop_map_inline:Nn \g__xeCJK_features_id_prop
1542     {
1543         \tl_clear:c { l__xeCJK_fontname_##1_tl }
1544         \clist_clear:c { l__xeCJK_fontfeat_##1_clist }
1545     }
1546     \clist_clear:N \l__xeCJK_pass_features_clist
1547     \clist_clear:N \l__xeCJK_sub_key_clist
1548     \clist_map_inline:Nn \g__xeCJK_sub_key_clist
1549     {
1550         \bool_set_false:c { l__xeCJK_copy_##1_bool }
1551         \bool_set_false:c { l__xeCJK_add_##1_bool }
1552     }
1553     \bool_set_eq:NN \l__xeCJK_AutoFakeBold_bool \g__xeCJK_AutoFakeBold_bool
1554     \bool_set_eq:NN \l__xeCJK_AutoFakeSlant_bool \g__xeCJK_AutoFakeSlant_bool
1555     \fp_set_eq:NN \l__xeCJK_EmboldenFactor_fp \g__xeCJK_EmboldenFactor_fp
1556     \fp_set_eq:NN \l__xeCJK_SlantFactor_fp \g__xeCJK_SlantFactor_fp
1557 }

```

(End definition for \\_\_xeCJK\_set\_init:)

\xeCJK\_set\_family:xnn 设置一个 CJK 新字体族,与 \newfontfamily 类似,增加 FallBack 选项。

```

1558 \cs_new_protected_nopar:Npn \xeCJK_set_family:xnn #1#2#3
1559 {
1560     \group_begin:
1561     \__xeCJK_set_init:
1562     \tl_set:Nx \l__xeCJK_family_aux_tl {#1}
1563     \clist_set:Nx \l__xeCJK_fontoptions_clist {#2}
1564     \tl_set:Nx \l__xeCJK_fontname_tl {#3}
1565     \keys_set_known:nxN { xeCJK / features }
1566     { \g__xeCJK_default_features_clist, #2 } \l__xeCJK_pass_features_clist
1567     \__xeCJK_parse_features:
1568     \__xeCJK_pass_features:
1569     \__xeCJK_check_family:V \l__xeCJK_family_aux_tl
1570     \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_family_aux_tl }
1571     {
1572         \group_begin:
1573         \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1574         \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }

```

```

1575     { \l__xeCJK_pass_features_clist } { \l__xeCJK_fontname_tl }
1576     \prop_gput:NnV \exp_not:N \g__xeCJK_family_name_prop
1577     { \l__xeCJK_family_aux_tl } \exp_not:N \l__xeCJK_fontspec_family_tl
1578     \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_family_aux_tl }
1579     \exp_not:n
1580     { {
1581         \exp_not:N \fontencoding { \c__xeCJK_encoding_tl }
1582         \exp_not:N \fontfamily { \l__xeCJK_fontspec_family_tl }
1583         \exp_not:N \selectfont
1584     } }
1585     \group_end:
1586 }
1587 \__xeCJK_save_family_info:
1588 \__xeCJK_add_sub_block:
1589 \group_end:
1590 }
1591 \cs_generate_variant:Nn \xeCJK_set_family:xnn { xcc }

```

*(End definition for \xeCJK\_set\_family:xnn)*

\\_\_xeCJK\_check\_family:n

```

1592 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
1593 {
1594     \prop_get:NnNT \g__xeCJK_family_prop {#1} \l__xeCJK_tmpa_tl
1595     {
1596         \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl { }
1597         \__xeCJK_warning:nxx { CJKfamily-redef } {#1} \l__xeCJK_tmpa_tl
1598     }
1599 }
1600 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
1601 \__xeCJK_msg_new:nn { CJKfamily-redef } { Redefining~CJKfamily~`#1'~(#2). }

```

*(End definition for \\_\_xeCJK\_check\_family:n)*

\\_\_xeCJK\_add\_font:nn

\\_\_xeCJK\_add\_font\_if\_new:nn

```

1602 \cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
1603 {
1604     \prop_put:Nxx \l__xeCJK_add_font_prop
1605     { \__xeCJK_map_features_id:n {#1} Font } {#2}
1606 }
1607 \cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
1608 {
1609     \prop_put_if_new:Nxx \l__xeCJK_add_font_prop
1610     { \__xeCJK_map_features_id:n {#1} Font } {#2}
1611 }
1612 \cs_generate_variant:Nn \__xeCJK_add_font:nn { nx, nc, nV, nv }
1613 \cs_generate_variant:Nn \__xeCJK_add_font_if_new:nn { nx, nc, nV, nv }

```

*(End definition for \\_\_xeCJK\_add\_font:nn and \\_\_xeCJK\_add\_font\_if\_new:nn)*

```

1614 \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1615 \cs_generate_variant:Nn \prop_get:Nn { NV }
1616 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1617 \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1618 \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

```

\__xeCJK_add_fake_bold:n
\__xeCJK_add_fake_slant:n 1619 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
1620 {
1621   \clist_put_left:cx { l__xeCJK_fontfeat_#1_clist }
1622   { FakeBold = { \fp_use:N \l__xeCJK_EmboldenFactor_fp } }
1623 }
1624 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
1625 {
1626   \clist_put_left:cx { l__xeCJK_fontfeat_#1_clist }
1627   { FakeSlant = { \fp_use:N \l__xeCJK_SlantFactor_fp } }
1628 }
(End definition for \__xeCJK_add_fake_bold:n and \__xeCJK_add_fake_slant:n)

```

```

\__xeCJK_parse_features:
\__xeCJK_parse_features:n 1629 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:
1630 { \prop_map_inline:Nn \g__xeCJK_features_id_prop { \__xeCJK_parse_features:n {##1} } }
1631 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:n #1
1632 {
1633   \__xeCJK_if_font_select:nTF {#1}
1634   {
1635     \__xeCJK_add_font:nv {#1} { l__xeCJK_fontname_#1_tl }
1636     \__xeCJK_if_it_or_sl:nTF {#1}
1637     {
1638       \__xeCJK_if_font_select:nF {bf#1}
1639       { \__xeCJK_add_font_if_new:nv {bf#1} { l__xeCJK_fontname_#1_tl } }
1640     }
1641     {
1642       \str_if_eq_x:nnT {#1} {bf}
1643       {
1644         \clist_map_inline:nn { it , sl }
1645         {
1646           \__xeCJK_if_font_select:nF {bf##1}
1647           { \__xeCJK_add_font:nV {bf##1} \l__xeCJK_fontname_bf_tl }
1648         }
1649       }
1650     }
1651   }
1652   { \__xeCJK_set_fake:n {#1} }
1653 }
(End definition for \__xeCJK_parse_features: and \__xeCJK_parse_features:n)

```

```

\__xeCJK_if_font_select_p:n
\__xeCJK_if_it_or_sl_p:n 1654 \prg_new_conditional:Npnn \__xeCJK_if_font_select:n #1 { p, T, F, TF }
\__xeCJK_if_font_select:nTF 1655 {
\__xeCJK_if_it_or_sl:nTF 1656   \exp_args:Nc \xeCJK_if_blank_x:nTF { l__xeCJK_fontname_#1_tl }
1657   \prg_return_false: \prg_return_true:
1658 }
1659 \prg_new_conditional:Npnn \__xeCJK_if_it_or_sl:n #1 { p, T, F, TF }
1660 {
1661   \bool_if:nTF { \str_if_eq_x_p:nn { it } {#1} || \str_if_eq_x_p:nn { sl } {#1} }
1662   \prg_return_true: \prg_return_false:
1663 }

```

(End definition for \\_xeCJK\_if\_font\_select:n and \\_xeCJK\_if\_it\_or\_sl:n)

\\_xeCJK\_set\_fake:n

```
1664 \cs_new_protected_nopar:Npn \_xeCJK_set_fake:n #1
1665 {
1666   \str_if_eq_x:nnTF {#1} { bf }
1667   { \bool_if:NT \l__xeCJK_AutoFakeBold_bool { \_xeCJK_add_fake_bold:n {#1} } }
1668   {
1669     \bool_if:NTF \l__xeCJK_AutoFakeSlant_bool
1670     {
1671       \bool_if:nT
1672       { \_xeCJK_if_it_or_sl_p:n {#1} ||
1673         ( \str_if_eq_x_p:nn {#1} {bf} ) && ! ( \_xeCJK_if_font_select_p:n {it} ) ) ||
1674         ( \str_if_eq_x_p:nn {#1} {bfsl} ) && ! ( \_xeCJK_if_font_select_p:n {sl} ) )
1675       }
1676       { \_xeCJK_add_fake_slant:n {#1} }
1677     }
1678     { \_xeCJK_if_it_or_sl:nT {#1} { \_xeCJK_map_it_sl:n {#1} } }
1679     \bool_if:nT
1680     { \l__xeCJK_AutoFakeBold_bool
1681       && ! ( \_xeCJK_if_it_or_sl_p:n {#1} )
1682       && ! ( \_xeCJK_if_font_select_p:n {bf} )
1683     }
1684     { \_xeCJK_add_fake_bold:n {#1} }
1685   }
1686   \_xeCJK_add_font_if_new:nn {#1} *
1687 }
```

(End definition for \\_xeCJK\_set\_fake:n)

\\_xeCJK\_map\_it\_sl:n

```
1688 \cs_new_protected_nopar:Npn \_xeCJK_map_it_sl:n #1
1689 {
1690   \_xeCJK_if_map_font_select:nT {#1}
1691   {
1692     \_xeCJK_add_font:nx {#1} { \_xeCJK_get_map_font:n {#1} }
1693     \_xeCJK_if_font_select:nF {bf#1}
1694     { \_xeCJK_add_font_if_new:nx {bf#1} { \_xeCJK_get_map_font:n {#1} } }
1695   }
1696 }
```

(End definition for \\_xeCJK\_map\_it\_sl:n)

\\_xeCJK\_get\_map\_font:n

```
1697 \cs_new_nopar:Npn \_xeCJK_get_map_font:n #1
1698 { \tl_use:c { l__xeCJK_fontname_\str_if_eq_x:nnTF {#1} {it} {sl} {it} _tl } }
(End definition for \_xeCJK_get_map_font:n)
```

\\_xeCJK\_if\_map\_font\_select\_p:n

```
\_xeCJK_if_map_font_select:nTF 1699 \prg_new_conditional:Npnn \_xeCJK_if_map_font_select:n #1 { p, T, F, TF }
1700 {
1701   \xeCJK_if_blank_x:nTF { \_xeCJK_get_map_font:n {#1} }
1702   \prg_return_false: \prg_return_true:
1703 }
```

(End definition for \\_\_xCJK\_if\_map\_font\_select:n)

\\_\_xCJK\_pass\_features:

```
1704 \cs_new_protected_nopar:Npn \__xCJK_pass_features:
1705 {
1706   \prop_map_inline:Nn \g__xCJK_features_id_prop
1707   {
1708     \clist_if_empty:cF { l__xCJK_fontfeat_##1_clist }
1709     {
1710       \clist_put_right:Nx \l__xCJK_pass_features_clist
1711       { ##2Features = { \tl_use:c { l__xCJK_fontfeat_##1_clist } } }
1712     }
1713   }
1714   \prop_map_inline:Nn \l__xCJK_add_font_prop
1715   { \clist_put_right:Nx \l__xCJK_pass_features_clist { ##1 = { ##2 } } }
1716   \xCJK_if_blank_x:nF \l__xCJK_monoscale_tl
1717   { \clist_put_left:Nx \l__xCJK_pass_features_clist \l__xCJK_monoscale_tl }
1718 }
```

(End definition for \\_\_xCJK\_pass\_features:)

\g\_\_xCJK\_family\_prop

\g\_\_xCJK\_family\_name\_prop

\g\_\_xCJK\_family\_options\_prop

```
1719 \prop_new:N \g__xCJK_family_prop
1720 \prop_new:N \g__xCJK_family_name_prop
1721 \prop_new:N \g__xCJK_family_options_prop
```

(End definition for \g\_\_xCJK\_family\_prop, \g\_\_xCJK\_family\_name\_prop, and \g\_\_xCJK\_family\_options\_prop)

\\_\_xCJK\_save\_family\_info:

```
1722 \cs_new_protected_nopar:Npn \__xCJK_save_family_info:
1723 {
1724   \group_begin:
1725   \clist_map_inline:Nn \g__xCJK_CJK_subclass_clist
1726   {
1727     \clist_remove_all:Nn \l__xCJK_fontoptions_clist {##1}
1728     \clist_remove_all:Nn \l__xCJK_fontoptions_clist { ##1 = }
1729   }
1730   \prop_gput:NVV \g__xCJK_family_prop \l__xCJK_family_aux_tl \l__xCJK_fontname_tl
1731   \prop_gput:NVV \g__xCJK_family_options_prop \l__xCJK_family_aux_tl \l__xCJK_fontoptions_clist
1732   \group_end:
1733 }
```

(End definition for \\_\_xCJK\_save\_family\_info:)

\\_\_xCJK\_add\_sub\_block:

```
1734 \cs_new_protected_nopar:Npn \__xCJK_add_sub_block:
1735 {
1736   \clist_remove_duplicates:N \l__xCJK_sub_key_clist
1737   \clist_map_inline:Nn \l__xCJK_sub_key_clist
1738   {
1739     \tl_set:Nx \l__xCJK_sub_family_tl { \l__xCJK_family_aux_tl/##1 }
1740     \bool_if:cT { l__xCJK_copy_##1_bool }
1741     {
1742       \__xCJK_check_family:V \l__xCJK_sub_family_tl
1743       \prop_get:NVNT \g__xCJK_family_prop \l__xCJK_family_aux_tl \l__xCJK_tmpa_tl
```

```

1744     { \prop_gput:NVV \g__xeCJK_family_prop \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl }
1745 \prop_get:NVNT \g__xeCJK_family_options_prop \l__xeCJK_family_aux_tl \l__xeCJK_tmpa_clist
1746     {
1747         \clist_remove_all:Nn \l__xeCJK_tmpa_clist { ##1 = * }
1748     \prop_gput:NVV \g__xeCJK_family_options_prop \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
1749     }
1750     \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_sub_family_tl }
1751     {
1752         \xeCJK_family_if_exist:xT { \l__xeCJK_family_aux_tl }
1753         {
1754             \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
1755             { \l__xeCJK_family_aux_tl } \exp_not:N \l__xeCJK_tmpa_tl
1756             {
1757                 \prop_gput:NnV \exp_not:N \g__xeCJK_family_name_prop
1758                 { \l__xeCJK_sub_family_tl } \exp_not:N \l__xeCJK_tmpa_tl
1759             \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_sub_family_tl }
1760             \exp_not:n
1761             { {
1762                 \exp_not:N \fontencoding { \c__xeCJK_encoding_tl }
1763                 \exp_not:N \fontfamily { \l__xeCJK_tmpa_tl }
1764                 \exp_not:N \selectfont
1765             } }
1766         }
1767     }
1768 }
1769 }
1770 \bool_if:cT { l__xeCJK_add_##1_bool }
1771 {
1772     \xeCJK_set_family:xcc \l__xeCJK_sub_family_tl
1773     { l__xeCJK_##1_fontfeat_clist } { l__xeCJK_##1_fontname_tl }
1774 }
1775 }
1776 }
(End definition for \__xeCJK_add_sub_block:)

