

xeCJK 宏包

ctex.org*

2012/07/20 v3.0.7

目录

1	简介	2
2	基本用法	2
3	用户手册	3
3.1	宏包选项	3
3.2	字体设置与选择	5
3.2.1	X _Y TeX 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	10
4	已知问题和兼容性	11
5	xeCJK 代码实现	11
5.1	运行环境检查	11
5.2	内部工具	12
5.3	功能开关	14
5.4	字符类别设定	15
5.5	字符输出规则	21
5.6	段末孤字处理	29
5.7	增加 CJK 子分区	32
5.8	标点处理	35
5.9	后备字体	42
5.10	CJK 字体族声明方式	44
5.11	字体切换	53
5.12	数学字体设置	57
5.13	处理等宽字体和抄录环境	58
5.14	xeCJK 其它选项	60
5.15	xeCJK 初始化设置	60
5.16	兼容性修补	62

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

6	例子	65
6.1	xeCJK-example-autofake.tex	65
6.2	xeCJK-example-fallback.tex	66
6.3	xeCJK-example-subCJKblock.tex	67
6.4	xeCJK-example-CJKecglue.tex	69
6.5	xeCJK-example-checksingle.tex	69

1 简介

xeCJK 是一个 \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 \LaTeX 的一些最新特性,需要 \LaTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 的依赖 \LaTeX 3 项目的 expl3, xparse 和 l3keys2e 宏包,并依赖 fontspec¹ 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-kit² 项目进行维护,目前主要维护者是刘海洋³ 和李清⁴。

2 基本用法

与其他 \LaTeX 宏包一样,引入 xecjk 宏包只要在导言区使用

```
\usepackage{xeCJK}
```

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

可以在各种文档类中使用 xecjk 宏包,最简单的示例是:

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

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

上述示例设置了中文字体 SimSun(宋体)。运行此示例要求系统安装了设置的字体,源文件用 UTF-8 编码保存,使用 \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⁵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。

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

²<http://code.google.com/p/ctex-kit/>

³leoliu.pku@gmail.com

⁴sobenlee@gmail.com

⁵<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 {$\langle key1 \rangle = \langle var1 \rangle$, $\langle key2 \rangle = \langle var2 \rangle$, ...}</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 = $\langle true false \rangle$</code>
--------------------------	--

打开/关闭对中文的特殊处理。

<code>CJKnumber \star</code>	<code>CJKnumber = $\langle true false \rangle$</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 \star</code>	<code>indentfirst = $\langle true false \rangle$</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 = $\langle true false \rangle$</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 \star</code>	<code>CJKmath = $\langle true false \rangle$</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 = $\{\backslash hspace\{0pt plus .08\baselineskip\}\}$</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>}	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。
<hr/> xCJKe glue <hr/>	xCJKe glue = {<true false glue>}	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。
<hr/> CheckSingle <hr/>	CheckSingle = <true false>	是否避免单个 CJK 文字单独占一个段落的最后一行。
<hr/> InlineEnv InlineEnv+ InlineEnv- <hr/>	InlineEnv = {env1, env2, env3, ...}	在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 L ^A T _E 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 ^A T _E 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。
<hr/> AutoFallBack <hr/>	AutoFallBack = <true false>	当文档中有个别生僻字时,可以使用这个选项,自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。
<hr/> AutoFakeBold ☆ <hr/>	AutoFakeBold = {<true false 数字>}	全局设定当没有声明对应的粗体时,是否使用 伪粗体 ;当输入的是数字时,将使用伪粗体,并将使用输入的数字作为伪粗体的默认粗细程度。
<hr/> AutoFakeSlant ☆ <hr/>	AutoFakeSlant = {<true false 数字>}	全局设定当没有声明对应的斜体时,是否使用 伪斜体 ;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。
<hr/> EmboldenFactor ☆ <hr/>	EmboldenFactor = {<数字 4>}	设置伪粗体的默认粗细程度。
<hr/> SlantFactor ☆ <hr/>	SlantFactor = {<数字 0.167>}	设置伪斜体的粗细程度,范围是 -0.999 ~ 0.999。
<hr/> PunctStyle <hr/>	PunctStyle = {<quanjiao banjiao kaiming hangmoban CCT plain>}	设置标点处理格式,有效值分别为 quanjiao 全角式:所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度; banjiao 半角式:所有标点占半个汉字宽度; kaiming 开明式:句末点号用全角,其他半角; hangmoban 行末半角式:所有标点占一个汉字宽度,行首行末对齐; CCT CCT 格式; plain 原样(不调整标点间距)。

<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 dimen 的单位最好用 em 等相对距离单位,而不建议使用诸如 pt 之类的绝对距离单位。这里的设置对除了 plain 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 \xeCJKsetwidth。		

<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/> Mono <hr/>	<p><code>Mono = <Scale> Exspace</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/> <hr/> <code>\CJKflexiblespacing</code> <hr/>	<p><code>\CJKfixedspacing</code></p> <p><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/> <hr/> <code>\setCJKmonoexspace</code> <hr/>	<p><code>\setCJKmonoscale</code></p> <p><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 <{family}> [<{font features}>] <{font name}></code></p> <p>声明新的 CJK 字体族 <i><family></i> 并指定字体。</p>
<hr/> <code>\CJKfamily</code> <hr/>	<p><code>\CJKfamily <{family}></code></p> <p>用于在文档中切换 CJK 字体族, <i><family></i> 必须预先声明。</p>
<hr/> <code>\newCJKfontfamily</code> ★ <hr/>	<p><code>\newCJKfontfamily [<{family}>] \<{font-switch}> [<{font features}>] <{font name}></code></p> <p>声明新的 CJK 字体族 <i><family></i> 并指定字体,并定义 <code>\<{font-switch}></code>,在文档中可以使用它来切换 CJK 字体族。可以不必指定 <i><family></i>,这时候 <i><family></i> 将等于 <i><font-switch></i>。事实上, <code>\newCJKfontfamily</code> 是 <code>\setCJKfamilyfont</code> 和 <code>\CJKfamily</code> 的合并。例如</p> <pre> \newCJKfontfamily[song]\songti{SimSun} </pre> <p>等价于</p> <pre> \setCJKfamilyfont{song}{SimSun} \newcommand\songti{\CJKfamily{song}} </pre>
<hr/> <code>\CJKfontspec</code> <hr/>	<p><code>\CJKfontspec [<{font features}>] <{font name}></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/>	<p><code>\defaultCJKfontfeatures <{font features}></code></p> <p>全局设置 CJK 字体族的默认选项。例如,使用</p> <pre> \defaultCJKfontfeatures{Scale=0.962216} </pre> <p>可以将全部 CJK 字体缩小为 0.962216。</p>

<hr/> <hr/>	<p><code>\addCJKfontfeatures</code> <code>\addCJKfontfeatures {}</code></p> <p>临时增加当前使用的 CJK 字体的选项。例如, 使用</p> <pre>\addCJKfontfeatures{Scale=1.1}</pre> <p>可以将文档中当前使用的 CJK 字体放大为 1.1。</p>
<hr/> <hr/>	<p><code>\CJKrmdefault</code> 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>。</p>
<hr/> <hr/>	<p><code>\CJKsfdefault</code> 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>。</p>
<hr/> <hr/>	<p><code>\CJKttdefault</code> 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>。</p>
<hr/> <hr/>	<p><code>\CJKfamilydefault</code> 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</p> <pre>\renewcommand\familydefault{\sfdefault} \renewcommand\CJKfamilydefault{\CJKsfdefault}</pre> <p>可以将全文的 CJK 和西文默认字体改为无衬线字体族。</p>
<hr/> <hr/>	<p><code>\setCJKmathfont</code> ★ <code>\setCJKmathfont [] {}</code></p> <p>设置数学公式中的 CJK 字体族。如果使用了 <code>CJKmath</code> 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。</p>
<hr/> <hr/>	<p><code>\setCJKfallbackfamilyfont</code> ★ <code>\setCJKfallbackfamilyfont {<family>} [] {}</code></p> <p>设置 CJK 字体族 <code><family></code> 的备用字体。例如, 使用</p> <pre>\setCJKmainfont{SimSun} \setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}</pre> <p>可以将 <code>SimSun-ExtB</code> 作为 <code>SimSun</code> 的备用字体。</p>
<hr/> <hr/>	<p><code>FallBack</code> <code>FallBack = { [] {} }</code></p> <p><code>xeCJK</code> 在 <code></code> 里增加了 <code>FallBack</code> 这个选项。用来在声明主字体时, 同时设置备用字体。例如, 上面的例子等价于:</p> <pre>\setCJKmainfont[FallBack=SimSun-ExtB]{SimSun}</pre> <p>如果 <code>FallBack</code> 的值为空, 将设置的是备用字体。例如,</p> <pre>\setCJKmainfont[FallBack,AutoFakeBold,Scale=.97]{SimSun-ExtB}</pre> <p>等价于</p> <pre>\setCJKfallbackfamilyfont{\CJKrmdefault}[AutoFakeBold,Scale=.97]{SimSun-ExtB}</pre>

```
\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_YTeX 的字体名查找

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

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

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

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

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

```
Times New Roman:style=cursiva,kurzíva,kursiv,Πλάγια,Italic,
Kursivoitu,Italique,Dőlt,Corsivo,Cursief,kursywa,Itálico,Курсив,
Ítalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta cursiva,tučné kurzíva,fed kursiv,
Fett Kursiv,Έντρονα Πλάγια,Bold Italic,Negrita Cursiva,
Lihavoitu Kursivoi,Gras Italique,Félkövéř dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrito Itálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın Ítalik,
Krepko poševno,nghiêng đặm,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντρονα,Bold,Negrita,
Lihavoitu,Gras,Félkövéř,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
Полужирный,Fet,Kalın,Krepko,đặm,Lodia
Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,
thườg,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⁶。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。

```
\xeCJKDeclareSubCJKBlock * \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 `` 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

⁶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。

`\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。

<code>\xeCJKCancelSubCJKBlock</code>	<code>\xeCJKCancelSubCJKBlock {<block1,block2,...>}</code>
--------------------------------------	--

在文档中取消对 CJK 分区的声明。

<code>\xeCJKRestoreSubCJKBlock</code>	<code>\xeCJKRestoreSubCJKBlock {<block1,block2,...>}</code>
---------------------------------------	---

在文档中恢复对 CJK 分区的声明。

3.4 设置 CJK 字符范围

<code>\xeCJKDeclareCharClass</code> ★	<code>\xeCJKDeclareCharClass {<class>} {<class range>}</code>
---------------------------------------	---

`<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, 不要轻易改变字符类别。

<code>\xeCJKResetPunctClass</code> ★	<code>\xeCJKResetPunctClass</code>
--------------------------------------	------------------------------------

用于重置标点符号所属的字符类。使用 `\xeCJKDeclareCharClass` 更改字符类别之后, 为了确保标点处理的正确性, 应该在它之后使用 `\xeCJKResetPunctClass`。

<code>\normalspacedchars</code>	<code>\normalspacedchars {<char list>}</code>
---------------------------------	---

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。

<code>\xeCJKsetwidth</code> ★	<code>\xeCJKsetwidth {<标点列表>} {<dimen>}</code>
-------------------------------	--

`<标点列表>` 可以是单个标点, 也可以是多个标点。例如,

`\xeCJKsetwidth{。?}{0.7em}`

将设置句号和问号所占的宽度为 0.7em。

<code>\xeCJKsetkern</code> ★	<code>\xeCJKsetkern {<标点 1>} {<标点 2>} {<dimen>}</code>
------------------------------	--

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 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 \ProvidesExplPackage{\ExplFileName}{\ExplFileDate}{3.0.7}{\ExplFileDescription}
```

5.1 运行环境检查

xeCJK 必须使用 XeTeX 引擎的支持。

```
3 \msg_new:nnn { xecjk } { Require-XeTeX }
4 {
5   The~xeCJK~package~requires~XeTeX~to~function.\\
6   You~must~change~your~typesetting~engine~to~"xelatex"~\\
7   instead~of~plain~"latex"~or~"pdflatex"~or~"lualatex".\\
8   Loading~xeCJK~will~abort!
9 }
10 \xetex_if_engine:F { \msg_critical:nn { xecjk } { Require-XeTeX } }
```

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 } { 2012/07/16 } { }
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\c_space_tl 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_tmpb_int 33 \int_new:N \l_xeCJK_tmpa_int
\l_xeCJK_tmppc_int 34 \int_new:N \l_xeCJK_tmpb_int
35 \int_new:N \l_xeCJK_tmppc_int
(End definition for \l_xeCJK_tmpa_int, \l_xeCJK_tmpb_int, and \l_xeCJK_tmppc_int)
```

```
\l_xeCJK_tmpa_dim
\l_xeCJK_tmpb_dim 36 \dim_new:N \l_xeCJK_tmpa_dim
\l_xeCJK_tmppc_dim 37 \dim_new:N \l_xeCJK_tmpb_dim
38 \dim_new:N \l_xeCJK_tmppc_dim
(End definition for \l_xeCJK_tmpa_dim, \l_xeCJK_tmpb_dim, and \l_xeCJK_tmppc_dim)
```

```
\l_xeCJK_tmpa_tl
\l_xeCJK_tmpb_tl 39 \tl_new:N \l_xeCJK_tmpa_tl
40 \tl_new:N \l_xeCJK_tmpb_tl
(End definition for \l_xeCJK_tmpa_tl and \l_xeCJK_tmpb_tl)
```

```
\l_xeCJK_tmpa_clist
\l_xeCJK_tmpb_clist 41 \clist_new:N \l_xeCJK_tmpa_clist
\l_xeCJK_tmppc_clist 42 \clist_new:N \l_xeCJK_tmpb_clist
43 \clist_new:N \l_xeCJK_tmppc_clist
(End definition for \l_xeCJK_tmpa_clist, \l_xeCJK_tmpb_clist, and \l_xeCJK_tmppc_clist)
```

```
\xeCJK_AtEndPreamble:n 在 \document 前后加上各种钩子。
\xeCJK_AfterPreamble:n 44 \tl_new:N \g_xeCJK_at_end_preamble_hook
\xeCJK_AfterEndPreamble:n 45 \tl_new:N \g_xeCJK_after_preamble_hook
46 \tl_new:N \g_xeCJK_after_end_preamble_hook
47 \cs_new:Nn \xeCJK_AtEndPreamble:n
48 { \tl_gput_right:Nn \g_xeCJK_at_end_preamble_hook {#1} }
49 \cs_new:Nn \xeCJK_AfterPreamble:n
50 { \tl_gput_right:Nn \g_xeCJK_after_preamble_hook {#1} }
51 \cs_new:Nn \xeCJK_AfterEndPreamble:n
52 { \tl_gput_right:Nn \g_xeCJK_after_end_preamble_hook {#1} }
53 \AtBeginDocument { \g_xeCJK_after_preamble_hook }
54 \tl_put_left:Nn \document { \group_end: \g_xeCJK_at_end_preamble_hook \group_begin: }
55 \tl_put_right:Nn \document { \g_xeCJK_after_end_preamble_hook \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 56 \group_begin:
57 \char_set_lccode:nn { '\Z } { '\t }
58 \char_set_lccode:nn { '\Y } { '\c }
59 \char_set_lccode:nn { '\; } { '\: }
60 \char_set_lccode:nn { '\ / } { '\ \ }
61 \tl_map_inline:nn { \T \F \C \J \K } { \char_set_lccode:nn { '#1 } { '#1 } }
62 \tl_map_function:nN { \O \U \Z \E \R \M \A \Y \R \; \ / } \char_set_catcode_other:N
63 \tl_to_lowercase:n
64 {
```

```

65 \group_end:
66 \prg_new_conditional:Nnn \xeCJK_if_outer_macro:c { p , T , F , TF }
67 {
68   \exp_last_unbraced:Nf \xeCJK_if_outer_macro_aux:w
69   { \cs_meaning:c {#1} } OUZER ~ MAYRO ; \q_stop
70 }
71 \cs_new_nopar:Npn \xeCJK_if_outer_macro_aux:w #1 OUZER ~ MAYRO ; #2 \q_stop
72 {
73   \if_cs_exist:w c_xeCJK_prefix_ #1 _aux_tl \cs_end:
74   \prg_return_true: \else: \prg_return_false: \fi:
75 }
76 \tl_new:c { c_xeCJK_prefix_/_aux_tl }
77 \tl_new:c { c_xeCJK_prefix_/long/_aux_tl }
78 \tl_new:c { c_xeCJK_prefix_/protected/_aux_tl }
79 \tl_new:c { c_xeCJK_prefix_/protected/long/_aux_tl }
80 }

```

(End definition for \xeCJK_if_outer_macro:c)

\xeCJK_peek_catcode_ignore_spaces:NTF 对 I3token 打的补丁,使 \xeCJK_peek_catcode_ignore_spaces:NTF 可用于 \outer 宏。