```

\\_\_xeCJK\_copy\_family:nn

```

1777 \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
1778 {
1779     \xeCJK_family_if_exist:xT {#2}
1780     {
1781         \tl_map_inline:nn
1782         { \g__xeCJK_family_prop \g__xeCJK_family_options_prop \g__xeCJK_family_name_prop }
1783         { \prop_get:NnNT ##1 {#2} \l__xeCJK_tmpa_tl { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_tl } }
1784         \cs_gset_eq:cc { __xeCJK/family/#1 } { __xeCJK/family/#2 }
1785     }
1786 }
1787 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }
(End definition for \__xeCJK_copy_family:nn)

```

## 5.11 字体切换

\l\_xeCJK\_font\_coor\_tl 缓存当前字体的原始格式,以加速编译。  
\xeCJK\_select\_font:  
\\_\_xeCJK\_switch\_font:nn

```

1788 \tl_set:Nn \l_xeCJK_font_coor_tl { __xeCJK/\l_xeCJK_family_tl/\f@series/\f@shape/\f@size }
1789 \cs_new_protected_nopar:Npn \xeCJK_select_font:
1790 {
1791   \cs_if_exist_use:cF { \l_xeCJK_font_coor_tl }
1792   {
1793     \tl_set:Nx \l__xeCJK_current_coor_tl \l_xeCJK_font_coor_tl
1794     \cs_if_exist_use:cT { __xeCJK/family/\l_xeCJK_family_tl }
1795     { \exp_last_unbraced:NNV \cs_gset_eq:cN \l__xeCJK_current_coor_tl \tex_font:D }
1796   }
1797 }
1798 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
1799 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
1800 {
1801   \str_if_eq_x:nnF {#1} {#2}
1802   {
1803     \__xeCJK_info:nxx { CJK-block } {#1} {#2}
1804     \tl_remove_all:Nn \l_xeCJK_family_tl { /#1 }
1805     \str_if_eq_x:nnF {#2} { CJK }
1806     {
1807       \tl_set:Nx \l__xeCJK_tmpa_tl \l_xeCJK_family_tl
1808       \tl_put_right:Nx \l_xeCJK_family_tl { /#2 }
1809       \xeCJK_family_if_exist:xF \l_xeCJK_family_tl
1810       {
1811         \xeCJK_family_if_exist:xF { \CJKfamilydefault/#2 }
1812         { \__xeCJK_copy_family:xx \l_xeCJK_family_tl { \CJKfamilydefault/#2 } }
1813         { \__xeCJK_copy_family:xx \l_xeCJK_family_tl \l__xeCJK_tmpa_tl }
1814       }
1815     }
1816   }
1817   \xeCJK_select_font:
1818 }
1819 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~'#1'~to~'#2'. }
(End definition for \l_xeCJK_font_coor_tl, \xeCJK_select_font:, and \__xeCJK_switch_font:nn)

```

\xeCJK\_family\_if\_exist:xTF

```

1820 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T, F, TF }
1821 {
1822   \cs_if_free:cTF { __xeCJK/family/#1 }
1823   { \prg_return_false: }
1824   {
1825     \prop_get:NxNF \g__xeCJK_family_name_prop {#1} \l__xeCJK_family_name_tl
1826     { \use:c { __xeCJK/family/#1 } }
1827     \prg_return_true:
1828   }
1829 }
1830 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
(End definition for \xeCJK_family_if_exist:xTF)

```

**\CJKfamily** 用于切换 CJK 字体族。

```

1831 \NewDocumentCommand \CJKfamily { t+ t- m }
1832 {
1833   \xeCJK_if_blank_x:nTF {#3}

```

```

1834 {
1835   \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
1836   \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
1837 }
1838 {
1839   \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
1840   {
1841     \xeCJK_family_if_exist:xTF {#3}
1842     {
1843       \tl_set:Nx \l_xeCJK_family_tl {#3}
1844       \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
1845       \IfBooleanT {#1} { \use:c { __xeCJK/family/#3 } }
1846     }
1847     { \__xeCJK_family_unknown_warning:x {#3} }
1848   }
1849 }
1850 \tex_ignorespaces:D
1851 }

```

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

\xeCJK\_family\_if\_exist\_use:x

```

\__xeCJK_family_unknown_warning:x
\g__xeCJK_unknown_family_seq
1852 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
1853 {
1854   \xeCJK_family_if_exist:xTF {#1}
1855   { \use:c { __xeCJK/family/#1 } }
1856   { \__xeCJK_family_unknown_warning:x {#1} }
1857 }
1858 \cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:x #1
1859 {
1860   \seq_if_in:NxF \g__xeCJK_unknown_family_seq {#1}
1861   {
1862     \seq_gput_right:Nx \g__xeCJK_unknown_family_seq {#1}
1863     \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
1864   }
1865 }
1866 \seq_new:N \g__xeCJK_unknown_family_seq
1867 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
1868 {
1869   Unknown~CJK~family~'
1870   \str_case_x:nnn {#1}
1871   {
1872     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1873     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1874     \CJKttdefault { \token_to_str:N \CJKttdefault }
1875   }
1876   {#1}
1877   '~is~ignored.\\\
1878   Try~to~use~
1879   \str_case_x:nnn {#1}
1880   {
1881     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1882     \CJKsfdefault { \token_to_str:N \setCJKsansfont }

```

```

1883     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1884   }
1885   { \token_to_str:N \setCJKfamilyfont {#1} }
1886   [...]{}~to~define~it.
1887 }

```

(End definition for \xeCJK\_family\_if\_exist\_use:x, \\_\_xeCJK\_family\_unknown\_warning:x, and \g\_\_xeCJK\_unknown\_family\_seq  
These functions are documented on page ??.)

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

```

1888 \newCJKfontfamily 1888 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
1889   { \xeCJK_set_family:xnn {#1} {#2} {#3} }
1890 \CJKfontspec 1890 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1891   {
1892     \tl_set:Nx \l__xeCJK_family_aux_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1893     \xeCJK_set_family:xnn \l__xeCJK_family_aux_tl {#3} {#4}
1894     \cs_new_protected_nopar:Npx #2 { \exp_not:N \CJKfamily { \l__xeCJK_family_aux_tl } }
1895   }
1896 \int_new:N \g__xeCJK_fontspec_int
1897 \int_gdecr:N \g__xeCJK_fontspec_int
1898 \NewDocumentCommand \CJKfontspec { O{} m }
1899   {
1900     \tl_set:Nx \l__xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1901     \cs_if_free:cT \l__xeCJK_CJKfontspec_id_tl
1902     {
1903       \int_gincr:N \g__xeCJK_fontspec_int
1904       \tl_gset:Nx \g__xeCJK_family_spec_tl
1905         { CJKfontspec ( \int_use:N \g__xeCJK_fontspec_int ) }
1906       \xeCJK_set_family:xnn \g__xeCJK_family_spec_tl {#1} {#2}
1907       \tl_gset:cx \l__xeCJK_CJKfontspec_id_tl \g__xeCJK_family_spec_tl
1908     }
1909     \exp_args:Nv \CJKfamily \l__xeCJK_CJKfontspec_id_tl
1910     \tex_ignorespaces:D
1911   }

```

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

**\defaultCJKfontfeatures** 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。

```

1912 \addCJKfontfeatures 1912 \clist_new:N \g__xeCJK_default_features_clist
1913 \NewDocumentCommand \defaultCJKfontfeatures { m }
1914   { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
1915 \defaultCJKfontfeatures { Script = CJK }
1916 \@onlypreamble \defaultCJKfontfeatures
1917 \NewDocumentCommand \addCJKfontfeatures { m }
1918   {
1919     \prop_if_in:NVTF \g__xeCJK_family_prop \l__xeCJK_family_tl
1920     {
1921       \group_begin:
1922       \clist_set:Nx \l__xeCJK_tmpa_clist {#1}
1923       \prop_get:NvN \g__xeCJK_family_prop \l__xeCJK_family_tl \l__xeCJK_tmpa_tl
1924       \clist_map_inline:Nn \g__xeCJK_CJK_subclass_clist
1925         {
1926           \clist_if_in:NnT \l__xeCJK_tmpa_clist {##1}
1927           {

```

```

1928         \clist_remove_all:Nn \l__xeCJK_tmpa_clist {##1}
1929     \prop_get:NxNF \g__xeCJK_family_prop { \l__xeCJK_family_tl/##1 } \l__xeCJK_tmpb_tl
1930     {
1931         \prop_get:NxNF \g__xeCJK_family_prop
1932         { \CJKfamilydefault/##1 } \l__xeCJK_tmpb_tl
1933         { \tl_set:Nx \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
1934     }
1935     \clist_set:Nx \l__xeCJK_tmpa_clist
1936     { ##1 = { [ \l__xeCJK_tmpa_clist ] { \l__xeCJK_tmpb_tl } } }
1937     \clist_map_break:
1938 }
1939 }
1940 \prop_get:NVNF \g__xeCJK_family_options_prop \l__xeCJK_family_tl \l__xeCJK_tmpb_clist
1941 { \clist_clear:N \l__xeCJK_tmpb_clist }
1942 \clist_clear:N \l__xeCJK_tmpc_clist
1943 \clist_map_inline:Nn \l__xeCJK_tmpb_clist
1944 {
1945     \clist_if_in:NnF \l__xeCJK_tmpa_clist {##1}
1946     { \clist_put_right:Nn \l__xeCJK_tmpc_clist {##1} }
1947 }
1948 \clist_put_left:NV \l__xeCJK_tmpa_clist \l__xeCJK_tmpc_clist
1949 \use:x
1950 { \group_end: \CJKfontspec [ \l__xeCJK_tmpa_clist ] { \l__xeCJK_tmpa_tl } }
1951 }
1952 { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
1953 \tex_ignorespaces:D
1954 }
1955 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
1956 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1957 {
1958     \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
1959     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1960 }

```

(End definition for \defaultCJKfontfeatures and \addCJKfontfeatures These functions are documented on page 7.)

**\setCJKmainfont** 设置文档的 CJK 普通字体、无衬线字体。

```

\setCJKsansfont 1961 \NewDocumentCommand \setCJKmainfont { O{ } m }
1962 { \xeCJK_set_family:xnn \CJKrmdefault {#1} {#2} }
1963 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1964 \NewDocumentCommand \setCJKsansfont { O{ } m }
1965 { \xeCJK_set_family:xnn \CJKsfdefault {#1} {#2} }

```

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

```

\CJKrmdefault
\CJKsfdefault 1966 \tl_if_exist:NF \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
\CJKttdefault 1967 \tl_if_exist:NF \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 1968 \tl_if_exist:NF \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1969 \tl_if_exist:NF \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1970 \tl_new:c { __xeCJK/family/\CJKfamilydefault }

```

(End definition for \CJKrmdefault and others. These functions are documented on page 7.)

补丁 \normalfont, \rmfamily, \n 和 \ttfamily, 使其同时对 CJK 字体族有效。

```

1971 \AtEndOfPackage

```

```

1972 {
1973   \etex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1974   \etex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }
1975   \etex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1976   \etex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1977   \cs_set_eq:NN \reset@font \normalfont
1978 }

```

为了保证 CJK 数学字体的正确设置,应在导言区内设置 CJK 字体。

```

1979 \@onlypreamble \setCJKmainfont
1980 \@onlypreamble \setCJKmathfont
1981 \@onlypreamble \setCJKsansfont
1982 \@onlypreamble \setCJKmonofont
1983 \@onlypreamble \setCJKromanfont

```

## 5.12 数学字体设置

**\setCJKmathfont** 设置 CJK 数学字体。

```

1984 \NewDocumentCommand \setCJKmathfont { O{} m }
1985 { \xeCJK_set_family:xnn \c__xeCJK_math_tl {#1} {#2} }

```

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

**CJKmath** 数学公式内的 CJK 字体。

```

1986 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }
1987 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }

```

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

**\xeCJKsetmathcode**

```

1988 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1989 {
1990   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
1991   \int_set:Nn \l__xeCJK_tmpc_int { \__xeCJK_math_type:n {#3} }
1992   \int_set:Nn \l_tmpa_int { \use:c { sym #4 } }
1993   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
1994   {
1995     \XeTeXmathcode \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int \l_tmpa_int \l__xeCJK_tmpa_int
1996     \int_incr:N \l__xeCJK_tmpa_int
1997   }
1998 }
1999 \cs_new_eq:NN \__xeCJK_math_type:n \mathchar@type

```

(End definition for \xeCJKsetmathcode)

**\xeCJK\_set\_mathfont:**

```

2000 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
2001 {
2002   \xeCJK_family_if_exist:xF \c__xeCJK_math_tl
2003   { \tl_set:Nx \l__xeCJK_tmpa_tl \c__xeCJK_math_tl }
2004   {
2005     \xeCJK_family_if_exist:xF \CJKfamilydefault
2006     { \tl_set:Nx \l__xeCJK_tmpa_tl \CJKfamilydefault }
2007     { \tl_set_eq:NN \l__xeCJK_tmpa_tl \q_no_value }
2008   }