```

81 \token_new:Nn \l_xeCJK_peek_search_token { ? }
82 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
83 {
84   \cs_set_eq:NN \l_xeCJK_peek_search_token #1
85   \cs_set:Npx \xeCJK_peek_true:w
86   {
87     \exp_not:N \group_align_safe_end:
88     \exp_not:n {#2}
89   }
90   \cs_set:Npx \xeCJK_peek_false:w
91   {
92     \exp_not:N \group_align_safe_end:
93     \exp_not:n {#3}
94   }
95   \group_align_safe_begin:
96   \peek_after:Nw \xeCJK_peek_ignore_spaces_execute_branches:
97 }
98 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NT #1#2
99 { \xeCJK_peek_catcode_ignore_spaces:NTF #1 {#2} { } }
100 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NF #1#2
101 { \xeCJK_peek_catcode_ignore_spaces:NTF #1 { } {#2} }
102 \cs_new_nopar:Nn \xeCJK_peek_ignore_spaces_execute_branches:
103 {
104   \if_meaning:w \l_peek_token \c_space_token
105   \tex_afterassignment:D \xeCJK_peek_ignore_spaces_execute_branches_aux:
106   \exp_after:wN \cs_set_eq:NN \exp_after:wN \xeCJK_peek_tmp:w
107   \else:
108   \exp_after:wN \xeCJK_peek_execute_branches_catcode:
109   \fi:
110 }
111 \cs_new_protected_nopar:Nn \xeCJK_peek_ignore_spaces_execute_branches_aux:
112 { \peek_after:Nw \xeCJK_peek_ignore_spaces_execute_branches: }
113 \cs_new_nopar:Nn \xeCJK_peek_execute_branches_catcode:

```

```

114 {
115   \if_catcode:w \exp_not:N \l_peek_token \exp_not:N \l_xeCJK_peek_search_token
116   \exp_after:wN \xeCJK_peek_true:w
117   \else:
118     \exp_after:wN \xeCJK_peek_false:w
119   \fi:
120 }
(End definition for \xeCJK_peek_catcode_ignore_spaces:NTF)

```

`\xeCJK_tl_if_blank:p:x` 判断是否为空或者仅含一个空格。

```

\exeCJK_tl_if_blank:xTF 121 \prg_new_conditional:Nnn \xeCJK_tl_if_blank:x { p , T , F , TF }
122 {
123   \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
124   \prg_return_true:
125   \else:
126     \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
127     \prg_return_true: \else: \prg_return_false: \fi:
128   \fi:
129 }
(End definition for \xeCJK_tl_if_blank:x)

```

`\xeCJK_msg_new:nn` 各种信息函数的缩略形式。

```

\exeCJK_error:n 130 \cs_new_nopar:Npn \xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\exeCJK_error:nx 131 \cs_new_nopar:Npn \xeCJK_error:n { \msg_error:nn { xeCJK } }
\exeCJK_warning:nx 132 \cs_new_nopar:Npn \xeCJK_error:nx { \msg_error:nnx { xeCJK } }
\exeCJK_info:nx 133 \cs_new_nopar:Npn \xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\exeCJK_info:nxx 134 \cs_new_nopar:Npn \xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
135 \cs_new_nopar:Npn \xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
136 \cs_new_nopar:Npn \xeCJK_info:nx { \msg_info:nnx { xeCJK } }
137 \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

```

138 \keys_define:nn { xeCJK / options }
139 {
140   xeCJKactive .choice:,
141   xeCJKactive / true .code:n = { \makexeCJKactive } ,
142   xeCJKactive / false .code:n = { \makexeCJKinactive } ,
143   xeCJKactive .default:n = { true },
144 }

```

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

`\makexeCJKactive`

```

\makexeCJKinactive 145 \NewDocumentCommand \makexeCJKactive { } { \XeTeXinterchartokenstate = \c_one }
146 \NewDocumentCommand \makexeCJKinactive { } { \XeTeXinterchartokenstate = \c_zero }
(End definition for \makexeCJKactive and \makexeCJKinactive)

```

抑制 BOM。

```

147 \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 $\text{Xe}_{\text{L}}\text{TeX}$ 中预定义类别, `xeCJK` 新增加了 `HalfLeft`、`HalfRight`、`NormalSpace`。

类别	说明	例子
<code>Default</code>	西文一般符号	abc123
<code>CJK</code>	CJK 表意符号	汉字あいう
<code>FullLeft</code>	全角左标点	(《: “
<code>FullRight</code>	全角右标点	, 。) 》”
<code>HalfLeft</code>	半角左标点	([{
<code>HalfRight</code>	半角右标点	,. ?)] }
<code>NormalSpace</code>	前后原始间距的符号	/
<code>Boundary</code>	边界	空格

`\xeCJK_Default_class` 这五类是 $\text{Xe}_{\text{L}}\text{TeX}$ 预定义的类别。

```

\xeCJK_CJK_class      148 \int_const:Nn \xeCJK_Default_class { 0 }
\xeCJK_FullLeft_class 149 \int_const:Nn \xeCJK_CJK_class      { 1 }
\xeCJK_FullRight_class 150 \int_const:Nn \xeCJK_FullLeft_class   { 2 }
\xeCJK_Boundary_class 151 \int_const:Nn \xeCJK_FullRight_class { 3 }
                      152 \int_const:Nn \xeCJK_Boundary_class { 255 }

```

(End definition for \xeCJK_Default_class and others.)

`\xeCJK_new_class:n` 新建一个字符类别。

```

\g_xeCJK_class_seq 153 \cs_new_nopar:Nn \xeCJK_new_class:n
                      154 {
                      155   \exp_args:Nc \newXeTeXintercharclass { xeCJK_#1_class }
                      156   \clist_gclear_new:c { g_xeCJK_#1_range_clist }
                      157   \seq_gput_right:Nv \g_xeCJK_class_seq { xeCJK_#1_class }
                      158 }
                      159 \seq_new:N \g_xeCJK_class_seq

```

(End definition for \xeCJK_new_class:n and \g_xeCJK_class_seq)

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

```

\xeCJK_HalfRight_class 160 \xeCJK_new_class:n { HalfLeft }
\xeCJK_NormalSpace_class 161 \xeCJK_new_class:n { HalfRight }
                        162 \xeCJK_new_class:n { NormalSpace }

```

(End definition for \xeCJK_HalfLeft_class, \xeCJK_HalfRight_class, and \xeCJK_NormalSpace_class)

`\g_xeCJK_base_class_clist`

```

\g_xeCJK_CJK_class_clist 163 \clist_set:Nn \g_xeCJK_base_class_clist
\g_xeCJK_CJK_class_prop 164 {
                        165   Default, CJK, FullLeft, FullRight, Boundary,
                        166   HalfLeft, HalfRight, NormalSpace
                        167 }
                        168 \clist_map_inline:Nn \g_xeCJK_base_class_clist
                        169 { \clist_gclear_new:c { g_xeCJK_#1_range_clist } }
                        170 \clist_new:N \g_xeCJK_CJK_class_clist
                        171 \prop_new:N \g_xeCJK_CJK_class_prop
                        172 \cs_new_nopar:Nn \xeCJK_save_CJK_class:n
                        173 {

```

```

174 \clist_gput_right:Nx \g_xeCJK_CJK_class_clist {#1}
175 \prop_gput:Nvx \g_xeCJK_CJK_class_prop { xeCJK_#1_class } {#1}
176 }
177 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
178 \clist_map_inline:nn { CJK, FullLeft, FullRight } { \xeCJK_save_CJK_class:n {#1} }
(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。

```

179 \cs_new_nopar:Nn \xeCJK_class_num:n { \tl_use:c { xeCJK_#1_class } }
(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_setcharclass_aux:Nn`

```

180 \NewDocumentCommand \xeCJKDeclareCharClass { > { \TrimSpaces } m m }
181 {
182   \clist_gput_right:cx { g_xeCJK_#1_range_clist } {#2}
183   \clist_map_inline:xn {#2}
184   {
185     \str_if_eq:x:nnF {##1} { -> }
186     {
187       \xeCJK_setcharclass_aux:Nn \xeCJKsetcharclass {##1}
188       { \xeCJK_class_num:n {#1} }
189     }
190   }
191 }
192 \NewDocumentCommand \xeCJK_setcharclass_aux:Nn
193 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
194 \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。

```

195 \NewDocumentCommand \xeCJKsetcharclass { m m m }
196 {
197   \xeCJK_check_num_range:nnNN {#1} {#2} \l_xeCJK_tmpa_int \l_xeCJK_tmpb_int
198   \int_set:Nn \l_xeCJK_tmpc_int {#3}
199   \prop_if_in:NVTF \g_xeCJK_CJK_class_prop \l_xeCJK_tmpc_int
200   { \cs_set_eq:NN \xeCJK_set_catcode:n \char_set_catcode_other:n }
201   { \cs_set_eq:NN \xeCJK_set_catcode:n \use_none:n }
202   \loop
203     \xeCJK_set_catcode:n \l_xeCJK_tmpa_int
204     \XeTeXcharclass \l_xeCJK_tmpa_int = \l_xeCJK_tmpc_int
205     \if_int_compare:w \l_xeCJK_tmpa_int < \l_xeCJK_tmpb_int
206       \int_incr:N \l_xeCJK_tmpa_int
207   \repeat
208 }
(End definition for \xeCJKsetcharclass)

```

`\xeCJK_check_num_range:nnNN`

```

209 \cs_new_nopar:Nn \xeCJK_check_num_range:nnNN
210 {
211   \bool_if:nTF { \xeCJK_tl_if_blank_p:x {#1} || \xeCJK_tl_if_blank_p:x {#2} }
212   {

```

```

213     \int_set:Nn #3 { \xCJK_tl_if_blank:xTF {#1} {#2} {#1} }
214     \int_set_eq:NN #3 #4
215   }
216   {
217     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
218     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
219   }
220 }

```

(End definition for \xCJK_check_num_range:nnNN)

\xCJKResetPunctClass 用于重置标点符号所属的字符类。

```

221 \NewDocumentCommand \xCJKResetPunctClass { }
222 {
223   \xCJKDeclareCharClass { HalfLeft }
224   { "28 , "2D , "5B , "60 , "7B }
225   \xCJKDeclareCharClass { HalfRight }
226   { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , }
227   \xCJKDeclareCharClass { FullLeft }
228   {
229     "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
230     "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
231     "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
232     "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6 ,
233   }
234   \xCJKDeclareCharClass { FullRight }
235   {
236     "00B7 , "2019 , "201D , "2014 , "2025 , "2026 , "2030 , "2500 , "3001 ,
237     "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 , "3015 ,
238     "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 , "3047 ,
239     "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C , "309D ,
240     "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
241     "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE , "FE50 ,
242     "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C , "FE5E ,
243     "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B , "FF1F ,
244     "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 , "FF69 ,
245     "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E , "FF9F ,
246   }
247 }

```

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

设置 CJK 文字的范围。

```

248 \xCJKDeclareCharClass { CJK }
249 {

```

- Hangul Jamo (谚文字母)

```

250     "1100 -> "11FF ,

```

- CJK Radicals Supplement (中日韩部首补充)

```

251     "2E80 -> "2EFF ,

```

- Kangxi Radicals (康熙字典部首)

```

252     "2F00 -> "2FDF ,

```

	• Ideographic Description Characters (表意文字描述符)
253	"2FF0 -> "2FFF ,
	• CJK Symbols and Punctuation (中日韩符号和标点)
254	"3000 -> "303F ,
	• Hiragana (日文平假名)
255	"3040 -> "309F ,
	• Katakana (日文片假名)
256	"30A0 -> "30FF ,
	• Bopomofo (注音字母)
257	"3100 -> "312F ,
	• Hangul Compatibility Jamo (谚文兼容字母)
258	"3130 -> "318F ,
	• Kanbun (象形字注释标志)
259	"3190 -> "319F ,
	• Bopomofo Extended (注音字母扩展)
260	"31A0 -> "31BF ,
	• CJK Strokes (中日韩笔画)
261	"31C0 -> "31EF ,
	• Katakana Phonetic Extensions (日文片假名语音扩展)
262	"31F0 -> "31FF ,
	• Enclosed CJK Letters and Months (带圈中日韩字母和月份)
263	"3200 -> "32FF ,
	• CJK Compatibility (中日韩兼容)
264	"3300 -> "33FF ,
	• CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
265	"3400 -> "4DBF ,
	• Yijing Hexagrams Symbols (易经六十四卦符号)
266	"4DC0 -> "4DFF ,
	• CJK Unified Ideographs (中日韩统一表意文字)
267	"4E00 -> "9FFF ,
	• Yi Syllables (彝文音节)
268	"A000 -> "A48F ,

- Yi Radicals (彝文字根)
269 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
270 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
271 "AC00 -> "D7AF ,
- Kana Supplement (日文假名增补)
272 "B000 -> "B0FF ,
- Hangul Jamo Extended-B (谚文扩展 B)
273 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
274 "F900 -> "FAFF ,
- CJK Compatibility Forms (中日韩兼容形式)
275 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)
276 "FF00 -> "FFEF ,
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)
277 "20000 -> "2A6DF ,
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)
278 "2A700 -> "2B73F ,
- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)
279 "2B740 -> "2B81F ,
- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)
280 "2F800 -> "2FA1F ,
281 }

重置标点符号的字符类。

282 \xeCJKResetPunctClass

\normalspacedchars 声明前后不加间距的字符。

```
283 \NewDocumentCommand \normalspacedchars { m }
284 {
285   \tl_map_inline:nn {#1}
286     { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
287 }
288 \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。

```

289 \cs_new_nopar:Nn \xeCJK_inter_class_toks:nnn
290 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = {#3} }
291 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc }
(End definition for \xeCJK_inter_class_toks:nnn and \xeCJK_inter_class_toks:nnc)

```

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

```

292 \cs_new_nopar:Nn \xeCJK_clear_inter_class_toks:nn
293 {
294   \XeTeXinterchartoks
295   \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = { \c_empty_tl }
296 }
(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。

```

297 \cs_new_nopar:Nn \xeCJK_pre_inter_class_toks:nnn
298 {
299   \use:x
300   {
301     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
302     {
303       \exp_not:n {#3} \tex_the:D \XeTeXinterchartoks
304       \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2}
305     }
306   }
307 }
(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。

```

308 \cs_new_nopar:Nn \xeCJK_app_inter_class_toks:nnn
309 {
310   \use:x
311   {
312     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
313     {
314       \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1}
315       \xeCJK_class_num:n {#2} \exp_not:n {#3}
316     }
317   }
318 }
319 \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_get_inter_class_toks:nn` 取出相邻类别之间的内容。

```

320 \cs_new_nopar:Nn \xeCJK_get_inter_class_toks:nn
321 { \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} }
(End definition for \xeCJK_get_inter_class_toks:nn)

```

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

```

322 \cs_new_nopar:Nn \xeCJK_copy_inter_class_toks:nnnn
323 {
324   \use:x
325   {
326     \xeCJK_inter_class_toks:nnn {#1} {#2}
327     { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
328   }
329 }
(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。

```

330 \cs_new_nopar:Nn \xeCJK_clear_CJK_toks:
331 {
332   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
333   { \xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
334 }
(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	✓	✓	✓	✓	✓		✓	

```

335 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
336 {
337   \clist_map_inline:nn { CJK, FullLeft, FullRight }
338   {
339     \xeCJK_inter_class_toks:nnn {#1} {##1}
340     {
341       \c_group_begin_token
342       \xeCJK_select_font:
343       \xeCJK_clear_inter_class_toks:nn {#1} {##1}
344       \xeCJK_clear_CJK_toks:
345     }
346     \xeCJK_inter_class_toks:nnn {##1} {#1} { \c_group_end_token }
347   }
348   \xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
349   \clist_map_inline:nn { FullLeft, FullRight }
350   { \xeCJK_app_inter_class_toks:nnc {#1} {##1} { xeCJK_CJK_and_##1:N } }
351 }
352 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
353 {
354   \xeCJK_inter_class_toks:nnn { Boundary } {#1}

```

```

355     {
356       \bool_if:nTF
357       {
358         \l_xeCJK_xecglue_bool &&
359         \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
360         \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl
361       }
362       {
363         \tex_unskip:D
364         \bool_if:nTF
365         { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK_space } }
366         { \CJKecglue } { \c_space_token }
367       }
368       {
369         \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
370         { \xeCJK_if_last_node:nT { CJK_space } { \xeCJK_space_or_xecglue: } }
371       }
372     }
373     \str_if_eq_x:nnF {#1} { NormalSpace }
374     { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
375   }
376   \clist_map_inline:nn { Default, HalfRight, NormalSpace }
377   {
378     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
379     {
380       \peek_catcode:NTF \c_space_token
381       { { \xeCJK_make_node:n { default_space } } }
382       { { \xeCJK_make_node:n { default } } }
383     }
384     \str_if_eq_x:nnF {#1} { NormalSpace }
385     { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
386   }
387   \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }

```

\xeCJK_Boundary_and_CJK:

```

    \xeCJK_check_for_glue:
388   \cs_new_nopar:Nn \xeCJK_Boundary_and_CJK:
389   {
390     { \xeCJK_check_for_glue: }
391     \c_group_begin_token
392     \xeCJK_clear_CJK_toks:
393     \xeCJK_select_font:
394     \CJKsymbol
395   }
396   \cs_new_nopar:Nn \xeCJK_check_for_glue:
397   {
398     \bool_if:nTF
399     { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK_space } }
400     { \CJKglue }
401     {
402       \bool_if:nTF
403       {
404         \xeCJK_if_last_node_p:n { default }

```

```

405         \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
406         \xeCJK_if_last_node_p:n { default_itcorr }
407     }
408     { \CJKecglue }
409     {
410         \bool_if:nT
411         {
412             \l_xeCJK_xecglue_bool &&
413             \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
414             ( \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl ||
415               \dim_compare_p:nNn \tex_lastskip:D = \l_xeCJK_ecglue_dim )
416         }
417         {
418             \tex_unskip:D
419             \bool_if:nTF
420             {
421                 \xeCJK_if_last_node_p:n { default_space } ||
422                 \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
423                 \xeCJK_if_last_node_p:n { default } ||
424                 \xeCJK_if_last_node_p:n { default_itcorr }
425             }
426             { \CJKecglue }
427             {
428                 \bool_if:nTF
429                 {
430                     \xeCJK_if_last_node_p:n { CJK } ||
431                     \xeCJK_if_last_node_p:n { CJK_space }
432                 }
433                 { \bool_if:NTF \l_xeCJK_space_bool \c_space_token \CJKglue }
434                 { \c_space_token }
435             }
436         }
437     }
438 }
439 }
440 \tl_set:Nn \l_xeCJK_space_glue_tl
441 {
442     \tex_fontdimen:D \c_two \tex_font:D
443     \@plus \tex_fontdimen:D \c_three \tex_font:D
444     \@minus \tex_fontdimen:D \c_four \tex_font:D
445 }
446 (End definition for \xeCJK_Boundary_and_CJK: and \xeCJK_check_for_glue:)
447 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
448 {
449     \c_group_end_token
450     { \xeCJK_make_node:n { CJK } }
451     \xeCJK_ignorespaces:
452 }
453 \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }

```

\xeCJK_CJK_and_CJK:N

```

453 \cs_new_nopar:Nn \xeCJK_CJK_and_CJK:N { \CJKglue \CJKsymbol {#1} }

```

(End definition for \xeCJK_CJK_and_CJK:N)

```

454 \clist_map_inline:nn { FullLeft, FullRight }
455 {
456   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
457   {
458     \c_group_begin_token
459     \xeCJK_clear_CJK_toks:
460     \xeCJK_select_font:
461     \use:c { xeCJK_CJK_and_#1:N }
462   }
463   \xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
464   \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \c_group_end_token \tex_ignorespaces:D }
465 }
466 \clist_map_inline:Nn \g_xeCJK_base_class_clist
467 {
468   \bool_if:nF
469   { \str_if_eq_x_p:nn {#1} { FullLeft } || \str_if_eq_x_p:nn {#1} { FullRight } }
470   {
471     \xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \nobreak }
472     \xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \xeCJK_after_FullRight: }
473   }
474 }
475 \clist_map_inline:nn { CJK, FullLeft, FullRight }
476 {
477   \clist_map_inline:nn { FullLeft, FullRight }
478   { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
479 }

```

\xeCJK_punct_rule:n 用于抹去标点符号的左/右空白。

```

480 \cs_new_nopar:Nn \xeCJK_punct_rule:n
481 {
482   \tex_vrule:D
483   \@width \dim_eval:c { \l_xeCJK_punct_coors_tl/rule/#1 }
484   \@depth \c_zero_dim
485   \@height \c_zero_dim \scan_stop:
486 }

```

(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。

```

487 \cs_new_nopar:Nn \xeCJK_punct_glue:n
488 {
489   \skip_horizontal:n
490   {
491     \dim_eval:c { \l_xeCJK_punct_coors_tl/glue/#1 }
492     \@plus \dim_eval:n { ( \tl_use:c { \l_xeCJK_punct_coors_tl/glue/#1 } ) /4 }
493     \@minus \dim_eval:n { ( \tl_use:c { \l_xeCJK_punct_coors_tl/glue/#1 } ) /2 }
494   }
495 }

```

(End definition for \xeCJK_punct_glue:n)

\xeCJK_punct_kern:n 相邻两个标点之间的间距。

```
496 \cs_new_nopar:Nn \xeCJK_punct_kern:n
497 { \tex_kern:D \dim_eval:c { \l_xeCJK_punct_coortl/kern/#1 } }
(End definition for \xeCJK_punct_kern:n)
```

\dim_eval:c

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

\xeCJK_before_FullLeft:N

```
499 \cs_new_nopar:Nn \xeCJK_before_FullLeft:N
500 {
501   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
502   \xeCJK_punct_rule:n { l/#1 }
503   \CJKpunctsymbol {#1}
504 }
(End definition for \xeCJK_before_FullLeft:N)
```

\xeCJK_after_FullRight:

```
505 \cs_new_nopar:Nn \xeCJK_after_FullRight:
506 {
507   \xeCJK_get_punct_bounds:nV { r } \l_xeCJK_lastpunct_tl
508   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
509   \xeCJK_punct_glue:n { r/\l_xeCJK_lastpunct_tl }
510 }
(End definition for \xeCJK_after_FullRight:)
```

\xeCJK_CJK_and_FullLeft:N

```
511 \cs_new_nopar:Nn \xeCJK_CJK_and_FullLeft:N
512 {
513   \xeCJK_get_punct_bounds:nN { l } {#1}
514   \int_compare:nNf \etex_lastnodetype:D = \c_one
515     { \xeCJK_punct_glue:n { l/#1 } }
516   \xeCJK_before_FullLeft:N {#1}
517   \xeCJK_punct_if_middle:NT {#1}
518   {
519     \xeCJK_punct_rule:n { m/#1 }
520     \xeCJK_punct_glue:n { l/#1 }
521   }
522 }
(End definition for \xeCJK_CJK_and_FullLeft:N)
```

\xeCJK_CJK_and_FullRight:N

```
523 \cs_new_nopar:Nn \xeCJK_CJK_and_FullRight:N
524 {
525   \xeCJK_get_punct_bounds:nN { r } {#1}
526   \xeCJK_punct_if_long:NTF {#1} \CJKglue
527   {
528     \nobreak
529     \xeCJK_punct_if_middle:NT {#1}
530     {
531       \xeCJK_punct_glue:n { r/#1 }

```

```

532         \xeCJK_punct_rule:n { m/#1 }
533     }
534 }
535 \tl_gset:Nx \l_xeCJK_lastpunct_tl {#1}
536 \CJKpunctsymbol {#1}
537 }
(End definition for \xeCJK_CJK_and_FullRight:N)

```

\xeCJK_FullLeft_and_FullLeft:N

```

538 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullLeft:N
539 {
540     \nobreak
541     \xeCJK_get_punct_bounds:nN { l } {#1}
542     \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
543     \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
544     \xeCJK_before_FullLeft:N {#1}
545 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N)

```

\xeCJK_FullLeft_and_FullRight:N

```

546 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullRight:N
547 {
548     \nobreak
549     \xeCJK_get_punct_bounds:nN { r } {#1}
550     \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
551     \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
552     \nobreak
553     \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
554     \CJKpunctsymbol {#1}
555 }
(End definition for \xeCJK_FullLeft_and_FullRight:N)

```

\xeCJK_FullRight_and_FullLeft:N

```

556 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullLeft:N
557 {
558     \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
559     \xeCJK_get_punct_bounds:nN { l } {#1}
560     \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
561     \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
562     \xeCJK_punct_nobreak:
563     \xeCJK_before_FullLeft:N {#1}
564 }
(End definition for \xeCJK_FullRight_and_FullLeft:N)

```

\xeCJK_FullRight_and_FullRight:N

```

565 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullRight:N
566 {
567     \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
568     \xeCJK_get_punct_bounds:nN { r } {#1}
569     \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
570     \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
571     \nobreak

```

```

572 \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
573 \CJKpunctsymbol {#1}
574 }
(End definition for \xeCJK_FullRight_and_FullRight:N)

```

```

\xeCJK_if_last_node_p:n
\xeCJK_if_last_node:nTF 575 \prg_new_conditional:Nnn \xeCJK_if_last_node:n { p , T , F , TF }
576 {
577 \if_dim:w \dim_use:c { c_xeCJK_#1_node_dim } = \tex_lastkern:D
578 \prg_return_true: \else: \prg_return_false: \fi:
579 }
(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 580 \cs_new_nopar:Nn \xeCJK_def_node:nn
581 {
582 \dim_if_exist:cTF { c_xeCJK_#1_node_dim }
583 { \dim_gset:cn } { \dim_const:cn }
584 { c_xeCJK_#1_node_dim } {#2}
585 }
586 \cs_new_protected_nopar:Nn \xeCJK_make_node:n
587 {
588 \tex_kern:D - \dim_use:c { c_xeCJK_#1_node_dim }
589 \tex_kern:D \dim_use:c { c_xeCJK_#1_node_dim }
590 }
591 \xeCJK_def_node:nn { CJK } { 11 sp }
592 \xeCJK_def_node:nn { CJK_space } { 12 sp }
593 \xeCJK_def_node:nn { default } { 13 sp }
594 \xeCJK_def_node:nn { default_space } { 14 sp }
595 \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。

```

596 \keys_define:nn { xeCJK / options } { CJKglue .tl_set:N = \CJKglue }
(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

```

597 \keys_define:nn { xeCJK / options }
598 {
599 CJKecglue .code:n =
600 {
601 \cs_set:Npn \CJKecglue {#1}
602 \settowidth \l_xeCJK_ecglue_dim \CJKecglue
603 },
604 xCJKecglue .choice:,
605 xCJKecglue / true .code:n =
606 {
607 \bool_set_true:N \l_xeCJK_xecglue_bool
608 \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
609 },
610 xCJKecglue / false .code:n =
611 {

```

```

612     \bool_set_false:N \l_xeCJK_xecglue_bool
613     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_token
614   },
615   xCJKecglue / unknown .code:n =
616   {
617     \bool_set_true:N \l_xeCJK_xecglue_bool
618     \cs_set:Npn \CJKecglue {#1}
619     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
620     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
621   },
622   xCJKecglue .default:n = { true } ,
623 }
624 \dim_new:N \l_xeCJK_ecglue_dim
625 \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。

```

626 \keys_define:nn { xeCJK / options }
627 {
628   CJKspace .choice:,
629   CJKspace / true .code:n = { \CJKspace },
630   CJKspace / false .code:n = { \CJKnospace },
631   CJKspace .default:n = { true },
632   space .meta:n = { CJKspace = true },
633   nospace .meta:n = { CJKspace = false },
634 }
(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
635 \NewDocumentCommand \CJKspace { }
636 {
637   \bool_set_true:N \l_xeCJK_space_bool
638   \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_peek_math:
639 }
640 \NewDocumentCommand \CJKnospace { }
641 {
642   \bool_set_false:N \l_xeCJK_space_bool
643   \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_ignore_spaces:
644 }
(End definition for \CJKspace and \CJKnospace)

```

\xeCJK_ignore_spaces:

```

\xeCJK_peek_math:
645 \cs_new_nopar:Nn \xeCJK_ignore_spaces:
646 {
647   \peek_catcode:NTF \c_space_token
648   {
649     \xeCJK_if_last_node:nT { CJK }
650     { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK_space } } }
651     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
652     { \xeCJK_space_or_xecglue: }
653     {
654       \bool_if:nT

```

```

655         {
656             \xeCJK_if_outter_macro_p:c { l_peek_token } ||
657             \token_if_cs_p:c { l_peek_token }
658         }
659         { \xeCJK_space_or_xecglue: }
660     }
661 }
662 {
663     \xeCJK_if_outter_macro:cF { l_peek_token }
664     {
665         \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
666         { \cs_if_exist:cF { l_peek_token } \exp_not:N }
667     }
668 }
669 }
670 \cs_new_nopar:Nn \xeCJK_peek_math:
671 {
672     \peek_catcode:NTF \c_space_token
673     {
674         \xeCJK_if_last_node:nT { CJK }
675         { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK_space } } }
676         \xeCJK_peek_catcode_ignore_spaces:NF \c_space_token
677         { \xeCJK_space_or_xecglue: }
678     }
679     {
680         \xeCJK_if_outter_macro:cF { l_peek_token }
681         {
682             \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
683             { \cs_if_exist:cF { l_peek_token } \exp_not:N }
684         }
685     }
686 }
(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。

```

687 \keys_define:nn { xeCJK / options }
688 {
689     CheckSingle .choice:,
690     CheckSingle / true .code:n = { \xeCJKenablechecksingle },
691     CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
692     CheckSingle .default:n = { true },
693     CJKchecksingle .meta:n = { CheckSingle = true } ,
694 }
(End definition for CheckSingle This function is documented on page 4.)

```

`\xeCJKenablechecksingle` 孤字处理功能开关。

```

\xeCJKdisablechecksingle
695 \bool_new:N \l_xeCJK_checksingle_bool
696 \NewDocumentCommand \xeCJKenablechecksingle { }
697 {
698     \bool_if:NF \l_xeCJK_checksingle_bool

```

```

699     {
700         \bool_set_true:N \l_xeCJK_checksingle_bool
701         \cs_set_eq:NN \xeCJK_checksingle_save_CJKsymbol:N \xeCJK_CJK_and_CJK:N
702         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle:N
703     }
704 }
705 \NewDocumentCommand \xeCJKdisablechecksingles { }
706 {
707     \bool_if:NT \l_xeCJK_checksingle_bool
708     {
709         \bool_set_false:N \l_xeCJK_checksingle_bool
710         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle_save_CJKsymbol:N
711     }
712 }

```

(End definition for \xeCJKenablechecksingles and \xeCJKdisablechecksingles)

\xeCJK_checksingles:N

\xeCJK_checksingles:NN

\xeCJK_checksingles:NNN

```

713 \cs_new_nopar:Nn \xeCJK_checksingles:N
714 {
715     \peek_catcode:NTF \c_catcode_other_token
716     { \xeCJK_checksingles:NN {#1} }
717     {
718         \bool_if:nTF
719         {
720             ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } )      &&
721             \tl_if_empty_p:x { \token_get_arg_spec:c { l_peek_token } } &&
722             \exp_args:NNc \exp_args:No \tl_if_single_token_p:n { l_peek_token } &&
723             \exp_args:NNc \exp_after:wN \token_if_other_p:N { l_peek_token }
724         }
725         { \xeCJK_checksingles:NN {#1} }
726         { \xeCJK_checksingles_save_CJKsymbol:N {#1} }
727     }
728 }
729 \cs_new_nopar:Nn \xeCJK_checksingles:NN
730 {
731     \peek_catcode:NTF \c_catcode_other_token
732     { \xeCJK_checksingles_save_CJKsymbol:N {#1} #2 }
733     {
734         \bool_if:nTF
735         {
736             ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
737             ( \token_if_space_p:c { l_peek_token } ||
738               \token_if_math_toggle_p:c { l_peek_token } ||
739               \token_if_cs_p:c { l_peek_token } )
740         }
741         {
742             \bool_if:nTF { \token_if_space_p:c { l_peek_token } }
743             {
744                 \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_other_token
745                 { \xeCJK_checksingles_save_CJKsymbol:N {#1} #2 \c_space_token }
746                 {
747                     \bool_if:nTF

```

```

748         {
749             ! ( \xCJK_if_outer_macro_p:c { l_peek_token } ) &&
750             ( \token_if_math_toggle_p:c { l_peek_token } ||
751               \token_if_cs_p:c { l_peek_token } )
752         }
753         { \xCJK_checksingl:e:NNN {#1} {#2} \c_space_token }
754         { \xCJK_checksingl:e_save_CJKsymbol:N {#1} #2 \c_space_token }
755     }
756 }
757 { \xCJK_checksingl:e:NNN {#1} {#2} { } }
758 }
759 { \xCJK_checksingl:e_save_CJKsymbol:N {#1} #2 }
760 }
761 }
762 \cs_new_nopar:Nn \xCJK_checksingl:e:NNN
763 {
764     \bool_if:nTF
765     {
766         \token_if_eq_meaning_p:NN \l_peek_token \par ||
767         \token_if_eq_meaning_p:NN \l_peek_token \[ % \]
768     }
769     { \CJKsymbol {#1} \nobreak #2 }
770     { \xCJK_checksingl:e_env:NNN {#1} {#2} {#3} }
771 }

```

(End definition for \xCJK_checksingl:e:N, \xCJK_checksingl:e:NN, and \xCJK_checksingl:e:NNN)

\xCJK_checksingl:e_env:NNN
 \xCJK_checksingl:e_env:NNNNn

```

772 \cs_new_nopar:Nn \xCJK_checksingl:e_env:NNN
773 {
774     \peek_catcode_remove:NTF \c_math_toggle_token
775     {
776         \peek_catcode:NTF \c_math_toggle_token
777         { \CJKsymbol {#1} \nobreak #2 \c_math_toggle_token }
778         { \xCJK_checksingl:e_save_CJKsymbol:N {#1} #2#3 \c_math_toggle_token }
779     }
780     {
781         \peek_meaning_remove:NTF \begin
782         { \xCJK_checksingl:e_env:NNNNn {#1} {#2} {#3} \begin }
783         {
784             \peek_meaning_remove:NTF \end
785             { \xCJK_checksingl:e_env:NNNNn {#1} {#2} {#3} \end }
786             { \xCJK_checksingl:e_save_CJKsymbol:N {#1} #2#3 }
787         }
788     }
789 }
790 \cs_new_nopar:Nn \xCJK_checksingl:e_env:NNNNn
791 {
792     \clist_if_in:NnTF \g_xeCJK_inline_env_clist {#5}
793     { \xCJK_checksingl:e_save_CJKsymbol:N {#1} #2#3 }
794     { \CJKsymbol {#1} \nobreak #2 }
795     \scan_stop: #4 {#5}
796 }