```

```

2009 \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_math_family_tl
2010 {
2011     \DeclareSymbolFont \c__xeCJK_math_tl \c__xeCJK_encoding_tl
2012     \l__xeCJK_math_family_tl \mddefault \shapedefault
2013     \SetSymbolFont \c__xeCJK_math_tl { bold } \c__xeCJK_encoding_tl
2014     \l__xeCJK_math_family_tl \bfdefault \shapedefault
2015     \clist_map_inline:nn { CJK, FullLeft, FullRight }
2016     {
2017         \clist_map_inline:cn { g__xeCJK_##1_range_clist }
2018         {
2019             \__xeCJK_set_char_class_aux:Nn \xeCJKsetmathcode {####1}
2020             \mathalpha \c__xeCJK_math_tl
2021         }
2022     }
2023 }
2024 }

```

(End definition for \xeCJK\_set\_mathfont:)

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

```

2025 \__xeCJK_AtEndPreamble:n
2026 {
2027     \normalfont
2028     \prop_if_empty:NTF \g__xeCJK_family_prop
2029     { \__xeCJK_warning:n { no-CJKfamily } }
2030     { \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: } }
2031 }
2032 \__xeCJK_msg_new:nn { no-CJKfamily }
2033 {
2034     It~seems~that~you~have~not~declare~a~CJKfamily.\\
2035     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
2036     \token_to_str:N \xeCJKmainfont[...]{...}\\\\
2037     in~the~preamble~to~declare~the~main~CJKfamily.\\
2038 }

```

## 5.13 处理等宽字体和抄录环境

```

\g__xeCJK_monoscale_fp
\g__xeCJK_exspace_dim

```

```

2039 \fp_new:N \g__xeCJK_monoscale_fp
2040 \fp_set_eq:NN \g__xeCJK_monoscale_fp \c_one_fp
2041 \dim_new:N \g__xeCJK_exspace_dim
(End definition for \g__xeCJK_monoscale_fp and \g__xeCJK_exspace_dim)

```

**\setCJKmonoscale**

```

2042 \NewDocumentCommand \setCJKmonoscale { }
2043 {
2044     \CJKflexiblespacing
2045     \xeCJK_set_monoscale:
2046     \addCJKfontfeatures { Scale = \fp_use:N \g__xeCJK_monoscale_fp }
2047 }
2048 \cs_new_protected_nopar:Npn \xeCJK_set_monoscale:
2049 {
2050     \group_begin:

```

```

2051     \fontfamily \ttdefault \selectfont
2052     \fp_gset:Nn \g__xeCJK_monoscale_fp
2053     { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } / ( \f@size ) }
2054   \group_end:
2055 }

```

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

#### \setCJKmonoexspace

```

2056 \NewDocumentCommand \setCJKmonoexspace { } { \xeCJK_set_monoexspace: }
2057 \cs_new_protected_nopar:Npn \xeCJK_set_monoexspace:
2058 {
2059   \group_begin:
2060   \fontfamily \ttdefault \selectfont
2061   \dim_gset:Nn \g__xeCJK_exspace_dim
2062   {
2063     \c_two \tex_fontdimen:D \c_two \tex_font:D
2064     - \fp_to_dim:n { \g__xeCJK_monoscale_fp * ( \f@size ) }
2065   }
2066   \group_end:
2067 }

```

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

#### \CJKfixedspacing

#### \CJKflexiblespacing

```

2068 \NewDocumentCommand \CJKfixedspacing { }
2069 {
2070   \bool_if:NF \l__xeCJK_fixed_spacing_bool
2071   {
2072     \bool_set_true:N \l__xeCJK_fixed_spacing_bool
2073     \tl_set_eq:NN \l__xeCJK_flexible_punctstyle_tl \l__xeCJK_punctstyle_tl
2074     \bool_set_eq:NN \l__xeCJK_fixed_xecglue_bool \l__xeCJK_xecglue_bool
2075     \cs_set_eq:NN \__xeCJK_flexible_ecglue: \CJKecglue
2076     \cs_set_eq:NN \__xeCJK_flexible_cjkgglue: \CJKglue
2077     \xeCJK_set_monoexspace:
2078     \xeCJKsetup
2079     {
2080       PunctStyle = plain ,
2081       CJKglue    = { \skip_horizontal:N \g__xeCJK_exspace_dim } ,
2082       CJKecglue  = { \skip_horizontal:n { .5\g__xeCJK_exspace_dim } } ,
2083       xCJKecglue = false ,
2084     }
2085   }
2086 }
2087 \NewDocumentCommand \CJKflexiblespacing { }
2088 {
2089   \bool_if:NT \l__xeCJK_fixed_spacing_bool
2090   {
2091     \bool_set_false:N \l__xeCJK_fixed_spacing_bool
2092     \exp_args:Nx \xeCJKsetup
2093     {
2094       PunctStyle = { \l__xeCJK_flexible_punctstyle_tl } ,
2095       CJKglue    = { \exp_not:o { \__xeCJK_flexible_cjkgglue: } } ,
2096       CJKecglue  = { \exp_not:o { \__xeCJK_flexible_ecglue: } } ,

```

```

2097         xCJKecglue = { \bool_if:NTF \l__xeCJK_fixed_xecglue_bool { true } { false } } ,
2098     }
2099 }
2100 }
2101 \bool_new:N \l__xeCJK_fixed_spacing_bool
2102 \__xeCJK_AfterPreamble:n { \tl_put_right:Nn \verbatim@font \CJKfixedspacing }
(End definition for \CJKfixedspacing and \CJKflexiblespacing These functions are documented on page 6.)

```

**\setCJKmonofont** 设置文档的 CJK 等宽字体族。

```

2103 \NewDocumentCommand \setCJKmonofont { s t+ O{ } m }
2104 {
2105     \IfBooleanTF {#1}
2106     { \xeCJK_set_family:xnn \CJKttdefault { Mono = Scale, #3 } {#4} }
2107     {
2108         \IfBooleanTF {#2}
2109         { \xeCJK_set_family:xnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
2110         { \xeCJK_set_family:xnn \CJKttdefault {#3} {#4} }
2111     }
2112 }
(End definition for \setCJKmonofont This function is documented on page 5.)

```

## 5.14 xeCJK 其它选项

**CJKnumber** 是否启用 CJKnumber 宏包和首行是否缩进,其中 CJKnumber 选项仅在 xeCJK 宏包调用的时候有效。并将 xeCJK 中未知的选项传递给 fontspec 宏包。

**indentfirst**

```

2113 \bool_new:N \g__xeCJK_indent_bool
2114 \keys_define:nn { xeCJK / options }
2115 {
2116     CJKnumber .bool_set:N = \g__xeCJK_number_bool ,
2117     indentfirst .bool_set:N = \g__xeCJK_indent_bool ,
2118     normalindentfirst .meta:n = { indentfirst = false } ,
2119     quiet .code:n =
2120     {
2121         \msg_redirect_module:nnn { xeCJK } { warning } { info }
2122         \msg_redirect_module:nnn { xeCJK } { info } { none }
2123         \xeCJK_if_package_loaded:nF { fontspec }
2124         { \PassOptionsToPackage { quiet } { fontspec } }
2125     },
2126     silent .code:n =
2127     {
2128         \msg_redirect_module:nnn { xeCJK } { warning } { none }
2129         \msg_redirect_module:nnn { xeCJK } { info } { none }
2130         \xeCJK_if_package_loaded:nF { fontspec }
2131         { \PassOptionsToPackage { silent } { fontspec } }
2132     },
2133     unknown .code:n =
2134     {
2135         \xeCJK_if_package_loaded:nTF { fontspec }
2136         { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2137         { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2138     },
2139 }

```

```

2140 \__xeCJK_msg_new:nn { key-unknown }
2141 {
2142   Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~'#1'.\\
2143   The~key~'#1'~is~being~ignored.
2144 }

```

(End definition for CJKnumber and indentfirst These functions are documented on page 3.)

## 5.15 xeCJK 初始化设置

xeCJK 宏包的初始化设置。

```

2145 \keys_set:nn { xeCJK / options }
2146 {
2147   CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
2148   CJKecglue      = \c_space_token ,
2149   xCJKecglue     = false ,
2150   CheckSingle    = false ,
2151   CJKspace       = false ,
2152   CJKmath        = false ,
2153   xeCJKactive    = true ,
2154   indentfirst    = true ,
2155   EmboldenFactor = 4 ,
2156   SlantFactor    = .167 ,
2157   PunctStyle     = quanjiao ,
2158   KaiMingPunct   = { ^^^^3002 ^^^^ff0e ^^^^ff1f ^^^^ff01 } ,
2159   LongPunct      = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^2025 ^^^^2026 } ,
2160   MiddlePunct    = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
2161   AllowBreakBetweenPuncts = false ,
2162 }

```

```

\CJKsymbol
\CJKpunctsymbol 2163 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2164 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}

```

(End definition for \CJKsymbol and \CJKpunctsymbol)

执行宏包选项,并载入 fontspec 宏包。

```

2165 \ProcessKeysOptions { xeCJK / options }
2166 \RequirePackage { fontspec } [ 2011/09/13 ]

```

\c\_\_xeCJK\_encoding\_tl

```

2167 \tl_const:Nx \c__xeCJK_encoding_tl { \g_fontspec_encoding_tl }
2168 (End definition for \c__xeCJK_encoding_tl)
2169 \bool_if:NT \g__xeCJK_indent_bool
2170 {
2171   \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
2172   \afterindenttrue
2173 }

```

**\xeCJKsetup** 在导言区或文档中设置 xeCJK 的接口。

```

2173 \NewDocumentCommand \xeCJKsetup { m }
2174 {
2175   \keys_set:nn { xeCJK / options } {#1}
2176   \tex_ignorespaces:D
2177 }

```

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

\xeCJKsetemboldenfactor

\xeCJKsetslantfactor

```
2178 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
2179 { \xeCJKsetup { EmboldenFactor = {#1} } }
2180 \NewDocumentCommand \xeCJKsetslantfactor { m }
2181 { \xeCJKsetup { SlantFactor = {#1} } }
```

(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor)

\punctstyle

\xeCJKplainchr

```
2182 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2183 \NewDocumentCommand \xeCJKplainchr { } { \punctstyle { plain } }
```

(End definition for \punctstyle and \xeCJKplainchr)

\CJKsetecglue

```
2184 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2185 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
```

(End definition for \CJKsetecglue)

## 5.16 兼容性修补

\\_\_xeCJK\_fix\_itcorr: 修复倾斜校正。

```
2186 \cs_new_protected_nopar:Npn \__xeCJK_fix_itcorr:
2187 {
2188   \scan_stop:
2189   \xeCJK_if_last_node:nTF { default }
2190   {
2191     \tex_unkern:D \tex_unkern:D
2192     \__xeCJK_itcorr_aux
2193     { \xeCJK_make_node:n { default_itcorr } }
2194   }
2195   { \__xeCJK_itcorr_aux }
2196 }
2197 \cs_new_eq:NN \__xeCJK_itcorr_aux \/
2198 \cs_set_protected_nopar:Npn \/
2199 {
2200   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
2201   { \__xeCJK_fix_itcorr: } { \__xeCJK_itcorr_aux }
2202 }
2203 <@@=>
2204 \cs_set_eq:NN @@italiccorr \/
2205 <@@=xeCJK>
```

(End definition for \\_\_xeCJK\_fix\_itcorr:)

\\_\_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。

\c\_\_xeCJK\_others\_class\_int

```
2206 \__xeCJK_AfterEndPreamble:n
2207 {
2208   \int_compare:nNnF
2209   { \c_three + ( \seq_count:N \g__xeCJK_class_seq ) } = \xe@alloc@intercharclass
2210   {
```

```

2211 \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
2212 {
2213   \seq_if_in:NnF \g__xeCJK_class_seq {#1}
2214   { \__xeCJK_set_others_toks:n {#1} }
2215 }
2216 }
2217 }
2218 \cs_new_protected_nopar:Npn \__xeCJK_set_others_toks:n #1
2219 {
2220   \int_set:Nn \c__xeCJK_Others_class_int {#1}
2221   \clist_map_inline:Nn \g__xeCJK_CJK_class_clist
2222   {
2223     \__xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2224     \__xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2225     \exp_args:Nnnx \__xeCJK_app_inter_class_toks:nnn {##1} { Others }
2226     { \__xeCJK_get_inter_class_toks:nn { Default } { Others } }
2227     \exp_args:Nnnx \__xeCJK_pre_inter_class_toks:nnn { Others } {##1}
2228     { \__xeCJK_get_inter_class_toks:nn { Others } { Default } }
2229     \xeCJK_if_blank_x:nT
2230     { \__xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2231     {
2232       \__xeCJK_copy_inter_class_toks:nnnn
2233       { Others } { Boundary } { Default } { Boundary }
2234     }
2235     \xeCJK_if_blank_x:nT
2236     { \__xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2237     {
2238       \__xeCJK_copy_inter_class_toks:nnnn
2239       { Boundary } { Others } { Boundary } { Default }
2240     }
2241   }
2242 }
2243 \int_new:N \c__xeCJK_Others_class_int
(End definition for \__xeCJK_set_others_toks:n and \c__xeCJK_Others_class_int)

```

\\_\_xeCJK\_patch:Nnn 给已有宏内容前后附加补丁。

```

2244 \cs_new_protected_nopar:Npn \__xeCJK_patch:Nnn #1#2#3
2245 { \tl_put_left:Nn #1 {#2} \tl_put_right:Nn #1 {#3} }
(End definition for \__xeCJK_patch:Nnn)

```

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。

```

2246 \__xeCJK_AfterPreamble:n
2247 {
2248   \tl_map_inline:nn
2249   {
2250     \textellipsis \textemdash \textperiodcentered \textcentereddot
2251     \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2252   }
2253   {
2254     \__xeCJK_patch:Nnn #1
2255     { \__xeCJK_group_begin: \makeCJKinactive }
2256     { \__xeCJK_group_end: }

```

```

2257     }
2258     \__xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2259     \cs_set_eq:NN \__xeCJK_aux_r \r
2260     \cs_set_protected_nopar:Npn \r #1 { { \makexeCJKinactive \__xeCJK_aux_r {#1} } }
2261     \xeCJK_if_package_loaded:nT { pifont }
2262     {
2263         \RenewDocumentCommand \Pifont { m }
2264         { \makexeCJKinactive \usefont {U} {#1} {m} {n} }
2265     }
2266 }
2267 \cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
2268 \cs_new_eq:NN \__xeCJK_group_end: \group_end:
    简单处理与 hyperref 宏包的兼容问题。
2269 \__xeCJK_AfterEndPreamble:n
2270 {
2271     \bool_if:nT
2272     {
2273         \xeCJK_if_package_loaded_p:n { hyperref } &&
2274         \tl_if_exist_p:N \pdfstringdefPreHook
2275     }
2276     {
2277         \tl_gput_right:Nn \pdfstringdefPreHook
2278         {
2279             \cs_set_eq:NN \makexeCJKinactive \prg_do_nothing:
2280             \cs_set_eq:NN \__xeCJK_group_begin: \prg_do_nothing:
2281             \cs_set_eq:NN \__xeCJK_group_end: \prg_do_nothing:
2282             \cs_set_eq:NN \CJKfamily \use_none:n
2283         }
2284     }
2285 }
    禁止在 xeCJK 宏包后再载入 CJK 宏包。
2286 \tl_set:cn { ver@CJK.sty } { 2100/01/01 }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

2287 \cs_if_free:NT \CJK@ifundefined
2288 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2289 \NewDocumentCommand \xeCJKcaption { o m }
2290 {
2291     \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
2292     \cs_set_nopar:Npx \__xeCJK_reset_at_catcode:
2293     {
2294         \exp_not:n { \char_set_catcode:nn { \@ } }
2295         { \char_value_catcode:n { \@ } }
2296     }
2297     \char_set_catcode_letter:N \@
2298     \file_input:n { #2.cpx }
2299     \__xeCJK_reset_at_catcode:
2300     \XeTeXdefaultencoding "UTF-8"
2301 }
    (End definition for \xeCJKcaption)

```

```

\XeCJK_ULprepunctchar:n
\XeCJK_ULpostpunctchar:n 2302 \cs_new_protected_nopar:Npn \XeCJK_ULprepunctchar:n #1
\XeCJK_ULroutines: 2303 {
2304   { \makeXeCJKinactive \CJKpunctsymbol {#1} \XeCJK_no_break: }
2305   \tex_ignorespaces:D
2306 }
2307 \cs_new_protected_nopar:Npn \XeCJK_ULpostpunctchar:n #1
2308 {
2309   { \makeXeCJKinactive \CJKpunctsymbol {#1} }
2310   \XeCJK_ignorespaces:
2311 }
2312 \cs_new_protected_nopar:Npn \XeCJK_ULroutines:
2313 {
2314   \__XeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKecglue \CJKsymbol }
2315   \__XeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2316   \__XeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKecglue \CJKsymbol }
2317   \__XeCJK_inter_class_toks:nnn { Boundary } { CJK } { \XeCJK_Boundary_and_CJK: }
2318   \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2319   {
2320     \__XeCJK_inter_class_toks:nnn {##1} { FullLeft } { \XeCJK_ULprepunctchar:n }
2321     \__XeCJK_inter_class_toks:nnn {##1} { FullRight } { \XeCJK_ULpostpunctchar:n }
2322   }
2323   \__XeCJK_UL_subclass_patch_tl
2324 }
(End definition for \XeCJK_ULprepunctchar:n, \XeCJK_ULpostpunctchar:n, and \XeCJK_ULroutines:)
对 ulem 宏包打补丁,以支持 CJKfntef 宏包。
2325 \__XeCJK_AfterPreamble:n
2326 {
2327   \cs_if_exist:NT \UL@hook
2328   {
2329     \addto@hook \UL@hook
2330     {
2331       \cs_set_eq:NN \__XeCJK_UL_CJKsymbol \CJKsymbol
2332       \cs_set_eq:NN \__XeCJK_UL_CJKpunctsymbol \CJKpunctsymbol
2333       \cs_set_protected_nopar:Npn \CJKsymbol #1
2334       {
2335         { \XeCJK_select_font: \__XeCJK_UL_CJKsymbol {#1} }
2336         { \XeCJK_make_node:n { CJK } } \XeCJK_ignorespaces:
2337       }
2338       \cs_set_protected_nopar:Npn \CJKpunctsymbol #1
2339       { { \XeCJK_select_font: \__XeCJK_UL_CJKpunctsymbol {#1} } }
2340       \XeCJK_ULroutines:
2341     }
2342   }
2343   \cs_if_exist:NT \XeTeX@CJKfntef@hook
2344   {
2345     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2346     { \XeCJK_select_font: \makeXeCJKinactive }
2347   }
2348 }
使用 CJKnumb 宏包。

```

```

2349 \bool_if:NT \g__xeCJK_number_bool
2350 {
2351   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2352   \cs_set_protected_nopar:Npn \CJKaddEncHook #1#2
2353     { \cs_set_nopar:cpn { __xeCJK_enc_#1 } {#2} }
2354   \cs_set_protected_nopar:Npn \Unicode #1#2
2355     { \tex_char:D \int_eval:n { (#1) * \c_two_hundred_fifty_six + (#2) } }
2356   \RequirePackage { CJKnumb }
2357   \cs_if_exist_use:cT { __xeCJK_enc_\CJK@UnicodeEnc } { }
2358   \tl_set:Nx \CJK@tenthousand { ^^^^4e07 }
2359   \tl_set:Nx \CJK@hundredmillion { ^^^^4ebf }
2360 }
2361 </package>

```

## 6 例子

### 6.1 xeCJK-example-autofake.tex

```

1 <*ex-autofake>
2 \documentclass{article}
3 \usepackage[AutoFakeBold,AutoFakeSlant]{xeCJK}
4 \setCJKmainfont[BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsun.ttc}
5 \setCJKsansfont[AutoFakeSlant=false,
6   BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsun.ttc}
7 \setCJKmonofont[ItalicFont=simkai.ttf]{simsun.ttc}
8 \begin{document}
9 \centering
10 \begin{tabular}{lllll}
11 \hline
12 {\bf rm} & md & up & \verb|\rmfamily\mdseries\upshape| & &
13           & & & {\rmfamily\mdseries\upshape English 中文字体} & \\\
14           & md & it & \verb|\rmfamily\mdseries\itshape| & &
15           & & & {\rmfamily\mdseries\itshape English 中文字体} & \\\
16           & md & sl & \verb|\rmfamily\mdseries\slshape| & &
17           & & & {\rmfamily\mdseries\slshape English 中文字体} & \\\
18           & bf & up & \verb|\rmfamily\bfseries\upshape| & &
19           & & & {\rmfamily\bfseries\upshape English 中文字体} & \\\
20           & bf & it & \verb|\rmfamily\bfseries\itshape| & &
21           & & & {\rmfamily\bfseries\itshape English 中文字体} & \\\
22           & bf & sl & \verb|\rmfamily\bfseries\slshape| & &
23           & & & {\rmfamily\bfseries\slshape English 中文字体} & \\\
24 {\bf sf} & md & up & \verb|\sffamily\mdseries\upshape| & &
25           & & & {\sffamily\mdseries\upshape English 中文字体} & \\\
26           & md & it & \verb|\sffamily\mdseries\itshape| & &
27           & & & {\sffamily\mdseries\itshape English 中文字体} & \\\
28           & md & sl & \verb|\sffamily\mdseries\slshape| & &
29           & & & {\sffamily\mdseries\slshape English 中文字体} & \\\
30           & bf & up & \verb|\sffamily\bfseries\upshape| & &
31           & & & {\sffamily\bfseries\upshape English 中文字体} & \\\
32           & bf & it & \verb|\sffamily\bfseries\itshape| & &
33           & & & {\sffamily\bfseries\itshape English 中文字体} & \\\

```

```

34      & bf & sl & \verb|\sffamily\bfseries\slshape| &
35      {\sffamily\bfseries\slshape English 中文字体} \\ \hline
36 {\bf tt} & md & up & \verb|\ttfamily\mdseries\upshape| &
37 {\ttfamily\mdseries\upshape English 中文字体} \\
38      & md & it & \verb|\ttfamily\mdseries\itshape| &
39 {\ttfamily\mdseries\itshape English 中文字体} \\
40      & md & sl & \verb|\ttfamily\mdseries\slshape| &
41 {\ttfamily\mdseries\slshape English 中文字体} \\ \cline{2-5}
42      & bf & up & \verb|\ttfamily\bfseries\upshape| &
43 {\ttfamily\bfseries\upshape English 中文字体} \\
44      & bf & it & \verb|\ttfamily\bfseries\itshape| &
45 {\ttfamily\bfseries\itshape English 中文字体} \\
46      & bf & sl & \verb|\ttfamily\bfseries\slshape| &
47 {\ttfamily\bfseries\slshape English 中文字体} \\ \hline
48 \end{tabular}
49 \end{document}
50 </ex-autofake>

```

## 6.2 xeCJK-example-fallback.tex

```

51 <*ex-fallback>
52 \documentclass{article}
53 \usepackage[AutoFallBack]{xeCJK}
54 \usepackage{CJKfntef}
55 \usepackage{array}
56 \setCJKmainfont[AutoFakeBold,AutoFakeSlant]{KaiTi_GB2312}
57 \setCJKfallbackfamilyfont{\CJKrmdefault}[BoldFont=SimHei]
58 { [SlantedFont=FangSong]{SimSun} ,
59   [BoldFont=*]           {SimSun-ExtB} }
60 \begin{document}
61 漢字源流考
62
63 \textbf{漢字源流考}
64
65 \textsl{漢字源流考}
66
67 \CJKunderwave{漢字源流考}
68 \begin{table}[ht]
69 \caption{生僻字测试}
70 \medskip\centering
71 \begin{tabular}{*4{<{\ttfamily U+}}l}}
72 𠂇 & 3400 & 𠂈 & 3401 & 𠂉 & 3402 & 𠂊 & 3403 & \\
73 𠂋 & 3404 & 𠂌 & 3405 & 𠂍 & 3406 & 𠂎 & 3407 & \\
74 𠂏 & 3408 & 𠂐 & 3409 & 𠂑 & 340A & 𠂒 & 340B & \\
75 𠂓 & 340C & 𠂔 & 340D & 𠂕 & 340E & 𠂖 & 341F & \\
76 𠂗 & 3410 & 𠂘 & 3411 & 𠂙 & 3412 & 𠂚 & 3413 & \\
77 𠂛 & 3414 & 𠂜 & 3415 & 𠂝 & 3416 & 𠂞 & 3417 & \\
78 𠂟 & 3418 & 𠂠 & 3419 & 𠂡 & 341A & 𠂢 & 341B & \\
79 𠂣 & 341C & 𠂤 & 341D & 𠂥 & 341E & 𠂦 & 341F & \\[1ex]
80 𠂧 & 20000 & 𠂨 & 20001 & 𠂩 & 20002 & 𠂪 & 20003 & \\
81 𠂫 & 20004 & 𠂬 & 20005 & 𠂭 & 20006 & 𠂮 & 20007 & \\
82 𠂯 & 20008 & 𠂰 & 20009 & 𠂱 & 2000A & 𠂲 & 2000B & \\
83 𠂳 & 2000C & 𠂴 & 2000D & 𠂵 & 2000E & 𠂶 & 2000F & \\

```

```

84 虫 & 20010 & 丙 & 20011 & 亏 & 20012 & 引 & 20013 \\
85 亥 & 20014 & 互 & 20015 & 共 & 20016 & 井 & 20017 \\
86 兂 & 20018 & 丽 & 20019 & 丕 & 2001A & 丙 & 2001B \\
87 巫 & 2001C & 王 & 2001D & 𠂇 & 2001E & 束 & 2001F \\
88 \end{tabular}
89 \end{table}
90 \end{document}
91 </ex-fallback>

```

### 6.3 xeCJK-example-subCJKblock.tex

```

92 <*ex-block>

93 \documentclass{article}
94 \usepackage{xeCJK}
95 \usepackage{array}
96 \xeCJKDeclareSubCJKBlock{Ext-A} { "3400 -> "4DBF }
97 \xeCJKDeclareSubCJKBlock{Ext-B} { "20000 -> "2A6DF }
98 \xeCJKDeclareSubCJKBlock{Kana} { "3040 -> "309F, "30A0 -> "30FF, "31F0 -> "31FF, }
99 \xeCJKDeclareSubCJKBlock{Hangul}{ "1100 -> "11FF, "3130 -> "318F, "A960 -> "A97F, "AC00 -
> "D7AF }
100 \setCJKmainfont[Ext-A=SimHei,Ext-B=SimSun-ExtB]{SimSun}
101 \setCJKmainfont[Kana]{Meiryō}
102 \setCJKmainfont[Hangul]{Malgun Gothic}
103 \parindent=2em
104 \begin{document}
105 \long\def\showtext{%
106 中日韩越统一表意文字（英語：CJKV Unified Ideographs），旧称中日韩统一表意文字（英
語：CJK Unified Ideographs），也称统一汉字（英語：Unihan），目的是要把分别来自中
文、日文、韩文、越文、壮文中，对于相同起源、本义相同、形状一样或稍异的表意文字
（主要为汉字，但也有仿汉字如：方块壮字、日文汉字（かんじ / kanji）、韩文汉字（한
자 / hanja）、越南的喃字（Ch Nôm）与越文汉字（Ch Nho，在越南也称作儒字），应赋予
其在 ISO 10646 及统一码标准中有相同编码。此计划原本只包含中文、日文及韩文中所使用的
汉字，是以旧称中日韩统一表意文字（CJK）。后来，此计划加入了越文的喃字，所以合称中
日韩越统一表意文字（CJKV）。
107
108 CJK 統合漢字（シージェーケーとうごうかんじ、CJK Unified Ideographs）は、ISO/IEC 10646 (Uni-
versal Multiple-Octet Coded Character Set, 略称 UCS) および Unicode にて採用されてい
る符号化用漢字集合およびその符号表である。CJK 統合漢字の名称は、中国語 (Chinese)、
日本語 (Japanese)、韓国語 (Korean) で使われている漢字をひとまとめにしたことから
きている。CJK 統合漢字の初版である Unified Repertoire and Ordering (URO) 第二版は
1992 年に制定されたが、1994 年にベトナムで使われていた漢字も含めることにしたため、
CJKV と呼ばれる事もある。CJKV は、中国語・日本語・韓国語・ベトナム語 (英語: Chi-
nese-Japanese-Korean-Vietnamese) の略。特に、その 4 言語で共通して使われる、または
使われていた文字体系である漢字 (チュノムを含む) のこと。ソフトウェアの国際化、中
でも文字コードに関する分野で用いられる。
109
110 \CJKspace
111 한중일월 통합 한자 (또는 한중일 통합 한자) 는 유니코드에 담겨 있는 한자들의 집합으
로, 한국, 중국, 일본에서 쓰이는 한자를 묶은 것이기 때문에 머리 글자를 따서 한중일
(CJK) 통합 한자라고 불렀는데, 최근에는 베트남에서 쓰이는 한자도 추가되었기에 한중일
월 (CJKV) 통합 한자로 부르게 되었다.
112