```

(End definition for \xeCJK_checksingle_env:NNN and \xeCJK_checksingle_env:NNNNn)

```
InlineEnv
InlineEnv+ 797 \keys_define:nn { xecjk / options }
InlineEnv- 798 {
799   InlineEnv .clist_gset:N = \g_xecjk_inline_env_clist ,
800   InlineEnv+ .code:n =
801   {
802     \clist_gput_right:Nx \g_xecjk_inline_env_clist {#1}
803     \clist_gremove_duplicates:N \g_xecjk_inline_env_clist
804   },
805   InlineEnv- .code:n =
806   {
807     \clist_map_inline:xn {#1}
808     { \clist_gremove_all:Nn \g_xecjk_inline_env_clist {##1} }
809   },
810 }
```

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

```
811 \cs_generate_variant:Nn \token_if_cs_p:N { c }
812 \cs_generate_variant:Nn \token_if_space_p:N { c }
813 \cs_generate_variant:Nn \token_get_arg_spec:N { c }
814 \cs_generate_variant:Nn \token_if_math_toggle_p:N { c }
815 \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 816 \tl_new:N \xeCJK_UL_subclass_patch_tl
817 \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: 818 \cs_new_nopar:Nn \xeCJK_erase_CJKsymbol:
819 {
820   \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
821   \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
822 }
823 \cs_new_nopar:Nn \xeCJK_restore_CJKsymbol:
824 {
825   \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block
826   \CJKsymbol
827 }
(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。

```
828 \NewDocumentCommand \xeCJKDeclareSubCJKBlock { m m }
829 { \xeCJKDeclareSubCharClass { CJK } {#1} {#2} }
830 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock This function is documented on page 9.)
```

`\xeCJKCancelSubCJKBlock`
`\xeCJKRestoreSubCJKBlock`

取消和恢复对 CJK 子区的声明。

```

831 \bool_new:N \l_xeCJK_sub_cancel_bool
832 \NewDocumentCommand \xeCJKCancelSubCJKBlock { m }
833 {
834   \bool_if:NF \l_xeCJK_sub_cancel_bool
835   {
836     \bool_set_true:N \l_xeCJK_sub_cancel_bool
837     \xeCJK_sub_restore_or_cancel:n {#1}
838   }
839 }
840 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { m }
841 {
842   \bool_if:NT \l_xeCJK_sub_cancel_bool
843   {
844     \bool_set_false:N \l_xeCJK_sub_cancel_bool
845     \xeCJK_sub_restore_or_cancel:n {#1}
846   }
847 }
848 \cs_new_nopar:Nn \xeCJK_sub_restore_or_cancel:n
849 {
850   \clist_map_inline:xn {#1}
851   {
852     \cs_if_free:cTF { xeCJK_ CJK/##1 _class }
853     { \xeCJK_error:nx { SubBlock-undefined } {##1} }
854     {
855       \xeCJKDeclareCharClass
856       { CJK \bool_if:NF \l_xeCJK_sub_cancel_bool { /##1 } }
857       { \tl_use:c { g_xeCJK_CJK/##1_range_clist } }
858     }
859   }
860 }
861 \xeCJK_msg_new:nn { SubBlock-undefined }
862 {
863   The~CJK~sub~block~'#1'~is~undefined.\\\
864   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \c_space_tl
865   to~declare~it.
866 }
```

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

`\xeCJKDeclareSubCharClass`

```

867 \NewDocumentCommand \xeCJKDeclareSubCharClass
868 { > { \TrimSpaces } m > { \TrimSpaces } m m }
869 {
870   \cs_if_free:cT { xeCJK_ #1/#2 _class }
871   {
872     \xeCJK_new_class:n { #1/#2 }
873     \xeCJK_set_sub_class_toks:nn {#1} {#2}
874     \xeCJK_new_sub_key:n {#2}
875     \xeCJK_UL_subclass_patch:nn {#1} {#2}
876   }
877   \xeCJKDeclareCharClass { #1/#2 } {#3}
```

```

878 }
879 \@onlypreamble \xeCJKDeclareSubCharClass
(End definition for \xeCJKDeclareSubCharClass)

```

\xeCJK_set_sub_class_toks:nn

```

880 \cs_new_nopar:Nn \xeCJK_set_sub_class_toks:nn
881 {
882   \clist_map_inline:Nn \g_xeCJK_base_class_clist
883   {
884     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
885     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
886     \str_if_eq_x:nnTF {##1} { CJK }
887     {
888       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
889       { \xeCJK_switch_font:nn {#1} {#2} }
890     }
891     {
892       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
893       { \xeCJK_erase_CJKsymbol: }
894       \xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
895       { \xeCJK_switch_font:nn {#1} {#2} \xeCJK_restore_CJKsymbol: }
896     }
897   }
898   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
899   \clist_if_empty:NF \g_xeCJK_CJK_subclass_clist
900   {
901     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
902     {
903       \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
904       \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
905       \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
906       { \xeCJK_switch_font:nn {#2} {##1} }
907       \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
908       { \xeCJK_switch_font:nn {##1} {#2} }
909     }
910   }
911   \clist_gput_right:Nn \g_xeCJK_CJK_subclass_clist {#2}
912   \xeCJK_save_CJK_class:n { #1/#2 }
913   \clist_map_inline:nn { CJK, FullLeft, FullRight }
914   {
915     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
916     { \xeCJK_switch_font:nn {#2} {#1} }
917   }
918 }
(End definition for \xeCJK_set_sub_class_toks:nn)

```

\xeCJK_UL_subclass_patch:nn

```

919 \cs_new_nopar:Nn \xeCJK_UL_subclass_patch:nn
920 {
921   \tl_put_right:Nn \xeCJK_UL_subclass_patch_tl
922   {
923     \clist_map_inline:nn { Default, HalfLeft, HalfRight }

```

```

924     {
925       \str_if_eq_x:nnTF {##1} { HalfLeft }
926       { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKecglue \CJKsymbol } }
927       { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
928       \xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
929       {
930         \xeCJK_erase_CJKsymbol:
931         \xeCJK_Boundary_and_CJK:
932         \xeCJK_switch_font:nn {#1} {#2}
933         \xeCJK_restore_CJKsymbol:
934       }
935     }
936   }
937 }
(End definition for \xeCJK_UL_subclass_patch:nn)

```

5.8 标点处理

`\g_xeCJK_punctstyle_clist` 标点处理方式。

```

938 \clist_new:N \g_xeCJK_punctstyle_clist
939 \clist_set:Nn \g_xeCJK_punctstyle_clist
940 { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
941 \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
942 { \tl_const:cn { c_xeCJK_ps_#1_tl } {#1} }
(End definition for \g_xeCJK_punctstyle_clist)

```

`\keys_define:nx`

```

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

```

PunctStyle

```

944 \keys_define:nn { xeCJK / options }
945 {
946   PunctStyle .choice_code:n =
947   {
948     \tl_set:Nx \l_xeCJK_punctstyle_tl { \l_keys_choice_tl }
949     \tl_if_eq:NNT \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
950     { \xeCJKallowbreakbetweenpuncts }
951   },
952 }
953 \keys_define:nx { xeCJK / options }
954 { PunctStyle .generate_choices:n = { \exp_not:V \g_xeCJK_punctstyle_clist } }
955 \keys_define:nn { xeCJK / options }
956 {
957   PunctStyle / banjiao .meta:n = { PunctStyle = halfwidth },
958   PunctStyle / quanjiao .meta:n = { PunctStyle = fullwidth },
959   PunctStyle / kaiming .meta:n = { PunctStyle = mixedwidth },
960   PunctStyle / hangmobanjiao .meta:n = { PunctStyle = marginkerning },
961   PunctStyle / unknown .code:n =
962   { \xeCJK_error:nx { PunctStyle-undefined } \l_keys_value_tl },
963 }
964 \xeCJK_msg_new:nn { PunctStyle-undefined }

```

```

965 {
966   Punctstyle~"#1"~is~not~available. \\\
967   You~can~only~use~one~of\\\
968   "\g_xeCJK_punctstyle_clist".\\
969 }

```

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

AllowBreakBetweenPuncts 相关选项声明。

```

KaiMingPunct 970 \keys_define:nn { xeCJK / options }
LongPunct    971 {
MiddlePunct  972   AllowBreakBetweenPuncts .choice:,
PunctWidth   973   AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
974   AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
975   AllowBreakBetweenPuncts .default:n = { true },
976   KaiMingPunct .code:n = { \xeCJK_set_special_punct:nn { mixedwidth } {#1} },
977   KaiMingPunct+ .code:n = { \xeCJK_add_special_punct:nn { mixedwidth } {#1} },
978   KaiMingPunct- .code:n = { \xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
979   LongPunct .code:n = { \xeCJK_set_special_punct:nn { long } {#1} },
980   LongPunct+ .code:n = { \xeCJK_add_special_punct:nn { long } {#1} },
981   LongPunct- .code:n = { \xeCJK_sub_special_punct:nn { long } {#1} },
982   MiddlePunct+ .code:n = { \xeCJK_add_special_punct:nn { middle } {#1} },
983   MiddlePunct .code:n = { \xeCJK_set_special_punct:nn { middle } {#1} },
984   MiddlePunct- .code:n = { \xeCJK_sub_special_punct:nn { middle } {#1} },
985   PunctWidth .tl_gset:N = \g_xeCJK_punct_width_tl ,
986 }

```

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

```

\g_xeCJK_allowbreakbetweenpuncts
\g_xeCJK_nobreakbetweenpuncts 987 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
988 { \cs_set_eq:NN \xeCJK_punct_nobreak: \xeCJK_zero_glue: }
989 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
990 { \cs_set_eq:NN \xeCJK_punct_nobreak: \nobreak }
991 \cs_new_protected_nopar:Nn \xeCJK_zero_glue: { \skip_horizontal:N \c_zero_skip }

```

(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts)

```

\g_xeCJK_special_ps_clist
\g_xeCJK_set_special_punct:nn 992 \clist_set:Nn \g_xeCJK_special_ps_clist { mixedwidth , long , middle }
\g_xeCJK_add_special_punct:nn 993 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
\g_xeCJK_sub_special_punct:nn 994 {
995   \tl_new:c { l_xeCJK_#1_punct_tl }
996   \prop_new:c { l_xeCJK_#1_punct_prop }
997 }
998 \cs_new_nopar:Nn \xeCJK_set_special_punct:nn
999 {
1000   \tl_set:cx { l_xeCJK_#1_punct_tl } {#2}
1001   \prop_clear:c { l_xeCJK_#1_punct_prop }
1002   \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
1003     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
1004 }
1005 \cs_new_nopar:Nn \xeCJK_add_special_punct:nn
1006 {
1007   \tl_put_right:cx { l_xeCJK_#1_punct_tl } {#2}

```

```

1008     \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
1009     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
1010 }
1011 \cs_new_nopar:Nn \xeCJK_sub_special_punct:nn
1012 {
1013     \tl_map_inline:xn {#2}
1014     {
1015         \tl_remove_all:cn { l_xeCJK_#1_punct_tl } {##1}
1016         \prop_remove:cn { l_xeCJK_#1_punct_prop } {##1}
1017     }
1018 }
1019 \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 1020 \cs_generate_variant:Nn \dim_set:Nn { Nc }
\dim_set_min:Nc 1021 \cs_generate_variant:Nn \dim_add:Nn { Nc }
1022 \cs_generate_variant:Nn \dim_set_min:Nn { Nc }
(End definition for \dim_set:Nc, \dim_add:Nc, and \dim_set_min:Nc)

```

`\l_xeCJK_punct_coor_tl` 记录当前字体状态下的标点处理方式。

```

1023 \tl_set:Nn \l_xeCJK_punct_coor_tl { \l_xeCJK_font_coor_tl/\l_xeCJK_punctstyle_tl }
(End definition for \l_xeCJK_punct_coor_tl)

```

`\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。

```

1024 \cs_new_nopar:Nn \xeCJK_get_punct_bounds:nN
1025 {
1026     \tl_if_exist:cF { \l_xeCJK_punct_coor_tl/rule/#1/#2 }
1027     {
1028         \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1029         {
1030             \clist_map_inline:nn { glue , rule , bound }
1031             { \tl_gset:cV { \l_xeCJK_punct_coor_tl/##1/#1/#2 } \c_zero_dim }
1032             \tl_gset:cV { \l_xeCJK_punct_coor_tl/rule/m/#2 } \c_zero_dim
1033         }
1034         {
1035             \tl_if_exist:cF { \l_xeCJK_punct_coor_tl/bound/#1/#2 }
1036             { { \xeCJK_select_font: \xeCJK_get_punct_dimen:N {#2} } }
1037             \xeCJK_punct_if_long:NTF {#2}
1038             {
1039                 \dim_zero:N \l_xeCJK_tmpa_dim
1040                 \dim_zero:N \l_xeCJK_tmpp_dim
1041                 \dim_zero:N \l_xeCJK_tmppc_dim
1042             }
1043             {
1044                 \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/#1/#2 }
1045                 \dim_set:Nc \l_xeCJK_tmpp_dim
1046                 { \l_xeCJK_punct_coor_tl/bound/\str_if_eq_x:nnTF {#1} lrl/#2 }
1047                 \dim_set:Nn \l_xeCJK_tmppc_dim
1048                 {

```

```

1049 \cs_if_exist_use:cF { g_xeCJK_punct_width/#2 }
1050 {
1051     \xeCJK_tl_if_blank:xTF \g_xeCJK_punct_width_tl
1052     { \c_zero_dim }
1053     { \g_xeCJK_punct_width_tl }
1054 }
1055 }
1056 \xeCJK_punct_if_middle:NTF {#2}
1057 {
1058     \dim_set:Nn \l_xeCJK_tmpc_dim
1059     {
1060         \dim_compare:nNnTF \l_xeCJK_tmpc_dim > \c_zero_dim
1061         { \l_xeCJK_tmpc_dim }
1062         {
1063             \tl_case:Nnn \l_xeCJK_punctstyle_tl
1064             {
1065                 \c_xeCJK_ps_halfwidth_tl { .5 em }
1066                 \c_xeCJK_ps_mixedwidth_tl { .5 em }
1067                 \c_xeCJK_ps_CCT_tl { .7 em }
1068                 \c_xeCJK_ps_fullwidth_tl { 1 em }
1069             }
1070             { 1 em }
1071         }
1072         - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/#2 }
1073     }
1074     \dim_set:Nn \l_xeCJK_tmpc_dim { .5 \l_xeCJK_tmpc_dim }
1075 }
1076 {
1077     \dim_compare:nNnTF \l_xeCJK_tmpc_dim > \c_zero_dim
1078     {
1079         \dim_set:Nn \l_xeCJK_tmpc_dim
1080         {
1081             \l_xeCJK_tmpc_dim - \l_xeCJK_tmpe_dim
1082             - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/#2 }
1083         }
1084     }
1085     {
1086         \dim_set_eq:NN \l_xeCJK_tmpc_dim \l_xeCJK_tmpe_dim
1087         \dim_set_min:Nn \l_xeCJK_tmpc_dim \l_xeCJK_tmpe_dim
1088         \tl_case:Nnn \l_xeCJK_punctstyle_tl
1089         {
1090             \c_xeCJK_ps_halfwidth_tl { \prg_do_nothing: }
1091             \c_xeCJK_ps_mixedwidth_tl
1092             {
1093                 \xeCJK_punct_if_mixedwidth:NT {#2}
1094                 {
1095                     \dim_add:Nn \l_xeCJK_tmpc_dim { .5 \l_xeCJK_tmpe_dim }
1096                     \dim_set_min:Nn \l_xeCJK_tmpc_dim \l_xeCJK_tmpe_dim
1097                 }
1098             }
1099             \c_xeCJK_ps_CCT_tl
1100             {

```

```

1101         \xeCJK_punct_if_mixedwidth:NTF {#2}
1102         { \dim_add:Nn \l_xeCJK_tmpc_dim { .5 \l_xeCJK_tmpa_dim } }
1103         { \dim_add:Nn \l_xeCJK_tmpc_dim { .3 \l_xeCJK_tmpa_dim } }
1104         \dim_set_min:Nn \l_xeCJK_tmpc_dim \l_xeCJK_tmpa_dim
1105     }
1106 }
1107 { \dim_set_eq:NN \l_xeCJK_tmpc_dim \l_xeCJK_tmpa_dim }
1108 }
1109 }
1110 \dim_set_max:Nn \l_xeCJK_tmpc_dim \c_zero_dim
1111 }
1112 \tl_gset:cx { \l_xeCJK_punct_coord_tl/rule/#1/#2 }
1113 { - \dim_use:N \l_xeCJK_tmpa_dim }
1114 \tl_gset:cx { \l_xeCJK_punct_coord_tl/rule/m/#2 }
1115 { - \dim_use:N \l_xeCJK_tmpb_dim }
1116 \tl_gset:cV { \l_xeCJK_punct_coord_tl/glue/#1/#2 } \l_xeCJK_tmpc_dim
1117 }
1118 }
1119 }
1120 \cs_generate_variant:Nn \xeCJK_get_punct_bounds:nN { nV }
(End definition for \xeCJK_get_punct_bounds:nN)

```

\xeCJKsetwidth

```

1121 \NewDocumentCommand \xeCJKsetwidth { m m }
1122 { \tl_map_inline:xn {#1} { \tl_gset:cn { g_xeCJK_punct_width/##1 } {#2} } }
(End definition for \xeCJKsetwidth This function is documented on page 10.)

```