```

113 처음에 유니코드에는 65,536( $2^{16}$ )자만 들어갈 수 있었기 때문에, 가장 많은 문자가 배당되는 한자를 위해서 한국, 중국, 일본에서 사용하는 한자 중에 모양이 유사하며 그 뜻이 같은 글자를 같은 코드로 통합했다. 따라서 문자 코드만으로 그 한자가 사용되는 언어를 알아 낼 수 없는데, 다만 중국의 간체자나 번체자, 일본의 구자체나 신자체 등 분명하게 모양이 다른 글자는 별도의 부호를 할당하고 있다. 이런 문자 할당 정책에 반발하여 TRON 과 같은 인코딩이 만들어지기도 했으나, 실제로 통합된 한자의 차이가 별로 크지 않기 때문에 문제가 되지 않는다는 의견도 있다.

```

114 \CJKnospace}
115 \showtext
116
117 \bigskip
118 \xeCJKCancelSubCJKBlock{Kana,Hangul}
119 \showtext
120
121 \bigskip
122 \xeCJKRestoreSubCJKBlock{Hangul}
123 \showtext
124
125 \begin{table}[ht]
126 \caption{生僻字测试}
127 \medskip\centering
128 \begin{tabular}{*4{<{\ttfamily U+}}|}
129 北 & 3400 & 西 & 3401 & 𠂇 & 3402 & 个 & 3403 & \\
130 牛 & 3404 & 乂 & 3405 & 𠂇 & 3406 & 𠂇 & 3407 & \\
131 𠂇 & 3408 & 𠂇 & 3409 & 𠂇 & 340A & 𠂇 & 340B & \\
132 𠂇 & 340C & 𠂇 & 340D & 𠂇 & 340E & 𠂇 & 341F & \\
133 𠂇 & 3410 & 𠂇 & 3411 & 𠂇 & 3412 & 𠂇 & 3413 & \\
134 𠂇 & 3414 & 𠂇 & 3415 & 𠂇 & 3416 & 𠂇 & 3417 & \\
135 𠂇 & 3418 & 𠂇 & 3419 & 𠂇 & 341A & 𠂇 & 341B & \\
136 𠂇 & 341C & 𠂇 & 341D & 𠂇 & 341E & 𠂇 & 341F & \\[1ex]
137 𠂇 & 20000 & 𠂇 & 20001 & 𠂇 & 20002 & 𠂇 & 20003 & \\
138 𠂇 & 20004 & 𠂇 & 20005 & 𠂇 & 20006 & 𠂇 & 20007 & \\
139 𠂇 & 20008 & 𠂇 & 20009 & 𠂇 & 2000A & 𠂇 & 2000B & \\
140 𠂇 & 2000C & 𠂇 & 2000D & 𠂇 & 2000E & 𠂇 & 2000F & \\
141 𠂇 & 20010 & 𠂇 & 20011 & 𠂇 & 20012 & 𠂇 & 20013 & \\
142 𠂇 & 20014 & 𠂇 & 20015 & 𠂇 & 20016 & 𠂇 & 20017 & \\
143 𠂇 & 20018 & 𠂇 & 20019 & 𠂇 & 2001A & 𠂇 & 2001B & \\
144 𠂇 & 2001C & 𠂇 & 2001D & 𠂇 & 2001E & 𠂇 & 2001F & \\
145 \end{tabular}
146 \end{table}
147 \end{document}
148 \end{ex-block}

```

## 6.4 xeCJK-example-CJKecglue.tex

```

149 \*ex-ecglue)
150 \documentclass{minimal}
151 \usepackage{xeCJK}
152 \setCJKmainfont[BoldFont=SimHei]{SimSun}
153 \long\def\showtext{%
154 这是 English 中文 {\itshape Chinese} 中文 \TeX\
155 间隔 \textit{Italic} 中文\textbf{字体} a 数学 $b$ 数学 $c$ $d$\par
156 这是English中文{\itshape Chinese}中文\TeX\

```

```

157 间隔\textit{Italic}中文\textbf{字体}a数学$b$数学$c$ $d$\par
158 This is an example. 这是一个例子}
159 \begin{document}
160 \showtext
161
162 \hrulefill\bigskip
163
164 \xeCJKsetup{xCJKecglue=\quad}
165 \showtext
166 \end{document}
167 \

```

## 6.5 xeCJK-example-checksingle.tex

```

168 \*ex-single)
169 \documentclass{minimal}
170 \usepackage{xeCJK}
171 \setCJKmainfont{SimSun}
172 \catcode`\。=\active
173 \def。{. }
174 \def\foo{一}
175 \long\def\showtext{一二三四五六七八九十一二三三四五六七八九十一二三三四五六七八九十。
176 $$x^2+y^2$$
177 一二三四五六七八九十一二三三四五六七八九十一二三三四五六七八九十\foo
178 \begin{equation}
179 x^2+y^2
180 \end{equation}
181 一二三四五六七八九十一二三三四五六七八九十一二三三四五六七八九十。}
182 \begin{document}
183 \hsize=30em
184 \parindent=0pt
185 \showtext
186
187 \hrulefill\bigskip
188
189 \xeCJKsetup{CheckSingle}
190 \showtext
191 \end{document}
192 \

```

## 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	
\/	61, 63, 2197, 2198, 2204
\@@italiccorr	2204
\@afterindentfalse	2170
\@afterindenttrue	2170, 2171
\@depth	486
\@height	487
\@ifpackagelater	19
\@minus	446, 495
\@onlypreamble	836,

887, 1916, 1979, 1980, 1981, 1982, 1983  
 \@pkgextension ..... 128  
 \@plus ..... 445, 494, 2147  
 \@width ..... 485  
 \[ ..... 770  
 \] ..... 770  
 \\_xeCJK\_AfterEndPreamble:n .....  
 ..... 45, 52, 2206, 2269  
 \\_xeCJK\_AfterPreamble:n .....  
 ..... 31, 45, 50, 2102, 2246, 2325  
 \\_xeCJK\_AtEndPreamble:n .... 45, 48, 2025  
 \\_xeCJK\_UL\_CJKpunctsymbol ... 2332, 2339  
 \\_xeCJK\_UL\_CJKsymbol ..... 2331, 2335  
 \\_xeCJK\_UL\_subclass\_patch:nn 883, 927, 927  
 \\_xeCJK\_UL\_subclass\_patch\_tl .....  
 ..... 819, 819, 929, 2323  
 \\_xeCJK\_add\_fake\_bold:n .....  
 ..... 1619, 1619, 1667, 1684  
 \\_xeCJK\_add\_fake\_slant:n 1619, 1624, 1676  
 \\_xeCJK\_add\_font:nV ..... 1647  
 \\_xeCJK\_add\_font:nn .... 1602, 1602, 1612  
 \\_xeCJK\_add\_font:nv ..... 1635  
 \\_xeCJK\_add\_font:nx ..... 1692  
 \\_xeCJK\_add\_font\_if\_new:nn .....  
 ..... 1602, 1607, 1613, 1686  
 \\_xeCJK\_add\_font\_if\_new:nv ..... 1639  
 \\_xeCJK\_add\_font\_if\_new:nx ..... 1694  
 \\_xeCJK\_add\_special\_punct:nn .....  
 ..... 985, 988, 990, 1000, 1013  
 \\_xeCJK\_add\_sub\_block: .. 1588, 1734, 1734  
 \\_xeCJK\_app\_inter\_class\_toks:nnc 321, 352  
 \\_xeCJK\_app\_inter\_class\_toks:nnn ...  
 ..... 321, 321, 326, 350, 376, 902, 2225  
 \\_xeCJK\_aux\_r ..... 2259, 2260  
 \\_xeCJK\_calc\_kern:NNN ... 1167, 1176, 1181  
 \\_xeCJK\_check\_family:V ..... 1569, 1742  
 \\_xeCJK\_check\_family:n .. 1592, 1592, 1600  
 \\_xeCJK\_check\_for\_glue: .... 390, 392, 398  
 \\_xeCJK\_check\_num\_range:nnNN .....  
 ..... 206, 228, 228, 1990  
 \\_xeCJK\_checksingle\_save\_CJKsymbol:N  
 ..... 704,  
 713, 729, 735, 748, 757, 762, 781, 789, 796  
 \\_xeCJK\_class\_num:n .....  
 ..... 178, 178, 192, 306, 310, 313  
 \\_xeCJK\_clear\_CJK\_toks: .....  
 ..... 332, 332, 346, 394, 461  
 \\_xeCJK\_clear\_inter\_class\_toks:nn ..  
 ..... 314, 314, 335, 345  
 \\_xeCJK\_copy\_family:nn .. 1777, 1777, 1787  
 \\_xeCJK\_copy\_family:xx ..... 1812, 1813  
 \\_xeCJK\_copy\_inter\_class\_toks:nnnn .  
 ..... 327, 327, 892,  
 893, 906, 911, 912, 2223, 2224, 2232, 2238  
 \\_xeCJK\_declare\_char\_class:nn .....  
 181, 184, 242, 244, 246, 253, 268, 862, 885  
 \\_xeCJK\_erase\_CJKsymbol: 821, 821, 901, 938  
 \\_xeCJK\_error:n ..... 129, 130  
 \\_xeCJK\_error:nx .. 129, 131, 866, 970, 2136  
 \\_xeCJK\_fallback\_save\_CJKsymbol:N ..  
 ..... 1326, 1335, 1341, 1351  
 \\_xeCJK\_family\_unknown\_warning:x ...  
 ..... 1847, 1852, 1856, 1858  
 \\_xeCJK\_fix\_itscorr: .... 2186, 2186, 2201  
 \\_xeCJK\_flexible\_cjkgglue: ... 2076, 2095  
 \\_xeCJK\_flexible\_ecglue: .... 2075, 2096  
 \\_xeCJK\_get\_inter\_class\_toks:nn 312,  
 312, 319, 324, 330, 2226, 2228, 2230, 2236  
 \\_xeCJK\_get\_map\_font:n .....  
 ..... 1692, 1694, 1697, 1697, 1701  
 \\_xeCJK\_get\_sub\_features:nn .....  
 ..... 1376, 1444, 1452, 1473  
 \\_xeCJK\_get\_sub\_features\_aux:n ....  
 ..... 1459, 1473, 1481  
 \\_xeCJK\_gobble\_brace:N .....  
 ..... 1456, 1473, 1473, 1478  
 \\_xeCJK\_group\_begin: .... 2255, 2267, 2280  
 \\_xeCJK\_group\_end: .... 2256, 2268, 2281  
 \\_xeCJK\_if\_font\_select:n ..... 1654  
 \\_xeCJK\_if\_font\_select:nF 1638, 1646, 1693  
 \\_xeCJK\_if\_font\_select:nTF ... 1633, 1654  
 \\_xeCJK\_if\_font\_select\_p:n .....  
 ..... 1654, 1673, 1674, 1682  
 \\_xeCJK\_if\_it\_or\_sl:n ..... 1659  
 \\_xeCJK\_if\_it\_or\_sl:nT ..... 1678  
 \\_xeCJK\_if\_it\_or\_sl:nTF .... 1636, 1654  
 \\_xeCJK\_if\_it\_or\_sl\_p:n . 1654, 1672, 1681  
 \\_xeCJK\_if\_map\_font\_select:n ..... 1699  
 \\_xeCJK\_if\_map\_font\_select:nT .... 1690  
 \\_xeCJK\_if\_map\_font\_select:nTF ... 1699  
 \\_xeCJK\_if\_map\_font\_select\_p:n ... 1699  
 \\_xeCJK\_if\_outer\_macro\_aux:w .... 69, 72  
 \\_xeCJK\_ignore\_spaces: .... 646, 648, 648  
 \\_xeCJK\_info:nx ..... 129, 135  
 \\_xeCJK\_info:nxx ..... 129, 136, 1803  
 \\_xeCJK\_int\_until\_do\_aux:wn .....  
 ..... 218, 220, 223, 225  
 \\_xeCJK\_inter\_class\_toks:nnc ... 309, 480  
 \\_xeCJK\_inter\_class\_toks:nnn .....  
 309, 309, 311, 315, 341, 348, 356, 380,  
 389, 448, 454, 458, 465, 466, 934, 935,  
 936, 2314, 2315, 2316, 2317, 2320, 2321

\_xeCJK_inter_class_toks:nxx	.....	309, 318, 323, 329
\_xeCJK_itcorr_aux	.	2192, 2195, 2197, 2201
\_xeCJK_map_features_id:n		1487, 1605, 1610
\_xeCJK_map_it_sl:n	....	1678, 1688, 1688
\_xeCJK_math_type:n	.....	1991, 1999
\_xeCJK_msg_new:nn	.	129, 129, 869, 972, 1355, 1601, 1819, 1867, 1956, 2032, 2140
\_xeCJK_new_class:n	.....	152, 152, 159, 160, 161, 880
\_xeCJK_new_sub_key:n	.....	882, 1420, 1421, 1451, 1486
\_xeCJK_parse_features:	.	1567, 1629, 1629
\_xeCJK_parse_features:n		1629, 1630, 1631
\_xeCJK_pass_features:	..	1568, 1704, 1704
\_xeCJK_patch:Nnn	..	2244, 2244, 2254, 2258
\_xeCJK_peek_execute_branches_catcode:	.....	101, 106
\_xeCJK_peek_false:w	.....	87, 111
\_xeCJK_peek_ignore_spaces_execute_branches:	.....	89, 95, 105
\_xeCJK_peek_ignore_spaces_execute_branches_aux:	.....	843, 852, 856
\_xeCJK_peek_math:	.....	641, 648, 673
\_xeCJK_peek_tmp:w	.....	99
\_xeCJK_peek_true:w	.....	86, 109
\_xeCJK_pre_inter_class_toks:nnn	....	316, 316, 387, 473, 474, 896, 900, 913, 915, 923, 2227
\_xeCJK_punct_glue:n	.....	489, 489, 511, 517, 522, 533
\_xeCJK_punct_if_both_left_right:NN		1147
\_xeCJK_punct_if_both_left_right:NNT		1254
\_xeCJK_punct_if_both_left_right:NNTF	.....	1142, 1195
\_xeCJK_punct_if_both_left_right_p:NN	.....	1142
\_xeCJK_punct_if_long:NNTF		528, 1052, 1142
\_xeCJK_punct_if_long_p:N		1142, 1237, 1282
\_xeCJK_punct_if_middle:NT	....	519, 531
\_xeCJK_punct_if_middle:NNTF	..	1070, 1142
\_xeCJK_punct_if_middle_p:N	.....	1142
\_xeCJK_punct_if_mixedwidth:NT		1112, 1197
\_xeCJK_punct_if_mixedwidth:NNTF		1120, 1142
\_xeCJK_punct_if_mixedwidth_p:N		1142, 1207
\_xeCJK_punct_if_right:N	.....	1142
\_xeCJK_punct_if_right:NNTF	.....	1188
\_xeCJK_punct_if_right:NT	.....	1186
\_xeCJK_punct_if_right:NNTF		1142, 1249, 1251
\_xeCJK_punct_if_right_p:N	.....	1142
\_xeCJK_punct_kern:n	.....	498, 498, 545, 553, 563, 572
\_xeCJK_punct_nobreak:	.....	564, 996, 998
\_xeCJK_punct_rule:n	.....	482, 482, 504, 510, 521, 534, 560, 569
\_xeCJK_reset_at_catcode:	...	2292, 2299
\_xeCJK_restore_CJKsymbol:	.....	821, 826, 903, 941
\_xeCJK_save_CJK_class:n	...	171, 177, 920
\_xeCJK_save_family_info:		1587, 1722, 1722
\_xeCJK_set_char_catcode:n	..	209, 210, 213
\_xeCJK_set_char_class:nnn	..	191, 201, 204
\_xeCJK_set_char_class_aux:Nn	.....	179, 191, 196, 2019
\_xeCJK_set_fake:n	.....	1652, 1664, 1664
\_xeCJK_set_init:	.....	1535, 1535, 1561
\_xeCJK_set_others_toks:n		2206, 2214, 2218
\_xeCJK_set_special_punct:nn	.....	984, 987, 991, 1000, 1006
\_xeCJK_set_sub_class_toks:nn		881, 888, 888
\_xeCJK_space_or_xecglue:	.....	
\_xeCJK_sub_restore_or_cancel:n	...	
\_xeCJK_sub_special_punct:nn	.....	986, 989, 992, 1000, 1019
\_xeCJK_switch_font:nn	.....	897, 903, 914, 916, 924, 940, 1788, 1799
\_xeCJK_warning:n	.....	132, 1952, 2029
\_xeCJK_warning:nx	...	129, 133, 1350, 1863
\_xeCJK_warning:nxx	.....	134, 1597
\_xeCJK_zero_glue:	.....	996, 999
\_	.....	25, 872, 2142
<b>A</b>		
\addCJKfontfeature	.....	1955, 1958
\addCJKfontfeatures	7, 1912, 1917, 1955, 2046	
\addto@hook	.....	2329
KaiMingPunct	.....	5
KaiMingPunct+	.....	5
KaiMingPunct-	.....	5
FallBack	.....	8
\AllowBreakBetweenPuncts	.....	978
\AtBeginDocument	.....	54
\AtEndOfPackage	.....	31, 1971
\AutoFakeBold	.....	1388, 1503
\AutoFakeSlant	.....	1388, 1503
\AutoFallBack	.....	1312
<b>B</b>		
\baselineskip	.....	2147
\begin	.....	784, 785
\bfdefault	.....	2014

\BoldFeatures	1487	\c_four	446, 1304, 2211
\BoldFont	1487	\c_group_begin_token	343, 393, 460
\BoldItalicFeatures	1487	\c_group_end_token	348, 450, 466
\BoldItalicFont	1487	\c_math_toggle_token	654, 777, 779, 780, 781
\BoldSlantedFeatures	1487	\c_one	144, 516, 1303, 2211
\BoldSlantedFont	1487	\c_one_fp	2040
\bool_if:cT	1740, 1770	\c_space_tl	119
\bool_if:NF	701, 840, 863, 1323, 2070	\c_space_token	97, 368, 382, 435, 436, 616, 650, 675, 679, 748, 756, 757, 2148
\bool_if:nF	470	\c_ten	407, 424
\bool_if:NT	710, 849, 1332, 1667, 2030, 2089, 2168, 2349	\c_ten_thousand	44
\bool_if:nT	412, 657, 1236, 1280, 1671, 1679, 2271	\c_three	445, 1304, 1306, 2209
\bool_if:NTF	435, 1669, 2097	\c_two	444, 1303, 2053, 2063
\bool_if:nTF	230, 358, 366, 400, 404, 421, 430, 721, 737, 745, 750, 767, 1149, 1205, 1661	\c_two_hundred_fifty_six	2355
\bool_new:N	628, 698, 837, 1320, 1384, 1385, 2101, 2113	\c_zero	116, 119, 145, 2200
\bool_set_eq:NN	1553, 1554, 2074	\c_zero_dim	486, 487, 1042, 1067, 1074, 1088, 1096, 1129, 1240
\bool_set_false:c	1430, 1431, 1439, 1442, 1550, 1551	\c_zero_skip	999, 2147
\bool_set_false:N	615, 645, 712, 851, 1334, 1394, 1405, 1507, 1516, 2091	\char_set_catcode:nn	2294
\bool_set_true:c	1438, 1443	\char_set_catcode_ignore:n	146
\bool_set_true:N	610, 620, 640, 703, 842, 1325, 1392, 1397, 1403, 1408, 1510, 1519, 2072	\char_set_catcode_letter:N	2297
<b>C</b>			
\__xeCJK_Boundary_class_int	147, 151	\char_set_catcode_other:N	63
\__xeCJK_CJK_class_int	147, 148	\char_set_catcode_other:n	209
\__xeCJK_Default_class_int	147, 147	\char_set_lccode:nn	58, 59, 60, 61, 62
\__xeCJK_encoding_tl	1581, 1762, 2011, 2013, 2167, 2167	\char_value_catcode:n	2295
\__xeCJK_FullLeft_class_int	147, 149	\CheckSingle	690
\__xeCJK_FullRight_class_int	147, 150	\CJK@hundredmillion	2359
\__xeCJK_HalfLeft_class_int	159	\CJK@ifundefined	2287, 2288
\__xeCJK_HalfRight_class_int	159	\CJK@tenthousand	2358
\__xeCJK_math_tl	1985, 1986, 2002, 2003, 2011, 2013, 2020	\CJK@UnicodeEnc	2351, 2357
\__xeCJK_NormalSpace_class_int	159	\CJKaddEncHook	2352
\__xeCJK_Others_class_int	2206, 2220, 2243	\CJKecglue	368, 371, 376, 387, 410, 428, 600, 604, 605, 611, 621, 622, 623, 668, 685, 934, 2075, 2314, 2316
\__xeCJK_pkg_extension_tl	125, 128	xCJKecglue	4
\__xeCJK_ps_CCT_tl	1081, 1118	\CJKfamily	6, 1537, 1573, 1831, 1831, 1894, 1909, 1973, 1974, 1975, 1976, 2282
\__xeCJK_ps_fullwidth_tl	1082, 1192	\CJKfamilydefault	7, 1811, 1812, 1932, 1966, 1969, 1970, 1973, 2005, 2006
\__xeCJK_ps_halfwidth_tl	1079, 1109, 1208	\CJKfixedspacing	6, 2068, 2068, 2102
\__xeCJK_ps_marginkerning_tl	1183	\CJKflexiblespacing	6, 2044, 2068, 2087
\__xeCJK_ps_mixedwidth_tl	1080, 1110	\CJKfontspec	6, 1888, 1898, 1950
\__xeCJK_ps_plain_tl	957, 1040, 1172, 1288	\CJKglue	402, 435, 455, 528, 598, 599, 2076
\c_catcode_other_token	718, 734, 747	\CJKmath	1986
\c_eleven	361, 415	\CJKnospace	633, 638, 643
\c_empty_tl	116, 315, 1448, 1459	\CJKnumber	2113
		\CJKpunctsymbol	505, 538, 556, 575, 2163, 2164, 2304, 2309, 2332, 2338
		\CJKrmdefault	7, 1872, 1881, 1962, 1966, 1966, 1969, 1974
		\CJKsetecglue	2184, 2184, 2185

\CJKsfdefault	1419, 1451, 1591, 1600, 1612, 1613,
..	7, 1873, 1882, 1965, 1966, 1967, 1975
\CJKspace	629, 632, 638, 638
\CJKsymbol	350,
	396, 455, 465, 772, 780, 797, 823, 824,
	828, 829, 934, 935, 1326, 1327, 1335,
	2163, 2163, 2314, 2315, 2316, 2331, 2333
\CJKsymbol_Block	823, 828
\CJKttdefault	7, 1874,
	1883, 1966, 1968, 1976, 2106, 2109, 2110
\clist_clear:c	1544
\clist_clear:N	1455, 1463, 1546, 1547, 1941, 1942
\clist_gclear_new:c	155, 168
\clist_gput_right:cx	186
\clist_gput_right:Nn	919
\clist_gput_right:Nx	173, 805, 1423
\clist_gremove_all:Nn	811
\clist_gremove_duplicates:N	806
\clist_gset:Nn	1914
\clist_if_empty:cF	1708
\clist_if_empty:NF	907
\clist_if_in:NnF	1945
\clist_if_in:NnT	1926
\clist_if_in:NnTF	795
\clist_map_break:	1937
\clist_map_function:nN	177
\clist_map_inline:cn	2017
\clist_map_inline:Nn	167,
	334, 468, 890, 909, 949, 1001, 1159,
	1286, 1548, 1725, 1737, 1924, 1943, 2221
\clist_map_inline:nn	198, 337, 339, 351, 354, 378,
	456, 477, 479, 921, 931, 1644, 2015, 2318
\clist_map_inline:xn	187, 810, 858, 1373
\clist_new:N	41, 42, 43, 169, 820, 946, 1420, 1912
\clist_put_left:cx	1377, 1621, 1626
\clist_put_left:NV	1948
\clist_put_left:Nx	1717
\clist_put_right:Nn	1946
\clist_put_right:Nx	1435, 1710, 1715
\clist_remove_all:Nn	1727, 1728, 1747, 1928
\clist_remove_duplicates:N	1736
\clist_set:cx	1470
\clist_set:Nn	162, 947, 1000
\clist_set:Nx	1483, 1563, 1922, 1935
\cs_end:	74
\cs_generate_variant:Nn	176, 198, 311, 326,
	500, 814, 815, 816, 817, 818, 951, 1027,
	1028, 1029, 1030, 1139, 1158, 1242,
	1419, 1451, 1591, 1600, 1612, 1613,
	1614, 1615, 1616, 1617, 1618, 1787, 1830
\cs_gset_eq:cc	1784
\cs_gset_eq:cN	1795
\cs_gset_eq:NN	823, 824, 828
\cs_gset_protected_nopar:cpx	1570, 1578, 1750, 1759
\cs_if_exist:cF	669, 686
\cs_if_exist:NT	2327, 2343
\cs_if_exist_use:cF	1064, 1791
\cs_if_exist_use:cT	1794, 2357
\cs_if_free:cT	1901
\cs_if_free:cTF	1822
\cs_if_free:NT	28, 2287
\cs_if_free:NTF	2288
\cs_meaning:c	70
\cs_new_eq:NN	1798,
	1955, 1963, 1999, 2185, 2197, 2267, 2268
\cs_new_nopar:Npn	72,
	95, 106, 129, 130, 131, 132, 133, 134,
	135, 136, 178, 312, 1487, 1697, 2163, 2164
\cs_new_protected:Npn	48, 50, 52, 83, 91, 93, 218, 223
\cs_new_protected_nopar:Npn	44, 104, 152, 171, 184,
	204, 228, 309, 314, 316, 321, 327, 332,
	390, 398, 455, 482, 489, 498, 501, 507,
	513, 525, 540, 548, 558, 567, 582, 588,
	648, 673, 716, 732, 765, 775, 793, 821,
	826, 856, 888, 927, 999, 1006, 1013,
	1019, 1031, 1033, 1036, 1167, 1181,
	1243, 1258, 1296, 1338, 1367, 1421,
	1452, 1473, 1481, 1535, 1558, 1592,
	1602, 1607, 1619, 1624, 1629, 1631,
	1664, 1688, 1704, 1722, 1734, 1777,
	1789, 1799, 1852, 1858, 2000, 2048,
	2057, 2186, 2218, 2244, 2302, 2307, 2312
\cs_new_protected_nopar:Npx	1894
\cs_set_eq:NN	85, 99, 209, 210, 611,
	616, 623, 641, 646, 704, 705, 713, 996,
	998, 1326, 1327, 1335, 1537, 1573,
	1977, 2075, 2076, 2170, 2204, 2259,
	2279, 2280, 2281, 2282, 2288, 2331, 2332
\cs_set_nopar:cpn	2353
\cs_set_nopar:Npn	2345
\cs_set_nopar:Npx	2292
\cs_set_protected:Npn	599, 604, 621
\cs_set_protected_nopar:Npn	2198, 2260, 2333, 2338, 2352, 2354
\cs_to_str:N	1892
<b>D</b>	
\DeclareSymbolFont	2011