```

\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
1123 \prg_new_conditional:Nnn \xeCJK_punct_if_right:N { p , T , F , TF }
1124 {
1125     \tl_if_exist:cTF { \l_xeCJK_punct_coord_tl/glue/r/#1 }
1126     \prg_return_true: \prg_return_false:
1127 }
1128 \prg_new_conditional:Nnn \xeCJK_punct_if_both_left_right:NN { p , T , F , TF }
1129 {
1130     \bool_if:nTF
1131     {
1132         ( \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/r/#1 } &&
1133         \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/r/#2 } ) ||
1134         ( \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/l/#1 } &&
1135         \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/l/#2 } )
1136     }
1137     \prg_return_true: \prg_return_false:
1138 }
1139 \cs_generate_variant:Nn \prg_new_conditional:Nnn { c }
1140 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
1141 {
1142     \prg_new_conditional:cnn { xeCJK_punct_if_#1:N } { p , T , F , TF }
1143     {
1144         \prop_if_in:cnTF { l_xeCJK_#1_punct_prop } {##1}
1145         \prg_return_true: \prg_return_false:
1146     }
1147 }

```

(End definition for \xeCJK_punct_if_right:N and others.)

\xeCJK_get_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。

```
1148 \cs_new_nopar:Nn \xeCJK_get_kern:NN
1149 {
1150   \tl_if_exist:cF { \l_xeCJK_punct_coor_tl/kern/#1-#2 }
1151   {
1152     \dim_zero:N \l_xeCJK_tmpa_dim
1153     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1154     {
1155       \tl_if_exist:cTF { g_xeCJK_punct/kern/#1-#2 }
1156       { \dim_set:Nc \l_xeCJK_tmpa_dim { g_xeCJK_punct/kern/#1-#2 } }
1157       { \xeCJK_calc_kern:NN {#1} {#2} }
1158     }
1159     \tl_gset:cV { \l_xeCJK_punct_coor_tl/kern/#1-#2 } \l_xeCJK_tmpa_dim
1160   }
1161 }
1162 \cs_new_nopar:Nn \xeCJK_calc_kern:NN
1163 {
1164   \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_marginkerning_tl
1165   {
1166     \xeCJK_punct_if_right:NT {#1}
1167     { \dim_add:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/r/#1 } }
1168     \xeCJK_punct_if_right:NF {#2}
1169     { \dim_add:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/l/#2 } }
1170   }
1171   {
1172     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_fullwidth_tl
1173     { \xeCJK_calc_kern:NnNN \l_xeCJK_tmpa_dim { 1.5 em } {#1} {#2} }
1174     {
1175       \xeCJK_punct_if_both_left_right:NNTF {#1} {#2}
1176       {
1177         \xeCJK_punct_if_mixedwidth:NT {#1}
1178         {
1179           \xeCJK_calc_kern:NnNN \l_xeCJK_tmpa_dim { 1 em } {#1} {#2}
1180           \dim_compare:nNNT \l_xeCJK_tmpa_dim < { .1 em }
1181           { \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/l/#1 } }
1182         }
1183       }
1184       {
1185         \bool_if:nTF
1186         {
1187           \xeCJK_punct_if_mixedwidth_p:N {#1} &&
1188           ! ( \tl_if_eq_p:NN \l_xeCJK_punctstyle_tl \c_xeCJK_ps_halfwidth_tl )
1189         }
1190         {
1191           \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coor_tl/glue/r/#1 }
1192           \dim_set:Nn \l_xeCJK_tmpa_dim { .7 \l_xeCJK_tmpa_dim }
1193         }
1194         {
1195           \xeCJK_calc_kern:NnNN \l_xeCJK_tmpa_dim { 1 em } {#1} {#2}

```

```

1196         \dim_compare:nNnT \l_xeCJK_tmpa_dim < { .1 em }
1197         {
1198         \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#1 }
1199         \dim_set_min:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/r/#1 }
1200         \dim_set:Nc \l_xeCJK_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/l/#2 }
1201         \dim_set_min:Nc \l_xeCJK_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/r/#2 }
1202         \dim_set_max:Nn \l_xeCJK_tmpa_dim \l_xeCJK_tmpb_dim
1203         }
1204     }
1205 }
1206 }
1207 \bool_if:nT
1208 { \xeCJK_punct_if_long_p:N {#1} || \xeCJK_punct_if_long_p:N {#2} }
1209 { \dim_set_max:Nn \l_xeCJK_tmpa_dim { .1 em } }
1210 }
1211 \dim_set_max:Nn \l_xeCJK_tmpa_dim \c_zero_dim
1212 }
1213 \cs_generate_variant:Nn \xeCJK_get_kern:NN { V }
(End definition for \xeCJK_get_kern:NN and \xeCJK_calc_kern:NN)

```

`\xeCJK_calc_kern:NnNN` #3 和 #4 为相邻的两个标点,#2 为要确定的相邻两个标点总共占的宽度,#1 是尺寸寄存器,用于记录根据所占宽度计算得出的相邻两个标点间距。

```

1214 \cs_new_nopar:Nn \xeCJK_calc_kern:NnNN
1215 {
1216     \dim_set:Nn #1
1217     {
1218         #2
1219         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1220             \xeCJK_punct_if_right:NTF {#3} { bound } { glue } /l/#3 }
1221         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1222             \xeCJK_punct_if_right:NTF {#4} { glue } { bound } /r/#4 }
1223         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#3 }
1224         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#4 }
1225         \xeCJK_punct_if_both_left_right:NNT {#3} {#4}
1226         { + \tl_use:c { \l_xeCJK_punct_coord_tl/rule/m/#4 } }
1227     }
1228 }
(End definition for \xeCJK_calc_kern:NnNN)

```

`\xeCJK_get_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。

```

1229 \cs_new_nopar:Nn \xeCJK_get_punct_dimen:N
1230 {
1231     \xeCJK_get_glyph_bounds:nNN { left } {#1} \l_xeCJK_tmpa_dim
1232     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/l/#1 } \l_xeCJK_tmpa_dim
1233     \dim_set_eq:NN \l_xeCJK_tmpb_dim \l_xeCJK_tmpa_dim
1234     \xeCJK_get_glyph_bounds:nNN { right } {#1} \l_xeCJK_tmpa_dim
1235     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/r/#1 } \l_xeCJK_tmpa_dim
1236     \dim_add:Nn \l_xeCJK_tmpb_dim \l_xeCJK_tmpa_dim
1237     \tl_gset:cx { \l_xeCJK_punct_coord_tl/dimen/#1 }
1238     { \dim_eval:n { \etex_fontcharwd:D \tex_font:D `#1 - \l_xeCJK_tmpb_dim } }
1239     \bool_if:nT

```

```

1240 {
1241   \xeCJK_punct_if_long_p:N {#1} &&
1242   ! ( \str_if_eq_x_p:nn {#1} { ... } || \str_if_eq_x_p:nn {#1} { .. } )
1243 }
1244 {
1245   \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
1246   {
1247     \str_if_eq_x:nnF \c_xeCJK_ps_plain_tl {##1}
1248     {
1249       \tl_gset:cx { \l_xeCJK_font_coors_tl/##1/kern/#1-#1 }
1250       { - \dim_use:N \l_xeCJK_tmpb_dim }
1251     }
1252   }
1253 }
1254 }

```

(End definition for \xeCJK_get_punct_dimen:N)

`\xeCJK_get_glyph_bounds:nNN` 用 `\XeTeXglyphbounds` 计算标点的上下左右空白。

```

1255 \cs_new_nopar:Nn \xeCJK_get_glyph_bounds:nNN
1256 {
1257   \dim_set:Nn #3
1258   {
1259     \XeTeXglyphbounds
1260     \str_case_x:nnn {#1}
1261     {
1262       { left } { \c_one } { top } { \c_two }
1263       { right } { \c_three } { bottom } { \c_four }
1264     }
1265     { \c_three }
1266     \int_eval:n { \XeTeXcharglyph `#2 }
1267   }
1268 }

```

(End definition for \xeCJK_get_glyph_bounds:nNN)

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

```

1269 \NewDocumentCommand \xeCJKsetkern { m m m }
1270 { \tl_gset:cn { g_xeCJK_punct/kern/#1-#2 } {#3} }

```

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

5.9 后备字体

AutoFallBack 后备字体的宏包选项声明。

```

1271 \keys_define:nn { xeCJK / options }
1272 {
1273   AutoFallBack .choice:,
1274   AutoFallBack / true .code:n = { \xeCJKenablefallback },
1275   AutoFallBack / false .code:n = { \xeCJKdisablefallback },
1276   AutoFallBack .default:n = { true },
1277   fallback .meta:n = { AutoFallBack = true },
1278 }

```

(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 1279 \bool_new:N \l_xeCJK_fallback_bool
1280 \NewDocumentCommand \xeCJKenablefallback { }
1281 {
1282   \bool_if:NF \l_xeCJK_fallback_bool
1283   {
1284     \bool_set_true:N \l_xeCJK_fallback_bool
1285     \cs_set_eq:NN \xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1286     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_testsymbol:N
1287   }
1288 }
1289 \NewDocumentCommand \xeCJKdisablefallback { }
1290 {
1291   \bool_if:NT \l_xeCJK_fallback_bool
1292   {
1293     \bool_set_false:N \l_xeCJK_fallback_bool
1294     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_save_CJKsymbol:N
1295   }
1296 }
```

(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。

```
1297 \cs_new_nopar:Nn \xeCJK_fallback_testsymbol:N
1298 {
1299   \font_glyph_if_exist:NnTF \tex_font:D {`#1}
1300   { \xeCJK_fallback_save_CJKsymbol:N {#1} }
1301   {
1302     \xeCJK_family_if_exist:xTF { \xeCJK@family/FallBack }
1303     { {
1304       \tl_put_right:Nn \xeCJK@family { /FallBack }
1305       \xeCJK_select_font:
1306       \xeCJK_fallback_testsymbol:N {#1}
1307     } }
1308     {
1309       \xeCJK_warning:nx { fallback } {#1}
1310       \xeCJK_fallback_save_CJKsymbol:N {#1}
1311     }
1312   }
1313 }
1314 \xeCJK_msg_new:nn { fallback }
1315 {
1316   CJKfamily~'\xeCJK@family'~
1317   ( \prop_get:NV \g_xeCJK_family_prop \xeCJK@family )~
1318   does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {`#1}).\\
1319 }
```

(End definition for \xeCJK_fallback_testsymbol:N)

`\setCJKfallbackfamilyfont`

```
\xeCJK_set_family_fallback:nnn 1320 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{ } m }
1321 {
1322   \exp_args:Nx \tl_if_in:nnTF {#3} { , }
1323   { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} }
```

```

1324     { \xeCJK_set_family:nnn { #1/FallBack } {#2} {#3} }
1325   }
1326   \cs_new_nopar:Nn \xeCJK_set_family_fallback:nnn
1327   {
1328     \group_begin:
1329     \tl_set:Nx \l_xeCJK_family_fb_tl {#1}
1330     \prop_get:NVNF \g_xeCJK_family_prop \l_xeCJK_family_fb_tl \l_xeCJK_fontname_tl
1331     { \tl_clear:N \l_xeCJK_fontname_tl }
1332     \clist_map_inline:xn {#3}
1333     {
1334       \tl_put_right:Nn \l_xeCJK_family_fb_tl { /FallBack }
1335       \xeCJK_get_sub_features:nn \l_xeCJK_family_fb_tl {##1}
1336       \clist_put_left:cx { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist } {#2}
1337       \xeCJK_set_family:ncc \l_xeCJK_family_fb_tl
1338       { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist }
1339       { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontname_tl }
1340     }
1341     \group_end:
1342   }

```

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

5.10 CJK 字体族声明方式

```

1343 \bool_new:N \g_xeCJK_AutoFakeBold_bool
1344 \bool_new:N \g_xeCJK_AutoFakeSlant_bool
1345 \fp_new:N \g_xeCJK_EmboldenFactor_fp
1346 \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

```

1347 \keys_define:nn { xeCJK / options }
1348 {
1349   AutoFakeBold .choice:,
1350   AutoFakeBold / true .code:n =
1351   { \bool_set_true:N \g_xeCJK_AutoFakeBold_bool },
1352   AutoFakeBold / false .code:n =
1353   { \bool_set_false:N \g_xeCJK_AutoFakeBold_bool },
1354   AutoFakeBold / unknown .code:n =
1355   {
1356     \bool_set_true:N \g_xeCJK_AutoFakeBold_bool
1357     \fp_set:Nn \g_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1358   },
1359   AutoFakeBold .default:n = { true },
1360   AutoFakeSlant .choice:,
1361   AutoFakeSlant / true .code:n =
1362   { \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool },
1363   AutoFakeSlant / false .code:n =
1364   { \bool_set_false:N \g_xeCJK_AutoFakeSlant_bool },
1365   AutoFakeSlant / unknown .code:n =
1366   {
1367     \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool
1368     \fp_set:Nn \g_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1369   },
1370   AutoFakeSlant .default:n = { true },

```

```

1371     EmboldenFactor .fp_set:N = \g_xeCJK_EmboldenFactor_fp,
1372     SlantFactor    .fp_set:N = \g_xeCJK_SlantFactor_fp,
1373     BoldFont      .meta:n = { AutoFakeBold = true },
1374     boldfont      .meta:n = { AutoFakeBold = true },
1375     SlantFont     .meta:n = { AutoFakeSlant = true },
1376     slantfont     .meta:n = { AutoFakeSlant = true },
1377 }

```

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

\keys_set_known:nxN

```

1378 \cs_generate_variant:Nn \keys_set_known:nn { 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 1379 \clist_new:N \g_xeCJK_sub_key_clist
1380 \cs_new_nopar:Nn \xeCJK_new_sub_key:n
1381 {
1382     \clist_gput_right:Nx \g_xeCJK_sub_key_clist {#1}
1383     \keys_define:nn { xeCJK / features }
1384     {
1385         #1 .code:n =
1386         {
1387             \xeCJK_tl_if_blank:XTF {##1}
1388             {
1389                 \bool_set_false:c { l_xeCJK_copy_#1_bool }
1390                 \bool_set_false:c { l_xeCJK_add_#1_bool }
1391                 \tl_put_right:Nx \l_xeCJK_family_tl { /#1 }
1392             }
1393             {
1394                 \clist_put_right:Nx \l_xeCJK_sub_key_clist {#1}
1395                 \str_if_eq_x:nnTF {##1} *
1396                 {
1397                     \bool_set_true:c { l_xeCJK_copy_#1_bool }
1398                     \bool_set_false:c { l_xeCJK_add_#1_bool }
1399                 }
1400                 {
1401                     \bool_set_false:c { l_xeCJK_copy_#1_bool }
1402                     \bool_set_true:c { l_xeCJK_add_#1_bool }
1403                     \xeCJK_get_sub_features:nn {#1} {##1}
1404                 }
1405             }
1406         },
1407         #1 .default:n = \c_empty_tl,
1408     }
1409 }
1410 \cs_generate_variant:Nn \xeCJK_new_sub_key:n { x }
1411 \cs_new_nopar:Nn \xeCJK_get_sub_features:nn
1412 {
1413     \tl_set:Nx \l_xeCJK_tmpa_tl {#2}
1414     \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1415     \xeCJK_gobble_brace:N \l_xeCJK_tmpa_tl
1416     \exp_args:No \tl_if_head_eq_charcode:nNTF \l_xeCJK_tmpa_tl [ % ]

```

```

1417 {
1418     \exp_after:wN \xeCJK_get_sub_features_aux:n \l_xeCJK_tmpa_tl \c_empty_tl
1419     \xeCJK_tl_if_blank:xT \l_xeCJK_sub_fontname_tl
1420     {
1421         \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_tmpa_tl
1422         \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1423     }
1424 }
1425 { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_tmpa_tl }
1426 \xeCJK_tl_if_blank:xF \l_xeCJK_sub_fontname_tl
1427 { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_fontname_tl }
1428 { \tl_replace_all:Nnx \l_xeCJK_sub_fontname_tl * \l_xeCJK_fontname_tl }
1429 \clist_set:cx { \l_xeCJK_#1_fontfeat_clist } \l_xeCJK_sub_fontfeat_clist
1430 \tl_set:cx { \l_xeCJK_#1_fontname_tl } \l_xeCJK_sub_fontname_tl
1431 }

```

(End definition for \xeCJK_new_sub_key:n and \g_xeCJK_sub_key_clist)

```

\xeCJK_gobble_brace:N
\xeCJK_get_sub_features:nn 1432 \cs_new_nopar:Nn \xeCJK_gobble_brace:N
\exeCJK_get_sub_features_aux:n 1433 {
1434     \exp_args:No \tl_if_head_group:nT #1
1435     {
1436         \exp_last_unbraced:NNo \tl_set:Nn #1 #1
1437         \xeCJK_gobble_brace:N #1
1438     }
1439 }
1440 \cs_new_nopar:Npn \xeCJK_get_sub_features_aux:n [#1] #2
1441 {
1442     \clist_set:Nx \l_xeCJK_sub_fontfeat_clist {#1}
1443     \tl_set:Nx \l_xeCJK_sub_fontname_tl {#2}
1444 }

```

(End definition for \xeCJK_gobble_brace:N, \xeCJK_get_sub_features:nn, and \xeCJK_get_sub_features_aux:n)

Fallback

```

1445 \xeCJK_new_sub_key:n { Fallback }

```

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

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