\defaultCJKfontfeatures .....	\exp_not:V .....	962
..... 7, 1912, 1913, 1915, 1916		
\dim_add:Nc .....	<b>F</b>	
..... 1028, 1187, 1189	\f@series .....	1788
\dim_add:Nn .....	\f@shape .....	1788
..... 1029, 1114, 1121, 1122	\f@size .....	1788, 2053, 2064
\dim_compare:nNnT .....	\FallBack .....	1486
..... 1200, 1216	\fi: .....	75, 102, 112, 120, 121, 225, 580
\dim_compare:nNnTF .....	\file_input:n .....	2298
..... 1074, 1088, 1096	\font_glyph_if_exist:NnTF .....	1340
\dim_compare_p:nNn .....	\fontencoding .....	1581, 1762
..... 417	\fontfamily .....	1582, 1763, 2051, 2060
\dim_const:cn .....	\fontspec_set_family:Nnn .....	1574
..... 585	\fp_gset:Nn .....	2052
\dim_eval:c .....	\fp_new:N .....	1386, 1387, 2039
..... 485, 493, 499, 500	\fp_set:Nn .....	1398, 1409, 1511, 1520
\dim_eval:n .....	\fp_set_eq:NN .....	1555, 1556, 2040
.. 494, 495, 500, 1245, 1266, 1274, 1298	\fp_to_dim:n .....	2064
\dim_gset:cn .....	\fp_use:N .....	1532, 1622, 1627, 2046
..... 585		
\dim_gset:Nn .....	<b>G</b>	
..... 2061	\g__xeCJK_after_end_preamble_hook_tl	
\dim_if_exist:cTF .....	..... 47, 53, 56	
..... 584	\g__xeCJK_after_preamble_hook_tl	46, 51, 54
\dim_max:nn .....	\g__xeCJK_at_end_preamble_hook_tl	
..... 1032, 1220	..... 45, 49, 55	
\dim_min:cc .....	\g__xeCJK_AutoFakeBold_bool	
..... 1028, 1222, 1227	..... 1384, 1392, 1394, 1397, 1553	
\dim_min:nn .....	\g__xeCJK_AutoFakeSlant_bool	
..... 1030, 1034, 1092, 1106	..... 1385, 1403, 1405, 1408, 1554	
\dim_new:N .....	\g__xeCJK_base_class_clist	
..... 36, 37, 38, 627, 2041	..... 162, 162, 167, 468, 890	
\dim_set:Nc .... 1028, 1059, 1060, 1175, 1201	\g__xeCJK_CJK_class_clist	
\dim_set:Nn .....	..... 162, 169, 173, 334, 2221	
..... 1028, 1032,	\g__xeCJK_CJK_class_prop	162, 170, 174, 208
1034, 1062, 1072, 1089, 1091, 1098,	\g__xeCJK_CJK_subclass_clist	
1105, 1193, 1199, 1211, 1212, 1215, 1218	..... 819, 820, 907, 909, 919, 1725, 1924	
\dim_set_eq:NN .....	\g__xeCJK_class_seq	152, 156, 158, 2209, 2213
..... 1126	\g__xeCJK_default_features_clist	
\dim_to_fp:n .....	..... 1566, 1912, 1914	
..... 2053	\g__xeCJK_EmboldenFactor_fp	
\dim_use:c .....	..... 1386, 1398, 1412, 1513, 1555	
..... 579, 590, 591	\g__xeCJK_exspace_dim	
\dim_use:N .....	..... 2039, 2041, 2061, 2081, 2082	
..... 1132, 1134	\g__xeCJK_family_name_prop	1576, 1596,
\dim_zero:N .... 1054, 1055, 1056, 1171, 1185	1719, 1720, 1754, 1757, 1782, 1825, 2009	
\document .....	\g__xeCJK_family_options_prop	
..... 55, 56	..... 1719, 1721, 1731, 1745, 1748, 1782, 1940	
	\g__xeCJK_family_prop	1358,
<b>E</b>		1371, 1594, 1719, 1719, 1730, 1743,
xeCJKactive .....	1744, 1782, 1919, 1923, 1929, 1931, 2028	
..... 3	\g__xeCJK_family_spec_tl	1904, 1906, 1907
\else: .....		
..... 75, 100, 110, 118, 120, 580		
\EmboldenFactor .....		
..... 1388		
\end .....		
..... 787, 788		
\etex_fontcharwd:D .....		
..... 1276		
\etex_lastnodetype:D		361, 407, 415, 424, 516
\etex_protected:D ...		1973, 1974, 1975, 1976
\exp_after:wN		99, 101, 109, 111, 225, 726, 1459
\exp_args:Nc .....		154, 1656
\exp_args:NNc .....		725, 726
\exp_args:Nnnx .....		2225, 2227
\exp_args:No .....		725, 1457, 1475
\exp_args:Nv .....		1909
\exp_args:Nx .....		1363, 2092
\exp_last_unbraced:Nf .....		69
\exp_last_unbraced:NNo .....		1477
\exp_last_unbraced:NNV .....		1795
\exp_not:N .....		108, 669, 686,
..... 1576, 1577, 1581, 1582, 1583, 1754,		
1755, 1757, 1758, 1762, 1763, 1764, 1894		
\exp_not:n .....		
.. 319, 324, 1573, 1574, 1579, 1760, 2294		
\exp_not:o .....		2095, 2096



\l__xeCJK_family_aux_tl	1432, 1562, 1569, 1570, 1577, 1578, 1730, 1731, 1739, 1743, 1745, 1752, 1755, 1892, 1893, 1894
\l__xeCJK_family_fb_tl	1370, 1371, 1375, 1376, 1377, 1378, 1379, 1380
\l__xeCJK_family_name_tl	1825
\l__xeCJK_fixed_spacing_bool	2070, 2072, 2089, 2091, 2101
\l__xeCJK_fixed_xecglue_bool	2074, 2097
\l__xeCJK_flexible_punctstyle_tl	2073, 2094
\l__xeCJK_fontname_bf_tl	1647
\l__xeCJK_fontname_tl	1371, 1372, 1468, 1469, 1564, 1575, 1730
\l__xeCJK_fontoptions_clist	1563, 1727, 1728, 1731
\l__xeCJK_fontspec_family_tl	1574, 1577, 1582
\l__xeCJK_lastpunct_tl	503, 509, 510, 511, 537, 544, 545, 552, 553, 555, 560, 562, 563, 569, 571, 572, 574
\l__xeCJK_math_family_tl	2009, 2012, 2014
\l__xeCJK_monoscale_tl	1527, 1532, 1538, 1716, 1717
\l__xeCJK_pass_features_clist	1546, 1566, 1575, 1710, 1715, 1717
\l__xeCJK_peek_search_token	82, 85, 108
\l__xeCJK_punct_coor_tl	485, 493, 494, 495, 499, 1035, 1035, 1038, 1043, 1044, 1045, 1046, 1047, 1050, 1059, 1061, 1086, 1101, 1131, 1133, 1135, 1144, 1151, 1152, 1153, 1154, 1169, 1178, 1187, 1189, 1201, 1211, 1223, 1224, 1228, 1229, 1248, 1250, 1252, 1253, 1255, 1260, 1262, 1264, 1268, 1269, 1272, 1277, 1291
\l__xeCJK_SlantFactor_fp	1520, 1556, 1627
\l__xeCJK_space_bool	435, 640, 645
\l__xeCJK_space_glue_tl	362, 416, 442
\l__xeCJK_sub_cancel_bool	837, 840, 842, 849, 851, 863
\l__xeCJK_sub_family_tl	1739, 1742, 1744, 1748, 1750, 1758, 1759, 1772
\l__xeCJK_sub_fontfeat_clist	1455, 1463, 1470, 1483
\l__xeCJK_sub_fontname_tl	1460, 1462, 1466, 1467, 1468, 1469, 1471, 1484
\l__xeCJK_sub_key_clist	1435, 1547, 1736, 1737
\l__xeCJK_tmpa_clist	41, 41, 1745, 1747, 1748, 1922, 1926, 1928, 1935, 1936, 1945, 1948, 1950
\l__xeCJK_tmpa_dim	36, 36, 1054, 1059, 1092, 1106, 1114, 1115, 1121, 1122, 1123, 1126, 1132, 1171, 1175, 1176, 1178
\l__xeCJK_tmpa_int	33, 33, 206, 211, 213, 214, 215, 1990, 1993, 1995, 1996
\l__xeCJK_tmpa_tl	39, 39, 1042, 1043, 1044, 1045, 1046, 1047, 1454, 1456, 1457, 1459, 1462, 1466, 1594, 1596, 1597, 1743, 1744, 1755, 1758, 1763, 1783, 1807, 1813, 1923, 1933, 1950, 2003, 2006, 2007, 2009
\l__xeCJK_tmpb_clist	41, 42, 1940, 1941, 1943
\l__xeCJK_tmpb_dim	36, 37, 1055, 1060, 1092, 1100, 1106, 1134
\l__xeCJK_tmpb_int	33, 34, 206, 211, 1990, 1993
\l__xeCJK_tmpb_tl	39, 40, 1929, 1932, 1933, 1936
\l__xeCJK_tmpc_clist	41, 43, 1942, 1946, 1948
\l__xeCJK_tmpc_dim	36, 38, 1056, 1062, 1072, 1074, 1075, 1088, 1089, 1091, 1096, 1098, 1100, 1105, 1114, 1115, 1121, 1122, 1123, 1126, 1129, 1135
\l__xeCJK_tmpc_int	33, 35, 207, 208, 214, 1991, 1995
\l__xeCJK_xecglue_bool	360, 414, 610, 615, 620, 628, 2074
\l_keys_choice_tl	956
\l_keys_key_tl	2136, 2137
\l_keys_value_tl	970, 1398, 1409, 1511, 1520
\l_peek_token	97, 108, 769, 770
\l_tmpa_int	1992, 1995
\l__xeCJK_family_tl	1343, 1345, 1357, 1358, 1788, 1794, 1804, 1807, 1808, 1809, 1812, 1813, 1836, 1843, 1844, 1919, 1923, 1929, 1940
\l__xeCJK_font_coor_tl	1035, 1290, 1788, 1788, 1791, 1793
\l__xeCJK_punctstyle_tl	956, 957, 1035, 1040, 1077, 1107, 1172, 1183, 1192, 1208, 2073
SlantFactor	4
AllowBreakBetweenPuncts	5
\LongPunct	978

## M

\makexeCJKactive	140, 144, 144
\makexeCJKinactive	141, 144, 145, 2255, 2258, 2260, 2264, 2279, 2304, 2309, 2346
\mathalpha	2020
\mathchar@type	1999
EmboldenFactor	4
\mddefault	2012
\MiddlePunct	978

\Mono	1503	\prg_return_false:	75, 120, 126, 580, 1145, 1156, 1164, 1657, 1662, 1702, 1823
\msg_critical:nn	10, 20	\prg_return_true:	75, 117, 120, 126, 580, 1145, 1156, 1164, 1657, 1662, 1702, 1827
\msg_error:nn	30, 130	\ProcessKeysOptions	2165
\msg_error:nnx	131	\prop_clear:c	1009
\msg_info:nnx	135	\prop_clear:N	1540
\msg_info:nnxx	136	\prop_get:Nn	1488, 1615
\msg_new:nnn	3, 11, 23, 129	\prop_get:NnNF	1616
\msg_redirect_module:nnn	2121, 2122, 2128, 2129	\prop_get:NnNT	1594, 1754, 1783
\msg_warning:nn	132	\prop_get:NnNTF	1830
\msg_warning:nnx	133	\prop_get:Nv	1358
\msg_warning:nnxx	134	\prop_get:NvN	1923
<b>N</b>			
indentfirst	3	\prop_get:NvNF	1371, 1940
\newCJKfontfamily	6, 1888, 1890	\prop_get:NvNT	1743, 1745, 2009
\NewDocumentCommand	144, 145, 179, 196, 199, 240, 303, 638, 643, 699, 708, 831, 838, 847, 875, 995, 997, 1140, 1310, 1321, 1330, 1361, 1831, 1888, 1890, 1898, 1913, 1917, 1961, 1964, 1984, 1988, 2042, 2056, 2068, 2087, 2103, 2173, 2178, 2180, 2182, 2183, 2184, 2289	\prop_get:NxNF	1825, 1929, 1931
\newXeTeXintercharclass	154	\prop_gpop:NnNT	1596
InlineEnv	4	\prop_gput:Nnn	176
InlineEnv+	4	\prop_gput:NnV	1576, 1757, 1783
InlineEnv-	4	\prop_gput:NvV	1730, 1731, 1744, 1748
\normalfont	1973, 1977, 2027	\prop_gput:Nvx	174
\normalspacedchars	10, 303, 303, 308	\prop_if_empty:NTF	2028
<b>O</b>			
LongPunct	5	\prop_if_in:cnTF	1163
LongPunct+	5	\prop_if_in:NnF	1617
LongPunct-	5	\prop_if_in:NvTF	208, 1919
Mono	6	\prop_map_inline:Nn	1495, 1541, 1630, 1706, 1714
<b>P</b>			
\par	769	\prop_new:c	1004
\PassOptionsToPackage	2124, 2131, 2137	\prop_new:N	170, 1489, 1719, 1720, 1721
\pdfstringdefPreHook	2274, 2277	\prop_put:cn	1011, 1017
\pdftex_strcmp:D	116, 119	\prop_put:Nnn	1490, 1491, 1492, 1493, 1494, 1614
\peek_after:Nw	89, 105	\prop_put:Nxx	1604
\peek_catcode:NTF	382, 650, 675, 718, 734, 779	\prop_put_if_new:Nnn	1618
\peek_catcode_remove:NTF	777	\prop_put_if_new:Nxx	1609
\peek_meaning_remove:NTF	784, 787	\prop_remove:cn	1024
\Pifont	2263	\PunctStyle	952
\prg_do_nothing:	85, 824, 1109, 2279, 2280, 2281	\punctstyle	31, 2182, 2182, 2183
\prg_new_conditional:cpnn	1161	\PunctWidth	978
\prg_new_conditional:Npnn	67, 114, 123, 577, 1142, 1147, 1158, 1654, 1659, 1699	<b>Q</b>	
\prg_new_protected_conditional:Npnn	1820	\q_no_value	2007
<b>R</b>			
\r	2259, 2260	\q_stop	70, 72
\RenewDocumentCommand	2263	<b>R</b>	
\RequirePackage	21, 22, 2166, 2356	\r	2259, 2260
\reset@font	1977	\RenewDocumentCommand	2263
\reverse_if:N	221	\RequirePackage	21, 22, 2166, 2356
\rmfamily	1974	\reset@font	1977

<b>S</b>	
<code>\scan_stop:</code> .....	487, 798, 2188
<code>\selectfont</code> .....	1583, 1764, 2051, 2060
<code>\seq_count:N</code> .....	2209
<code>\seq_gput_right:Nv</code> .....	156
<code>\seq_gput_right:Nx</code> .....	1862
<code>\seq_if_in:NnF</code> .....	2213
<code>\seq_if_in:NxF</code> .....	1860
<code>\seq_new:N</code> .....	158, 1866
<code>\setCJKfallbackfamilyfont</code> .	7, 8, 1361, 1361
<code>\setCJKfamilyfont</code> .....	6, 1885, 1888, 1888
<code>\setCJKmainfont</code> 5, 1881, 1961, 1961, 1963, 1979	
<code>\setCJKmathfont</code> .....	7, 1980, 1984, 1984
<code>\setCJKmonoexspace</code> .....	6, 2056, 2056
<code>\setCJKmonofont</code> ...	5, 1883, 1982, 2103, 2103
<code>\setCJKmonoscale</code> .....	6, 2042, 2042
<code>\setCJKromanfont</code> .....	1963, 1983
<code>\setCJKsansfont</code> ...	5, 1882, 1961, 1964, 1981
<code>\SetSymbolFont</code> .....	2013
<code>\settowidth</code> .....	605, 622
<code>\sffamily</code> .....	1975
<code>\shapedefault</code> .....	2012, 2014
<code>\skip_horizontal:N</code> .....	999, 2081
<code>\skip_horizontal:n</code> .....	491, 2082, 2147
<code>\skip_if_eq_p:nn</code> .....	362, 416
<code>\SlantedFeatures</code> .....	1487
<code>\SlantedFont</code> .....	1487
<code>\SlantFactor</code> .....	1388
<code>\SplitArgument</code> .....	197
<code>\str_case_x:nnn</code> .....	1301, 1870, 1879
<code>\str_if_eq_x:nnF</code> .....	189, 375, 386, 1288, 1801, 1805
<code>\str_if_eq_x:nnT</code> .....	1642
<code>\str_if_eq_x:nnTF</code> .....	894, 933, 1061, 1436, 1666, 1698
<code>\str_if_eq_x_p:nn</code> 471, 1283, 1661, 1673, 1674	
<b>T</b>	
<code>\tex_afterassignment:D</code> .....	98
<code>\tex_char:D</code> .....	2355
<code>\tex_font:D</code> .....	444, 445, 446, 1276, 1340, 1795, 2053, 2063
<code>\tex_fontdimen:D</code> ..	444, 445, 446, 2053, 2063
<code>\tex_ignorespaces:D</code> .....	56, 466, 1850, 1910, 1953, 2176, 2305
<code>\tex_kern:D</code> .....	499, 590, 591
<code>\tex_lastkern:D</code> .....	579
<code>\tex_lastskip:D</code> .....	362, 416, 417
<code>\tex_penalty:D</code> .....	44
<code>\tex_the:D</code> .....	313
<code>\tex_unkern:D</code> .....	653, 678, 2191
<code>\tex_unskip:D</code> .....	365, 420
<code>\tex_vrule:D</code> .....	484
<code>\textcentereddot</code> .....	2250
<code>\textellipsis</code> .....	2250
<code>\textemdash</code> .....	2250
<code>\textperiodcentered</code> .....	2250
<code>\textquotedblleft</code> .....	2251
<code>\textquotedblright</code> .....	2251
<code>\textquoteright</code> .....	2251
<code>\tipaencoding</code> .....	2258
<code>\tl_case:Nnn</code> .....	1077, 1107
<code>\tl_clear:c</code> .....	1543
<code>\tl_clear:N</code> .....	1372, 1527, 1538
<code>\tl_const:cn</code> .....	950
<code>\tl_const:Nn</code> .....	1986
<code>\tl_const:Nx</code> .....	128, 2167
<code>\tl_gput_right:Nn</code> .....	49, 51, 53, 2277
<code>\tl_gset:cn</code> .....	1141, 1311
<code>\tl_gset:cV</code> .....	1135, 1178
<code>\tl_gset:cx</code> .....	1131, 1133, 1260, 1262, 1264, 1272, 1290, 1907
<code>\tl_gset:Nx</code> .....	537, 1904
<code>\tl_gset_eq:cN</code> .	1043, 1044, 1045, 1046, 1047
<code>\tl_if_eq:NnF</code> .....	1172
<code>\tl_if_eq:NNT</code> .....	957
<code>\tl_if_eq:NNTF</code> .....	1040, 1183, 1192
<code>\tl_if_eq_p:NN</code> .....	1208
<code>\tl_if_exist:cF</code> .....	1038, 1050, 1169
<code>\tl_if_exist:cTF</code> .....	125, 1144, 1174
<code>\tl_if_exist:NF</code> ....	1966, 1967, 1968, 1969
<code>\tl_if_exist_p:c</code> ....	1151, 1152, 1153, 1154
<code>\tl_if_exist_p:N</code> .....	2274
<code>\tl_if_head_eq_charcode:nNTF</code> .....	1457
<code>\tl_if_head_is_group:nT</code> .....	1475
<code>\tl_if_in:nnTF</code> .....	1363
<code>\tl_if_single_token_p:n</code> .....	725
<code>\tl_map_function:nN</code> .....	63
<code>\tl_map_inline:cn</code> .....	1010, 1016
<code>\tl_map_inline:nn</code> .	62, 305, 1027, 1781, 2248
<code>\tl_map_inline:xn</code> .....	1021, 1141
<code>\tl_new:c</code> .....	77, 78, 79, 80, 1003, 1970
<code>\tl_new:N</code> .....	39, 40, 45, 46, 47, 819
<code>\tl_put_left:Nn</code> .....	55, 2245
<code>\tl_put_right:cx</code> .....	1015
<code>\tl_put_right:Nn</code> .....	56, 929, 1345, 1375, 1973, 1974, 1975, 1976, 2102, 2245
<code>\tl_put_right:Nx</code> .....	1432, 1808
<code>\tl_remove_all:cn</code> .....	1023
<code>\tl_remove_all:Nn</code> .....	1804
<code>\tl_replace_all:Nnx</code> .....	1469
<code>\tl_set:cn</code> .....	2286
<code>\tl_set:cx</code> .....	1008, 1471

\tl_set:Nn	86, 87, 442, 1035, 1477, 1788, 1966, 1967, 1968, 1969, 2351	\xeCJK_after_FullRight:	474, 507, 507
\tl_set:NV	1042	\xeCJK_before_FullLeft:N	501, 501, 518, 546, 565
\tl_set:Nx	503, 555, 574, 956, 1370, 1454, 1462, 1466, 1468, 1484, 1532, 1562, 1564, 1739, 1793, 1807, 1843, 1892, 1900, 1933, 2003, 2006, 2358, 2359	\xeCJK_Boundary_and_CJK:	389, 390, 390, 939, 2317
\tl_set_eq:NN	1844, 2007, 2073	\xeCJK_calc_kern:NN	1167, 1167, 1242
\tl_to_lowercase:n	64	\xeCJK_calc_kern:nNN	1193, 1199, 1215, 1243, 1243
\tl_use:c	178, 494, 495, 864, 1086, 1101, 1211, 1248, 1250, 1252, 1253, 1255, 1268, 1269, 1277, 1291, 1698, 1711	\xeCJK_calc_kern:VN	544, 552, 562, 571
\token_get_arg_spec:c	724	\xeCJK_calc_punct_dimen:N	1051, 1258, 1258
\token_get_arg_spec:N	816	\xeCJK_checksingl:N	705, 716, 716
\token_if_cs_p:c	660, 742, 754	\xeCJK_checksingl:NN	716, 719, 728, 732
\token_if_cs_p:N	814	\xeCJK_checksingl:NNN	716, 756, 760, 765
\token_if_eq_meaning_p:NN	769, 770	\xeCJK_checksingl_env:NNN	773, 775, 775
\token_if_math_toggle:cTF	668, 685	\xeCJK_checksingl_env:NNNN	775, 785, 788, 793
\token_if_math_toggle:NTF	818	\xeCJK_CJK_and_CJK:N	454, 455, 455, 704, 705, 713
\token_if_math_toggle_p:c	741, 753	\xeCJK_CJK_and_FullLeft:N	513, 513
\token_if_math_toggle_p:N	817	\xeCJK_CJK_and_FullRight:N	525, 525
\token_if_other_p:N	726	\xeCJK_def_node:nn	582, 582, 593, 594, 595, 596, 597
\token_if_space_p:c	740, 745	\xeCJK_dim_set_max:Nn	1031, 1031, 1129, 1238, 1240
\token_if_space_p:N	815	\xeCJK_dim_set_min:Nn	1031, 1033, 1115, 1123
\token_new:Nn	82	\xeCJK_fallback_testsymbol:N	1327, 1338, 1338, 1347
\token_to_str:N	25, 872, 1872, 1873, 1874, 1881, 1882, 1883, 1885, 1958, 2036	\xeCJK_family_if_exist:x	1820
\TrimSpaces	179, 876	\xeCJK_family_if_exist:xF	1809
\ttdefault	2051, 2060	\xeCJK_family_if_exist:xF	1752, 1779
\ttfamily	1976	\xeCJK_family_if_exist:xF	1343, 1811, 1820, 1841, 1854, 2002, 2005
<b>U</b>			
\UL@hook	2327, 2329	\xeCJK_family_if_exist_use:x	1836, 1839, 1852, 1852
PunctStyle	4	\xeCJK_FullLeft_and_FullLeft:N	540, 540
PunctWidth	5	\xeCJK_FullLeft_and_FullRight:N	548, 548
\Unicode	2354	\xeCJK_FullRight_and_FullLeft:N	558, 558
\use:c	463, 1826, 1845, 1855, 1992	\xeCJK_FullRight_and_FullRight:N	567, 567
\use:x	1949	\xeCJK_get_punct_bounds:nN	515, 527, 543, 551, 561, 570, 1036, 1036, 1139
\use_none:n	210, 220, 223, 226, 1537, 1573, 2282	\xeCJK_get_punct_bounds:nV	509
\use_none:nn	1835	\xeCJK_glyph_bounds:nN	1261, 1263, 1296, 1296
\usefont	2264	\xeCJK_if_blank_x:n	114
AutoFakeBold	4, 5	\xeCJK_if_blank_x:nF	1716
AutoFakeSlant	4, 5	\xeCJK_if_blank_x:nT	1460, 2229, 2235
AutoFallBack	4	\xeCJK_if_blank_x:nTF	114, 232, 1066, 1428, 1467, 1656, 1701, 1833
<b>V</b>			
\verbatim@font	2102	\xeCJK_if_blank_x_p:n	114, 230, 724
<b>X</b>			
\xCJKecglue	600	\xeCJK_if_last_node:n	577
\xe@alloc@intercharclass	2209, 2211	\xeCJK_if_last_node:nT	372, 652, 677
\xeCJK@family	1844	\xeCJK_if_last_node:nTF	371, 577, 2189
\xeCJK@setfont	1798		

\xeCJK_if_last_node_p:n	367, 401, 406, 408, 423, 425, 426, 432, 433, 577	\xeCJK_ULroutines:	2302, 2312, 2340
\xeCJK_if_outer_macro:c	67	\xeCJKactive	137
\xeCJK_if_outer_macro:cF	666, 683	\xeCJKallowbreakbetweenpuncts	958, 981, 995, 995
\xeCJK_if_outer_macro:cTF	57	\xeCJKCancelSubCJKBlock	10, 837, 838
\xeCJK_if_outer_macro_p:c	57, 659, 723, 739, 752	\xeCJKcaption	2287, 2289
\xeCJK_if_package_loaded:n	123	\xeCJKDeclareCharClass	10, 179, 179
\xeCJK_if_package_loaded:nF	2123, 2130	\xeCJKDeclareSubCharClass	833, 875, 875, 887
\xeCJK_if_package_loaded:nT	2261	\xeCJKDeclareSubCJKBlock	10, 831, 831, 836, 872
\xeCJK_if_package_loaded:nTF	123, 2135	\xeCJKdisablechecksingle	694, 698, 708
\xeCJK_if_package_loaded_p:n	123, 2273	\xeCJKdisablefallback	1316, 1320, 1330
\xeCJK_ignorespaces:	452, 641, 646, 2310, 2336	\xeCJKenablechecksingle	693, 698, 699
\xeCJK_int_until_do:nn	211, 218, 218, 1993	\xeCJKenablefallback	1315, 1320, 1321
\xeCJK_make_node:n	383, 384, 451, 582, 588, 653, 678, 2193, 2336	\xeCJKmainfont	2036
\xeCJK_no_break:	44, 44, 473, 530, 542, 550, 554, 573, 772, 780, 797, 998, 2304	\xeCJKnobreakbetweenpuncts	982, 995, 997
\xeCJK_peek_catcode_ignore_spaces:Nf	93, 679	\xeCJKplainchr	2182, 2183
\xeCJK_peek_catcode_ignore_spaces:NT	91	\xeCJKResetPunctClass	10, 182, 202, 240, 240, 302, 834, 844, 853
\xeCJK_peek_catcode_ignore_spaces:Nf	82, 83, 92, 94, 654, 747	\xeCJKRestoreSubCJKBlock	10, 837, 847
\xeCJK_select_font:	344, 395, 462, 1051, 1346, 1788, 1789, 1798, 1817, 2335, 2339, 2346	\xeCJKsetcharclass	199, 199
\xeCJK_set_family:xcc	1378, 1772	\xeCJKsetecglue	2185
\xeCJK_set_family:xnn	1365, 1558, 1558, 1591, 1889, 1893, 1906, 1962, 1965, 1985, 2106, 2109, 2110	\xeCJKsetemboldenfactor	2178, 2178
\xeCJK_set_family_fallback:xnn	1361, 1364, 1367	\xeCJKsetkern	11, 1310, 1310
\xeCJK_set_mathfont:	2000, 2000, 2030	\xeCJKsetmathcode	1988, 1988, 2019
\xeCJK_set_monoexspace:	1526, 2056, 2057, 2077	\xeCJKsetslantfactor	2178, 2180
\xeCJK_set_monoscale:	1531, 2045, 2048	\xeCJKsetup	3, 2078, 2092, 2173, 2173, 2179, 2181, 2182, 2184
\xeCJK_ULpostpunctchar:n	2302, 2307, 2321	\xeCJKsetwidth	11, 1140, 1140
\xeCJK_ULprepunctchar:n	2302, 2302, 2320	\XeTeX@CJKfntef@hook	2343, 2345
		\xetex_if_engine:F	10
		\XeTeXcharclass	214, 306
		\XeTeXcharglyph	1307
		\XeTeXdefaultencoding	2291, 2300
		\XeTeXglyphbounds	25, 28, 1300
		\XeTeXinterchartokenstate	144, 145, 2200
		\XeTeXinterchartoks	310, 313
		\XeTeXmathcode	1995