```

BoldFont 1446 \cs_new_nopar:Nn \xeCJK_map_features_id:n
ItalicFont 1447 { \prop_get:Nn \g_xeCJK_features_id_prop { #1 } }
BoldItalicFont 1448 \prop_new:N \g_xeCJK_features_id_prop
SlantedFont 1449 \prop_put:Nnn \g_xeCJK_features_id_prop { bf } { Bold }
BoldSlantedFont 1450 \prop_put:Nnn \g_xeCJK_features_id_prop { it } { Italic }
BoldFeatures 1451 \prop_put:Nnn \g_xeCJK_features_id_prop { bfit } { BoldItalic }
ItalicFeatures 1452 \prop_put:Nnn \g_xeCJK_features_id_prop { sl } { Slanted }
BoldItalicFeatures 1453 \prop_put:Nnn \g_xeCJK_features_id_prop { bfsl } { BoldSlanted }
SlantedFeatures 1454 \prop_map_inline:Nn \g_xeCJK_features_id_prop
BoldSlantedFeatures 1455 {
1456     \keys_define:nn { xeCJK / features }
1457     {
1458         #2Font .tl_set_x:c = { \l_xeCJK_fontname_#1_tl },
1459         #2Features .tl_set_x:c = { \l_xeCJK_fontfeat_#1_clist } ,

```

```

1460     }
1461 }
(End definition for \g_xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

Mono

```

1462 \keys_define:nn { xeCJK / features }
1463 {
1464   AutoFakeBold .choice:,
1465   AutoFakeBold / false .code:n =
1466     { \bool_set_false:N \l_xeCJK_AutoFakeBold_bool },
1467   AutoFakeBold / unknown .code:n =
1468     {
1469       \bool_set_true:N \l_xeCJK_AutoFakeBold_bool
1470       \fp_set:Nn \l_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1471     },
1472   AutoFakeBold .default:n = { \g_xeCJK_EmboldenFactor_fp },
1473   AutoFakeSlant .choice:,
1474   AutoFakeSlant / false .code:n =
1475     { \bool_set_false:N \l_xeCJK_AutoFakeSlant_bool },
1476   AutoFakeSlant / unknown .code:n =
1477     {
1478       \bool_set_true:N \l_xeCJK_AutoFakeSlant_bool
1479       \fp_set:Nn \l_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1480     },
1481   AutoFakeSlant .default:n = { \g_xeCJK_SlantFactor_fp },
1482   Mono .choice:,
1483   Mono / Exspace .code:n =
1484     {
1485       \xeCJK_set_monoexspace:
1486       \tl_clear:N \l_xeCJK_monoscale_tl
1487     },
1488   Mono / Scale .code:n =
1489     {
1490       \xeCJK_set_monoscale:
1491       \tl_set:Nx \l_xeCJK_monoscale_tl { Scale = { \fp_use:N \g_xeCJK_monoscale_fp } }
1492     }
1493 }

```

(End definition for AutoFakeBold, AutoFakeSlant, and Mono)

\xeCJK_set_init:

```

1494 \cs_new_nopar:Nn \xeCJK_set_init:
1495 {
1496   \tl_clear:N \l_xeCJK_monoscale_tl
1497   \int_gincr:N \g_xeCJK_fontspec_int
1498   \prop_clear:N \l_xeCJK_add_font_prop
1499   \prop_map_inline:Nn \g_xeCJK_features_id_prop
1500     {
1501       \tl_clear:c { l_xeCJK_fontname_##1_tl }
1502       \clist_clear:c { l_xeCJK_fontfeat_##1_clist }
1503     }
1504   \clist_clear:N \l_xeCJK_pass_features_clist
1505   \clist_clear:N \l_xeCJK_sub_key_clist

```

```

1506 \clist_map_inline:Nn \g_xeCJK_sub_key_clist
1507 {
1508     \bool_set_false:c { l_xeCJK_copy_##1_bool }
1509     \bool_set_false:c { l_xeCJK_add_##1_bool }
1510 }
1511 \bool_set_eq:NN \l_xeCJK_AutoFakeBold_bool \g_xeCJK_AutoFakeBold_bool
1512 \bool_set_eq:NN \l_xeCJK_AutoFakeSlant_bool \g_xeCJK_AutoFakeSlant_bool
1513 \fp_set_eq:NN \l_xeCJK_EmboldenFactor_fp \g_xeCJK_EmboldenFactor_fp
1514 \fp_set_eq:NN \l_xeCJK_SlantFactor_fp \g_xeCJK_SlantFactor_fp
1515 }

```

(End definition for \xeCJK_set_init:)

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

```

1516 \cs_new_protected_nopar:Nn \xeCJK_set_family:nnn
1517 {
1518     \group_begin:
1519     \xeCJK_set_init:
1520     \tl_set:Nx \l_xeCJK_family_tl {#1}
1521     \clist_set:Nx \l_xeCJK_fontoptions_clist {#2}
1522     \tl_set:Nx \l_xeCJK_fontname_tl {#3}
1523     \keys_set_known:nxN { xeCJK / features }
1524     { \g_xeCJK_default_features_clist, #2 } \l_xeCJK_pass_features_clist
1525     \xeCJK_parse_features:
1526     \xeCJK_pass_features:
1527     \xeCJK_check_family:V \l_xeCJK_family_tl
1528     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1529     {
1530         \exp_not:n { \fontspec_set_family:Nnn \l_xeCJK_fontspec_family_tl }
1531         { \l_xeCJK_pass_features_clist } { \l_xeCJK_fontname_tl }
1532         \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1533         { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_fontspec_family_tl
1534         \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1535         \exp_not:n
1536         { {
1537             \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1538             \exp_not:N \fontfamily { \l_xeCJK_fontspec_family_tl }
1539             \exp_not:N \selectfont
1540         } }
1541     }
1542     \xeCJK_save_family_info:
1543     \xeCJK_add_sub_block:
1544     \group_end:
1545 }

```

\cs_generate_variant:Nn \xeCJK_set_family:nnn { ncc }

(End definition for \xeCJK_set_family:nnn)

\xeCJK_check_family:n

```

1547 \cs_new_nopar:Nn \xeCJK_check_family:n
1548 {
1549     \prop_get:NnNT \g_xeCJK_family_prop {#1} \l_xeCJK_tmpa_tl
1550     {
1551         \prop_gpop:NnNT \g_xeCJK_family_name_prop {#1} \l_xeCJK_tmpa_tl { }

```

```

1552         \xeCJK_warning:nxx { CJKfamily-redef } {#1} \l_xeCJK_tmpa_tl
1553     }
1554 }
1555 \cs_generate_variant:Nn \xeCJK_check_family:n { V }
1556 \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 1557 \cs_new_nopar:Nn \xeCJK_add_font:nn
1558 {
1559     \prop_put:Nxx \l_xeCJK_add_font_prop
1560     { \xeCJK_map_features_id:n {#1} Font } {#2}
1561 }
1562 \cs_new_nopar:Nn \xeCJK_add_font_if_new:nn
1563 {
1564     \prop_put_if_new:Nxx \l_xeCJK_add_font_prop
1565     { \xeCJK_map_features_id:n {#1} Font } {#2}
1566 }
1567 \cs_generate_variant:Nn \xeCJK_add_font:nn { nx, nc, nV, nv }
1568 \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)

1569 \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1570 \cs_generate_variant:Nn \prop_get:Nn { NV }
1571 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1572 \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1573 \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

```

\xeCJK_add_fake_bold:n
\xeCJK_add_fake_slant:n 1574 \cs_new_nopar:Nn \xeCJK_add_fake_bold:n
1575 {
1576     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1577     { FakeBold = { \fp_use:N \l_xeCJK_EmboldenFactor_fp } }
1578 }
1579 \cs_new_nopar:Nn \xeCJK_add_fake_slant:n
1580 {
1581     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1582     { FakeSlant = { \fp_use:N \l_xeCJK_SlantFactor_fp } }
1583 }
(End definition for \xeCJK_add_fake_bold:n and \xeCJK_add_fake_slant:n)

```

```

\xeCJK_parse_features:
\xeCJK_parse_features:n 1584 \cs_new_nopar:Nn \xeCJK_parse_features:
1585 { \prop_map_inline:Nn \g_xeCJK_features_id_prop { \xeCJK_parse_features:n {##1} } }
1586 \cs_new_nopar:Nn \xeCJK_parse_features:n
1587 {
1588     \xeCJK_if_font_select:nTF {#1}
1589     {
1590         \xeCJK_add_font:nv {#1} { l_xeCJK_fontname_#1_tl }
1591         \xeCJK_if_it_or_sl:nTF {#1}
1592         {
1593             \xeCJK_if_font_select:nF {bf#1}
1594             { \xeCJK_add_font_if_new:nv {bf#1} { l_xeCJK_fontname_#1_tl } }

```

```

1595     }
1596     {
1597         \str_if_eq_x:nnT {#1} {bf}
1598         {
1599             \clist_map_inline:nn { it , sl }
1600             {
1601                 \xeCJK_if_font_select:nF {bf##1}
1602                 { \xeCJK_add_font:nV {bf##1} \l_xeCJK_fontname_bf_tl }
1603             }
1604         }
1605     }
1606 }
1607 { \xeCJK_set_fake:n {#1} }
1608 }

```

(End definition for \xeCJK_parse_features: and \xeCJK_parse_features:n)

```

\xeCJK_if_font_select_p:n
  \xeCJK_if_it_or_sl_p:n 1609 \prg_new_conditional:Nnn \xeCJK_if_font_select:n { p, T, F, TF }
\xeCJK_if_font_select:nTF 1610 {
  \xeCJK_if_it_or_sl:nTF 1611   \exp_args:Nc \xeCJK_tl_if_blank:xTF { l_xeCJK_fontname_#1_tl }
1612   \prg_return_false: \prg_return_true:
1613 }
1614 \prg_new_conditional:Nnn \xeCJK_if_it_or_sl:n { p, T, F, TF }
1615 {
1616   \bool_if:nTF { \str_if_eq_x_p:nn { it } {#1} || \str_if_eq_x_p:nn { sl } {#1} }
1617   \prg_return_true: \prg_return_false:
1618 }

```

(End definition for \xeCJK_if_font_select:n and \xeCJK_if_it_or_sl:n)

\xeCJK_set_fake:n

```

1619 \cs_new_nopar:Nn \xeCJK_set_fake:n
1620 {
1621   \str_if_eq_x:nnTF {#1} { bf }
1622   { \bool_if:NT \l_xeCJK_AutoFakeBold_bool { \xeCJK_add_fake_bold:n {#1} } }
1623   {
1624     \bool_if:NTF \l_xeCJK_AutoFakeSlant_bool
1625     {
1626       \bool_if:nT
1627       { \xeCJK_if_it_or_sl_p:n {#1} ||
1628         ( \str_if_eq_x_p:nn {#1} {bf} ) && ! ( \xeCJK_if_font_select_p:n {it} ) ) ||
1629         ( \str_if_eq_x_p:nn {#1} {bfs} ) && ! ( \xeCJK_if_font_select_p:n {sl} ) )
1630       }
1631       { \xeCJK_add_fake_slant:n {#1} }
1632     }
1633     { \xeCJK_if_it_or_sl:nT {#1} { \xeCJK_map_it_sl:n {#1} } }
1634   } \bool_if:NT
1635   { \l_xeCJK_AutoFakeBold_bool
1636     && ! ( \xeCJK_if_it_or_sl_p:n {#1} )
1637     && ! ( \xeCJK_if_font_select_p:n {bf} )
1638   }
1639   { \xeCJK_add_fake_bold:n {#1} }
1640 }

```

```

1641     \xeCJK_add_font_if_new:nn {#1} *
1642   }
  (End definition for \xeCJK_set_fake:n)

```

\xeCJK_map_it_sl:n

```

1643 \cs_new_nopar:Nn \xeCJK_map_it_sl:n
1644 {
1645   \xeCJK_if_map_font_select:nT {#1}
1646   {
1647     \xeCJK_add_font:nx {#1} { \xeCJK_get_map_font:n {#1} }
1648     \xeCJK_if_font_select:nF {bf#1}
1649     { \xeCJK_add_font_if_new:nx {bf#1} { \xeCJK_get_map_font:n {#1} } }
1650   }
1651 }
  (End definition for \xeCJK_map_it_sl:n)

```

\xeCJK_get_map_font:n

```

1652 \cs_new_nopar:Nn \xeCJK_get_map_font:n
1653 { \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 1654 \prg_new_conditional:Nnn \xeCJK_if_map_font_select:n { p, T, F, TF }
1655 {
1656   \xeCJK_tl_if_blank:xTF { \xeCJK_get_map_font:n {#1} }
1657   \prg_return_false: \prg_return_true:
1658 }
  (End definition for \xeCJK_if_map_font_select:n)

```

\xeCJK_pass_features:

```

1659 \cs_new_nopar:Nn \xeCJK_pass_features:
1660 {
1661   \prop_map_inline:Nn \g_xeCJK_features_id_prop
1662   {
1663     \clist_if_empty:cF { l_xeCJK_fontfeat_##1_clist }
1664     {
1665       \clist_put_right:Nx \l_xeCJK_pass_features_clist
1666       { ##2Features = { \tl_use:c { l_xeCJK_fontfeat_##1_clist } } }
1667     }
1668   }
1669   \prop_map_inline:Nn \l_xeCJK_add_font_prop
1670   { \clist_put_right:Nx \l_xeCJK_pass_features_clist { ##1 = { ##2 } } }
1671   \xeCJK_tl_if_blank:xF \l_xeCJK_monoscale_tl
1672   { \clist_put_right:Nx \l_xeCJK_pass_features_clist \l_xeCJK_monoscale_tl }
1673 }
  (End definition for \xeCJK_pass_features:)

```

\g_xeCJK_family_prop

```

\g_xeCJK_family_name_prop 1674 \prop_new:N \g_xeCJK_family_prop
\g_xeCJK_family_options_prop 1675 \prop_new:N \g_xeCJK_family_name_prop
1676 \prop_new:N \g_xeCJK_family_options_prop
  (End definition for \g_xeCJK_family_prop, \g_xeCJK_family_name_prop, and \g_xeCJK_family_options_prop)

```

\xeCJK_save_family_info:

```
1677 \cs_new_nopar:Nn \xeCJK_save_family_info:
1678 {
1679   \group_begin:
1680   \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1681   {
1682     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist {##1}
1683     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist { ##1 = }
1684   }
1685   \prop_gput:NVV \g_xeCJK_family_prop \l_xeCJK_family_tl \l_xeCJK_fontname_tl
1686   \prop_gput:NVV \g_xeCJK_family_options_prop \l_xeCJK_family_tl \l_xeCJK_fontoptions_clist
1687   \group_end:
1688 }
(End definition for \xeCJK_save_family_info:)
```

\xeCJK_add_sub_block:

```
1689 \cs_new_nopar:Nn \xeCJK_add_sub_block:
1690 {
1691   \clist_remove_duplicates:N \l_xeCJK_sub_key_clist
1692   \clist_map_inline:Nn \l_xeCJK_sub_key_clist
1693   {
1694     \tl_set:Nx \l_xeCJK_sub_family_tl { \l_xeCJK_family_tl/##1 }
1695     \bool_if:cT { l_xeCJK_copy_##1_bool }
1696     {
1697       \xeCJK_check_family:V \l_xeCJK_sub_family_tl
1698       \prop_get:NVNT \g_xeCJK_family_prop \l_xeCJK_family_tl \l_xeCJK_tmpa_tl
1699       { \prop_gput:NVV \g_xeCJK_family_prop \l_xeCJK_sub_family_tl \l_xeCJK_tmpa_tl }
1700       \prop_get:NVNT \g_xeCJK_family_options_prop \l_xeCJK_family_tl \l_xeCJK_tmpa_clist
1701       {
1702         \clist_remove_all:Nn \l_xeCJK_tmpa_clist { ##1 = * }
1703         \prop_gput:NVV \g_xeCJK_family_options_prop \l_xeCJK_sub_family_tl \l_xeCJK_tmpa_clist
1704       }
1705       \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_sub_family_tl }
1706       {
1707         \xeCJK_family_if_exist:xT { \l_xeCJK_family_tl }
1708         {
1709           \prop_get:NnNT \exp_not:N \g_xeCJK_family_name_prop
1710           { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_tmpa_tl
1711           {
1712             \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1713             { \l_xeCJK_sub_family_tl } \exp_not:N \l_xeCJK_tmpa_tl
1714             \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_sub_family_tl }
1715             \exp_not:n
1716             { {
1717               \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1718               \exp_not:N \fontfamily { \l_xeCJK_tmpa_tl }
1719               \exp_not:N \selectfont
1720             } }
1721           }
1722         }
1723       }
1724     }
  }
}
```

```

1725     \bool_if:cT { l_xeCJK_add_##1_bool }
1726     {
1727         \xeCJK_set_family:ncc \l_xeCJK_sub_family_tl
1728         { l_xeCJK_##1_fontfeat_clist } { l_xeCJK_##1_fontname_tl }
1729     }
1730 }
1731 }
(End definition for \xeCJK_add_sub_block:)

```

\xeCJK_copy_family:nn

```

1732 \cs_new_nopar:Nn \xeCJK_copy_family:nn
1733 {
1734     \xeCJK_family_if_exist:xT {#2}
1735     {
1736         \tl_map_inline:nn
1737         { \g_xeCJK_family_prop \g_xeCJK_family_options_prop \g_xeCJK_family_name_prop }
1738         { \prop_get:NnNT ##1 {#2} \l_xeCJK_tmpa_tl { \prop_gput:NnV ##1 {#1} \l_xeCJK_tmpa_tl } }
1739         \cs_gset_eq:cc { xeCJK/family/#1 } { xeCJK/family/#2 }
1740     }
1741 }
1742 \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。

```

\l_xeCJK_select_font: 1743 \tl_set:Nn \l_xeCJK_font_coor_tl { xeCJK/\xeCJK@family/\f@series/\f@shape/\f@size }
\l_xeCJK_switch_font: 1744 \cs_new_nopar:Nn \xeCJK_select_font:
1745 {
1746     \cs_if_exist_use:cF \l_xeCJK_font_coor_tl
1747     {
1748         \tl_set:Nx \l_xeCJK_current_coor_tl \l_xeCJK_font_coor_tl
1749         \cs_if_exist_use:cT { xeCJK/family/\xeCJK@family }
1750         { \exp_last_unbraced:NNV \cs_gset_eq:cN \l_xeCJK_current_coor_tl \tex_font:D }
1751     }
1752 }
1753 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
1754 \cs_new_nopar:Nn \xeCJK_switch_font:nn
1755 {
1756     \str_if_eq_x:nnF {#1} {#2}
1757     {
1758         \xeCJK_info:nxx { CJK-block } {#1} {#2}
1759         \tl_remove_all:Nn \xeCJK@family { /#1 }
1760         \str_if_eq_x:nnF {#2} { CJK }
1761         {
1762             \tl_set:Nx \l_xeCJK_tmpa_tl \xeCJK@family
1763             \tl_put_right:Nx \xeCJK@family { /#2 }
1764             \xeCJK_family_if_exist:xF \xeCJK@family
1765             {
1766                 \xeCJK_family_if_exist:xF { \CJKfamilydefault/#2 }
1767                 { \xeCJK_copy_family:xx \xeCJK@family { \CJKfamilydefault/#2 } }
1768                 { \xeCJK_copy_family:xx \xeCJK@family \l_xeCJK_tmpa_tl }

```

```

1769     }
1770   }
1771 }
1772 \xeCJK_select_font:
1773 }
1774 \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:xF

```

1775 \prg_new_protected_conditional:Nnn \xeCJK_family_if_exist:x { T, F, TF }
1776 {
1777   \cs_if_free:cTF { xeCJK/family/#1 } \prg_return_false:
1778   {
1779     \prop_if_in:NxF \g_xeCJK_family_name_prop {#1} { \use:c { xeCJK/family/#1 } }
1780     \prg_return_true:
1781   }
1782 }
(End definition for \xeCJK_family_if_exist:xF)

```

\CJKfamily 用于切换 CJK 字体族。

```

1783 \NewDocumentCommand \CJKfamily { m }
1784 {
1785   \xeCJK_family_if_exist:xF {#1}
1786   { \tl_set:Nx \xeCJK@family {#1} }
1787   {
1788     \seq_if_in:NxF \g_xeCJK_unknown_family_seq {#1}
1789     {
1790       \seq_gput_right:Nx \g_xeCJK_unknown_family_seq {#1}
1791       \xeCJK_warning:nx { CJKfamily-Unknown } {#1}
1792     }
1793   }
1794   \tex_ignorespaces:D
1795 }
1796 \seq_new:N \g_xeCJK_unknown_family_seq
1797 \xeCJK_msg_new:nn { CJKfamily-Unknown }
1798 {
1799   Unknown~CJK~family~'#1'~is~ignored.\\\
1800   Try~to~use~
1801   \str_case:x:nnn {#1}
1802   {
1803     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1804     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1805     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1806   }
1807   { \token_to_str:N \setCJKfamilyfont {#1} }
1808   [...]{}~to~define~it.
1809 }
(End definition for \CJKfamily This function is documented on page 6.)

```

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

```

1810 \newCJKfontfamily
1811 \CJKfontspec { \xeCJK_set_family:nnn {#1} {#2} {#3} }

```

```

1812 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1813 {
1814   \IfNoValueTF {#1}
1815     { \tl_set:Nx \l_xeCJK_family_tl { \cs_to_str:N #2 } }
1816     { \tl_set:Nx \l_xeCJK_family_tl {#1} }
1817   \xeCJK_set_family:nnn \l_xeCJK_family_tl {#3} {#4}
1818   \cs_new_protected_nopar:Npx #2 { \exp_not:N \CJKfamily { \l_xeCJK_family_tl } }
1819 }
1820 \int_new:N \g_xeCJK_fontspec_int
1821 \int_gdecr:N \g_xeCJK_fontspec_int
1822 \NewDocumentCommand \CJKfontspec { O{} m }
1823 {
1824   \tl_set:Nx \l_xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1825   \cs_if_free:cT \l_xeCJK_CJKfontspec_id_tl
1826   {
1827     \int_gincr:N \g_xeCJK_fontspec_int
1828     \tl_gset:Nx \g_xeCJK_family_spec_tl
1829       { CJKfontspec ( \int_use:N \g_xeCJK_fontspec_int ) }
1830     \xeCJK_set_family:nnn \g_xeCJK_family_spec_tl {#1} {#2}
1831     \tl_gset:cx \l_xeCJK_CJKfontspec_id_tl \g_xeCJK_family_spec_tl
1832   }
1833   \exp_args:Nv \CJKfamily \l_xeCJK_CJKfontspec_id_tl
1834   \tex_ignorespaces:D
1835 }

```

(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。

```

\addCJKfontfeatures 1836 \clist_new:N \g_xeCJK_default_features_clist
1837 \NewDocumentCommand \defaultCJKfontfeatures { m }
1838 { \clist_gset:Nn \g_xeCJK_default_features_clist {#1} }
1839 \@onlypreamble \defaultCJKfontfeatures
1840 \NewDocumentCommand \addCJKfontfeatures { m }
1841 {
1842   \prop_if_in:NVTF \g_xeCJK_family_prop \xeCJK@family
1843   {
1844     \group_begin:
1845     \clist_set:Nx \l_xeCJK_tmpa_clist {#1}
1846     \prop_get:NVN \g_xeCJK_family_prop \xeCJK@family \l_xeCJK_tmpa_tl
1847     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1848     {
1849       \clist_if_in:NnT \l_xeCJK_tmpa_clist {##1}
1850       {
1851         \clist_remove_all:Nn \l_xeCJK_tmpa_clist {##1}
1852         \prop_get:NxNF \g_xeCJK_family_prop { \xeCJK@family/##1 } \l_xeCJK_tmpb_tl
1853         {
1854           \prop_get:NxNF \g_xeCJK_family_prop
1855             { \CJKfamilydefault/##1 } \l_xeCJK_tmpb_tl
1856           { \tl_set:Nx \l_xeCJK_tmpb_tl \l_xeCJK_tmpa_tl }
1857         }
1858         \clist_set:Nx \l_xeCJK_tmpa_clist
1859           { ##1 = { [ \l_xeCJK_tmpa_clist ] { \l_xeCJK_tmpb_tl } } }
1860         \clist_map_break:

```

```

1861     }
1862   }
1863   \prop_get:NVNF \g_xeCJK_family_options_prop \xeCJK@family \l_xeCJK_tmpb_clist
1864   { \clist_clear:N \l_xeCJK_tmpb_clist }
1865   \clist_clear:N \l_xeCJK_tmpe_clist
1866   \clist_map_inline:Nn \l_xeCJK_tmpe_clist
1867   {
1868     \clist_if_in:NnF \l_xeCJK_tmpe_clist {##1}
1869     { \clist_put_right:Nn \l_xeCJK_tmpe_clist {##1} }
1870   }
1871   \clist_put_left:NV \l_xeCJK_tmpe_clist \l_xeCJK_tmpe_clist
1872   \use:x
1873   { \group_end: \exp_not:N \CJKfontspec [ \l_xeCJK_tmpe_clist ] { \l_xeCJK_tmpe_tl } }
1874 }
1875 { \xeCJK_warning:n { addCJKfontfeature-ignored } }
1876 \tex_ignorespaces:D
1877 }
1878 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
1879 \xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1880 {
1881   \tl_to_str:N \addCJKfontfeature (s)~ignored.\\\
1882   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1883 }

```

(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 1884 \NewDocumentCommand \setCJKmainfont { O{} m }
1885   { \xeCJK_set_family:nnn \CJKrmdefault {#1} {#2} }
1886 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1887 \NewDocumentCommand \setCJKsansfont { O{} m }
1888   { \xeCJK_set_family:nnn \CJKsfdefault {#1} {#2} }

```

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

```

\CJKrmdefault
\CJKsfdefault 1889 \cs_if_free:NT \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
\CJKttdefault 1890 \cs_if_free:NT \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 1891 \cs_if_free:NT \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1892 \cs_if_free:NT \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1893 \tl_new:c { xeCJK/family/\CJKfamilydefault }

```

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

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

```

1894 \AtEndOfPackage
1895 {
1896   \tex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1897   \tex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }
1898   \tex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1899   \tex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1900   \cs_set_eq:NN \reset@font \normalfont
1901 }

```

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。

```

1902 \onlypreamble \setCJKmainfont
1903 \onlypreamble \setCJKmathfont
1904 \onlypreamble \setCJKsansfont
1905 \onlypreamble \setCJKmonofont
1906 \onlypreamble \setCJKromanfont

```

5.12 数学字体设置

\setCJKmathfont 设置 CJK 数学字体。

```

1907 \NewDocumentCommand \setCJKmathfont { O{ } m }
1908 { \xeCJK_set_family:nnn \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。

```

1909 \tl_const:Nn \c_xeCJK_math_tl { CJKmath }
1910 \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

```

1911 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1912 {
1913   \xeCJK_check_num_range:nnNN {#1} {#2} \l_xeCJK_tmpa_int \l_xeCJK_tmppb_int
1914   \int_set:Nn \l_xeCJK_tmppc_int { \xeCJK_math_type:n {#3} }
1915   \loop
1916     \XeTeXmathcode
1917     \l_xeCJK_tmpa_int = \l_xeCJK_tmppc_int \use:c { sym #4 } \l_xeCJK_tmpa_int
1918     \if_int_compare:w \l_xeCJK_tmpa_int < \l_xeCJK_tmppb_int
1919       \int_incr:N \l_xeCJK_tmpa_int
1920   \repeat
1921 }
1922 \cs_new_eq:NN \xeCJK_math_type:n \mathchar@type
(End definition for \xeCJKsetmathcode)

```

\xeCJK_set_mathfont:

```

1923 \cs_new_nopar:Nn \xeCJK_set_mathfont:
1924 {
1925   \xeCJK_family_if_exist:xF \c_xeCJK_math_tl
1926   { \tl_set:Nx \l_xeCJK_tmpa_tl \c_xeCJK_math_tl }
1927   {
1928     \xeCJK_family_if_exist:xF \CJKfamilydefault
1929     { \tl_set:Nx \l_xeCJK_tmpa_tl \CJKfamilydefault }
1930     { \tl_set_eq:NN \l_xeCJK_tmpa_tl \q_no_value }
1931   }
1932   \prop_get:NVNT \g_xeCJK_family_name_prop \l_xeCJK_tmpa_tl \l_xeCJK_math_family_tl
1933   {
1934     \DeclareSymbolFont \c_xeCJK_math_tl \g_fontspec_encoding_tl
1935     \l_xeCJK_math_family_tl \mddefault \shapedefault
1936     \SetSymbolFont \c_xeCJK_math_tl { bold } \g_fontspec_encoding_tl
1937     \l_xeCJK_math_family_tl \bfdefault \shapedefault
1938     \clist_map_inline:nn { CJK, FullLeft, FullRight }

```

```

1939         {
1940             \clist_map_inline:cn { g_xeCJK_##1_range_clist }
1941             {
1942                 \xeCJK_setcharclass_aux:Nn \xeCJKsetmathcode {####1}
1943                 \mathalpha \c_xeCJK_math_tl
1944             }
1945         }
1946     }
1947 }

```

(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。

```

1948 \xeCJK_AtEndPreamble:n
1949 {
1950     \normalfont
1951     \prop_if_empty:NTF \g_xeCJK_family_prop
1952     { \xeCJK_warning:n { no-CJKfamily } }
1953     { \bool_if:NT \g_xeCJK_math_bool { \xeCJK_set_mathfont: } }
1954 }
1955 \xeCJK_msg_new:nn { no-CJKfamily }
1956 {
1957     It~seems~that~you~have~not~declare~a~CJKfamily.\\
1958     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
1959     \token_to_str:N \xeCJKmainfont[...]{...}\\\\
1960     in~the~preamble~to~declare~the~main~CJKfamily.\\
1961 }

```

5.13 处理等宽字体和抄录环境

\g_xeCJK_monoscale_fp

\g_xeCJK_exspace_dim

```

1962 \fp_new:N \g_xeCJK_monoscale_fp
1963 \dim_new:N \g_xeCJK_exspace_dim

```

(End definition for \g_xeCJK_monoscale_fp and \g_xeCJK_exspace_dim)

\setCJKmonoscale

```

1964 \NewDocumentCommand \setCJKmonoscale { }
1965 {
1966     \CJKflexiblespacing
1967     \xeCJK_set_monoscale:
1968     \addCJKfontfeatures { Scale = \fp_use:N \g_xeCJK_monoscale_fp }
1969 }
1970 \cs_new_nopar:Nn \xeCJK_set_monoscale:
1971 {
1972     \group_begin:
1973     \fontfamily \ttdefault \selectfont
1974     \fp_gset:Nn \g_xeCJK_monoscale_fp
1975     { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } / ( \f@size ) }
1976     \group_end:
1977 }

```

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

\setCJKmonoexspace

```
1978 \NewDocumentCommand \setCJKmonoexspace { } { \xeCJK_set_monoexspace: }
1979 \cs_new_nopar:Nn \xeCJK_set_monoexspace:
1980 {
1981   \group_begin:
1982     \fontfamily \ttdefault \selectfont
1983     \dim_gset:Nn \g_xeCJK_exspace_dim
1984       { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size \p@ }
1985   \group_end:
1986 }
```

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

\CJKfixedspacing

\CJKflexiblespacing

```
1987 \NewDocumentCommand \CJKfixedspacing { }
1988 {
1989   \bool_if:NF \l_xeCJK_fixed_spacing_bool
1990   {
1991     \bool_set_true:N \l_xeCJK_fixed_spacing_bool
1992     \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1993     \bool_set_eq:NN \l_xeCJK_fixed_xecglue_bool \l_xeCJK_xecglue_bool
1994     \cs_set_eq:NN \xeCJK_flexible_ecglue: \CJKecglue
1995     \cs_set_eq:NN \xeCJK_flexible_cjkglue: \CJKglue
1996     \xeCJK_set_monoexspace:
1997     \xeCJKsetup
1998     {
1999       PunctStyle = plain ,
2000       CJKglue    = { \skip_horizontal:N \g_xeCJK_exspace_dim } ,
2001       CJKecglue  = { \skip_horizontal:n { .5\g_xeCJK_exspace_dim } } ,
2002       xCJKecglue = false ,
2003     }
2004   }
2005 }
2006 \NewDocumentCommand \CJKflexiblespacing { }
2007 {
2008   \bool_if:NT \l_xeCJK_fixed_spacing_bool
2009   {
2010     \bool_set_false:N \l_xeCJK_fixed_spacing_bool
2011     \exp_args:Nx \xeCJKsetup
2012     {
2013       PunctStyle = { \l_xeCJK_flexible_punctstyle_tl } ,
2014       CJKglue    = { \xeCJK_flexible_cjkglue: } ,
2015       CJKecglue  = { \xeCJK_flexible_ecglue: } ,
2016       xCJKecglue = { \bool_if:NTF \l_xeCJK_fixed_xecglue_bool { true } { false } } ,
2017     }
2018   }
2019 }
2020 \bool_new:N \l_xeCJK_fixed_spacing_bool
2021 \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。

```
2022 \NewDocumentCommand \setCJKmonofont { s t+ 0{ } m }
```

```

2023 {
2024   \IfBooleanTF {#1}
2025     { \xeCJK_set_family:nnn \CJKttdefault { Mono = Scale, #3 } {#4} }
2026     {
2027       \IfBooleanTF {#2}
2028         { \xeCJK_set_family:nnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
2029         { \xeCJK_set_family:nnn \CJKttdefault {#3} {#4} }
2030     }
2031 }

```

(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

```

2032 \bool_new:N \g_xeCJK_indent_bool
2033 \keys_define:nn { xeCJK / options }
2034 {
2035   CJKnumber      .bool_set:N = \g_xeCJK_number_bool ,
2036   indentfirst    .bool_set:N = \g_xeCJK_indent_bool ,
2037   normalindentfirst .meta:n = { indentfirst = false } ,
2038   quiet .code:n =
2039     {
2040       \msg_redirect_module:nnn { xeCJK } { warning } { info }
2041       \msg_redirect_module:nnn { xeCJK } { info } { none }
2042       \PassOptionsToPackage { quiet } { fontspec }
2043     },
2044   silent .code:n =
2045     {
2046       \msg_redirect_module:nnn { xeCJK } { warning } { none }
2047       \msg_redirect_module:nnn { xeCJK } { info } { none }
2048       \PassOptionsToPackage { silent } { fontspec }
2049     },
2050   unknown .code:n =
2051     {
2052       \@ifpackageloaded { fontspec }
2053       { \xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2054       { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2055     },
2056 }
2057 \xeCJK_msg_new:nn { key-unknown }
2058 {
2059   Sorry,~but~\l_keys_module_tl\c_space_tl does~not~have~a~key~called~'#1'.\\
2060   The~key~'#1'~is~being~ignored.
2061 }

```

(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。

```

2062 \keys_set:nn { xeCJK / options }

```

```

2063 {
2064   CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
2065   CJKecglue      = \c_space_token ,
2066   xCJKecglue     = false ,
2067   CheckSingle    = false ,
2068   CJKspace       = false ,
2069   CJKmath        = false ,
2070   xeCJKactive    = true ,
2071   indentfirst    = true ,
2072   EmboldenFactor = 4 ,
2073   SlantFactor    = .167 ,
2074   PunctStyle     = quanjiao ,
2075   KaiMingPunct   = { . 。 ? ! } ,
2076   LongPunct      = { — … ‣ — } ,
2077   MiddlePunct    = { · ∙ ∙ } ,
2078   AllowBreakBetweenPuncts = false ,
2079 }

\CJKsymbol
\CJKpunctsymbol 2080 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2081 \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。
2082 \ProcessKeysOptions { xeCJK / options }
2083 \RequirePackage { fontspec } [ 2011/09/13 ]

2084 \bool_if:NT \g_xeCJK_indent_bool
2085 {
2086   \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
2087   \@afterindenttrue
2088 }

\XeCJKsetup 在导言区或文档中设置 xeCJK 的接口。
2089 \NewDocumentCommand \XeCJKsetup { m }
2090 {
2091   \keys_set:nn { xeCJK / options } {#1}
2092   \tex_ignorespaces:D
2093 }
(End definition for \XeCJKsetup This function is documented on page 3.)

\XeCJKsetemboldenfactor
\XeCJKsetslantfactor 2094 \NewDocumentCommand \XeCJKsetemboldenfactor { m }
2095 { \XeCJKsetup { EmboldenFactor = {#1} } }
2096 \NewDocumentCommand \XeCJKsetslantfactor { m }
2097 { \XeCJKsetup { SlantFactor = {#1} } }
(End definition for \XeCJKsetemboldenfactor and \XeCJKsetslantfactor)

\punctstyle
\XeCJKplainchr 2098 \NewDocumentCommand \punctstyle { m } { \XeCJKsetup { PunctStyle = {#1} } }
2099 \cs_new_nopar:Npn \XeCJKplainchr { \punctstyle { plain } }
(End definition for \punctstyle and \XeCJKplainchr)

```

\CJKsetecglue

```
2100 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2101 \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。

```
2102 \cs_new_nopar:Nn \xeCJK_fix_itcorr:
2103 {
2104   \scan_stop:
2105   \xeCJK_if_last_node:nTF { default }
2106   {
2107     \tex_unkern:D \tex_unkern:D
2108     \xeCJK_itcorr_aux
2109     { \xeCJK_make_node:n { default_itcorr } }
2110   }
2111   { \xeCJK_itcorr_aux }
2112 }
2113 \cs_new_eq:NN \xeCJK_itcorr_aux \
2114 \cs_set_protected_nopar:Npn \
2115 {
2116   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
2117   { \xeCJK_fix_itcorr: } { \xeCJK_itcorr_aux }
2118 }
2119 \cs_set_eq:NN @@italiccorr \
(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。

```
\xeCJK_Others_class 2120 \xeCJK_AfterEndPreamble:n
2121 {
2122   \int_compare:nNnF
2123   { \c_three + \seq_count:N \g_xeCJK_class_seq } = \xe@alloc@intercharclass
2124   {
2125     \int_set_eq:NN \l_xeCJK_tmpa_int \c_three
2126     \int_until_do:nNnn \l_xeCJK_tmpa_int = \xe@alloc@intercharclass
2127     {
2128       \int_incr:N \l_xeCJK_tmpa_int
2129       \seq_if_in:NVF \g_xeCJK_class_seq \l_xeCJK_tmpa_int
2130       { \xeCJK_set_others_toks:n \l_xeCJK_tmpa_int }
2131     }
2132   }
2133 }
2134 \cs_new_nopar:Nn \xeCJK_set_others_toks:n
2135 {
2136   \int_set:Nn \xeCJK_Others_class {#1}
2137   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
2138   {
2139     \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2140     \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2141     \exp_args:Nnnx \xeCJK_app_inter_class_toks:nnn {##1} { Others }
2142     { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
```

```

2143 \exp_args:Nnnx \xeCJK_pre_inter_class_toks:nnn { Others } {##1}
2144 { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
2145 \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2146 {
2147   \xeCJK_copy_inter_class_toks:nnnn
2148   { Others } { Boundary } { Default } { Boundary }
2149 }
2150 \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2151 {
2152   \xeCJK_copy_inter_class_toks:nnnn
2153   { Boundary } { Others } { Boundary } { Default }
2154 }
2155 }
2156 }
2157 \int_new:N \xeCJK_Others_class
(End definition for \xeCJK_set_others_toks:n and \xeCJK_Others_class)

```

`\xeCJK_patch:Nnn` 给已有宏内容前后附加补丁。

```

2158 \cs_new_nopar:Nn \xeCJK_patch:Nnn
2159 { \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。

```

2160 \xeCJK_AfterPreamble:n
2161 {
2162   \tl_map_inline:nn
2163   {
2164     \textellipsis \textemdash \textperiodcentered \textcentereddot
2165     \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2166   }
2167   { \xeCJK_patch:Nnn #1 { \group_begin: \makexeCJKinactive } { \group_end: } }
2168   \xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2169   \cs_set_eq:NN \xeCJK_aux_r \r
2170   \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \xeCJK_aux_r {#1} } }
2171   \@ifpackageloaded { pifont }
2172   {
2173     \RenewDocumentCommand \Pifont { m }
2174     {
2175       \fontencoding {U} \fontfamily {#1} \fontseries {m} \fontshape {n}
2176       \selectfont \makexeCJKinactive
2177     }
2178   } { }
2179 }

```

禁止在 `xeCJK` 宏包后再载入 `CJK` 宏包。

```

2180 \tl_set:cn { ver@CJK.sty } { 2050/01/01 }

```

`\xeCJKcaption` 可以使用 `CJK` 宏包中的 `.cpx` 文件。

```

2181 \cs_if_free:NT \CJK@ifundefined
2182 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2183 \NewDocumentCommand \xeCJKcaption { o m }
2184 {

```

```

2185 \IfValueT {#1} { \XeTeXdefaultencoding "#1" }
2186 \cs_set_nopar:Nx \xeCJK_reset_at_catcode:
2187 {
2188     \exp_not:n { \char_set_catcode:nn { \@ } }
2189     { \char_value_catcode:n { \@ } }
2190 }
2191 \char_set_catcode_letter:N \@
2192 \file_input:n { #2.cpx }
2193 \xeCJK_reset_at_catcode:
2194 \XeTeXdefaultencoding "UTF-8"
2195 }

```

(End definition for \xeCJKcaption)

\xeCJK_ULprepunctchar:n

\xeCJK_ULpostpunctchar:n

\xeCJK_ULroutines:

```

2196 \cs_new_protected_nopar:Nn \xeCJK_ULprepunctchar:n
2197 {
2198     { \makeXeCJKinactive \CJKpunctsymbol {#1} \nobreak }
2199     \tex_ignorespaces:D
2200 }
2201 \cs_new_protected_nopar:Nn \xeCJK_ULpostpunctchar:n
2202 {
2203     { \makeXeCJKinactive \CJKpunctsymbol {#1} }
2204     \xeCJK_ignorespaces:
2205 }
2206 \cs_new_protected_nopar:Nn \xeCJK_ULroutines:
2207 {
2208     \xeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKecglue \CJKsymbol }
2209     \xeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2210     \xeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKecglue \CJKsymbol }
2211     \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }
2212     \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2213     {
2214         \xeCJK_inter_class_toks:nnn {##1} { FullLeft } { \xeCJK_ULprepunctchar:n }
2215         \xeCJK_inter_class_toks:nnn {##1} { FullRight } { \xeCJK_ULpostpunctchar:n }
2216     }
2217     \xeCJK_UL_subclass_patch_tl
2218 }

```

(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。

```

2219 \xeCJK_AfterPreamble:n
2220 {
2221     \cs_if_exist:NT \UL@hook
2222     {
2223         \addto@hook \UL@hook
2224         {
2225             \cs_set_eq:NN \xeCJK_UL_CJKsymbol \CJKsymbol
2226             \cs_set_eq:NN \xeCJK_UL_CJKpunctsymbol \CJKpunctsymbol
2227             \cs_set_nopar:Npn \CJKsymbol #1
2228             {
2229                 { \xeCJK_select_font: \xeCJK_UL_CJKsymbol {#1} }
2230                 \xeCJK_make_node:n { CJK } \xeCJK_ignorespaces:
2231             }

```

```

2232         \cs_set_nopar:Npn \CJKpunctsymbol #1
2233         { { \xeCJK_select_font: \xeCJK_UL_CJKpunctsymbol {#1} } }
2234         \xeCJK_ULroutines:
2235     }
2236 }
2237 \cs_if_exist:NT \XeTeX@CJKfntef@hook
2238 {
2239     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2240     { \xeCJK_select_font: \makeCJKinactive }
2241 }
2242 }
    使用 CJKnumb 宏包。
2243 \bool_if:NT \g_xeCJK_number_bool
2244 {
2245     \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2246     \cs_set_nopar:Npn \CJKaddEncHook #1#2 { \cs_set_nopar:cpn { xeCJK_enc_#1 } {#2} }
2247     \cs_set_nopar:Npn \Unicode #1#2
2248     { \tex_char:D \int_eval:n { (#1) * \c_two_hundred_fifty_six + (#2) } }
2249     \RequirePackage { CJKnumb }
2250     \use:c { xeCJK_enc_ \CJK@UnicodeEnc }
2251     \tl_set:Nn \CJK@tenthousand { 万 }
2252     \tl_set:Nn \CJK@hundredmillion { 亿 }
2253 }
2254 \</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}{l}
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} \\ \hline
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} \\ \cline{2-5}
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 汉字源流考
62
63 \textbf{汉字源流考}
64
65 \textsl{汉字源流考}
66
67 \CJKunderwave{汉字源流考}
68 \begin{table}[ht]
69 \caption{生僻字测试}
70 \medskip\centering
71 \begin{tabular}{*4{<{\ttfamily U+}>1}}{

```

```

72 北 & 3400 & 西 & 3401 & 尕 & 3402 & 个 & 3403 \\
73 丰 & 3404 & 乂 & 3405 & 冂 & 3406 & 𠂇 & 3407 \\
74 仝 & 3408 & 𠂇 & 3409 & 𠂇 & 340A & 𠂇 & 340B \\
75 𠂇 & 340C & 𠂇 & 340D & 𠂇 & 340E & 𠂇 & 341F \\
76 𠂇 & 3410 & 𠂇 & 3411 & 𠂇 & 3412 & 𠂇 & 3413 \\
77 𠂇 & 3414 & 𠂇 & 3415 & 𠂇 & 3416 & 𠂇 & 3417 \\
78 𠂇 & 3418 & 𠂇 & 3419 & 𠂇 & 341A & 𠂇 & 341B \\
79 𠂇 & 341C & 𠂇 & 341D & 𠂇 & 341E & 𠂇 & 341F \\[1ex]
80 𠂇 & 20000 & 𠂇 & 20001 & 𠂇 & 20002 & 𠂇 & 20003 \\
81 𠂇 & 20004 & 𠂇 & 20005 & 𠂇 & 20006 & 𠂇 & 20007 \\
82 𠂇 & 20008 & 𠂇 & 20009 & 𠂇 & 2000A & 𠂇 & 2000B \\
83 𠂇 & 2000C & 𠂇 & 2000D & 𠂇 & 2000E & 𠂇 & 2000F \\
84 𠂇 & 20010 & 𠂇 & 20011 & 𠂇 & 20012 & 𠂇 & 20013 \\
85 𠂇 & 20014 & 𠂇 & 20015 & 𠂇 & 20016 & 𠂇 & 20017 \\
86 𠂇 & 20018 & 𠂇 & 20019 & 𠂇 & 2001A & 𠂇 & 2001B \\
87 𠂇 & 2001C & 𠂇 & 2001D & 𠂇 & 2001E & 𠂇 & 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]{Meiryo}
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（英语：CJKV Unified Ideographs），旧称中日韩统一表意文字（英
语：CJK Unified Ideographs），也称统一汉字（英语：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: Chinese-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 집합으로, 한국, 중국, 일본에서 쓰이는 한자를 묶은 것이기 때문에 머리 글자를 따서 한중일 (CJK) 통합 한자라고 불렀는데, 최근에는 베트남에서 쓰이는 한자도 추가되었기에 한중일월 (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{|c>{\ttfamily U+}l|}}\end{tabular}

129 𠂇 & 3400 & 𠂈 & 3401 & 𠂉 & 3402 & 𠂊 & 3403 & \\\end{tabular}

130 𠂋 & 3404 & 𠂌 & 3405 & 𠂍 & 3406 & 𠂎 & 3407 & \\\end{tabular}

131 𠂏 & 3408 & 𠂐 & 3409 & 𠂑 & 340A & 𠂒 & 340B & \\\end{tabular}

132 𠂓 & 340C & 𠂔 & 340D & 𠂕 & 340E & 𠂖 & 341F & \\\end{tabular}

133 𠂗 & 3410 & 𠂘 & 3411 & 𠂙 & 3412 & 𠂚 & 3413 & \\\end{tabular}

134 𠂛 & 3414 & 𠂜 & 3415 & 𠂝 & 3416 & 𠂞 & 3417 & \\\end{tabular}

135 𠂟 & 3418 & 𠂠 & 3419 & 𠂡 & 341A & 𠂢 & 341B & \\\end{tabular}

136 𠂣 & 341C & 𠂤 & 341D & 𠂥 & 341E & 𠂦 & 341F & \\\[1ex]\end{tabular}

137 𠂧 & 20000 & 𠂨 & 20001 & 𠂩 & 20002 & 𠂪 & 20003 & \\\end{tabular}

138 𠂫 & 20004 & 𠂬 & 20005 & 𠂭 & 20006 & 𠂮 & 20007 & \\\end{tabular}

139 𠂯 & 20008 & 𠂰 & 20009 & 𠂱 & 2000A & 𠂲 & 2000B & \\\end{tabular}

140 𠂳 & 2000C & 𠂴 & 2000D & 𠂵 & 2000E & 𠂶 & 2000F & \\\end{tabular}

141 𠂷 & 20010 & 𠂸 & 20011 & 𠂹 & 20012 & 𠂺 & 20013 & \\\end{tabular}

142 𠂻 & 20014 & 𠂼 & 20015 & 𠂽 & 20016 & 𠂾 & 20017 & \\\end{tabular}

143 𠂿 & 20018 & 𠃀 & 20019 & 𠃁 & 2001A & 𠃂 & 2001B & \\\end{tabular}

144 𠃃 & 2001C & 𠃄 & 2001D & 𠃅 & 2001E & 𠃆 & 2001F & \\\end{tabular}

145 \end{tabular}

146 \end{table}

147 \end{document}

148 \langle /ex-block \rangle

6.4 xeCJK-example-CJKecglue.tex

```
149  $\langle$ *ex-ecglue $\rangle$ 
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  $\langle$ /ex-ecglue $\rangle$ 
```

6.5 xeCJK-example-checksingle.tex

```
168  $\langle$ *ex-single $\rangle$ 
169 \documentclass{minimal}
170 \usepackage{xeCJK}
171 \setCJKmainfont{SimSun}
172 \catcode`\。=\active
173 \def。{. }
174 \def\foo{一}
175 \long\def\showtext{一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十。
176   $$x^2+y^2$$
177   一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十\foo
178   \begin{equation}
179     x^2+y^2
180   \end{equation}
181   一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十。}
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  $\langle$ /ex-single $\rangle$ 
```

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,而直立的数字表示对应项使用时所在的行号。

70

\c_one	145, 514, 1262	\CJKpunctsymbol	503, 536, 554, 573, 2080, 2081, 2198, 2203, 2226, 2232
\c_space_tl	25, 126, 864, 2059	\CJKrmdefault	7, 57, 1803, 1885, 1889, 1889, 1892, 1897
\c_space_token	104, 366, 380, 433, 434, 613, 647, 672, 676, 745, 753, 754, 2065	\CJKsetecglue	2100, 2100, 2101
\c_ten	405, 422	\CJKsfdefault	7, 1804, 1888, 1889, 1890, 1898
\c_three	443, 1263, 1265, 2123, 2125	\CJKspace	110, 626, 629, 635, 635
\c_two	442, 1262, 1975, 1984	\CJKsymbol	348, 394, 453, 463, 769, 777, 794, 820, 821, 825, 826, 926, 927, 1285, 1286, 1294, 2080, 2080, 2208, 2209, 2210, 2225, 2227
\c_two_hundred_fifty_six	2248	\CJKsymbol_Block	820, 825
\c_xeCJK_math_tl	1908, 1909, 1925, 1926, 1934, 1936, 1943	\CJKttdefault	7, 1805, 1889, 1891, 1899, 2025, 2028, 2029
\c_xeCJK_ps_CCT_tl	1067, 1099	\CJKunderwave	67
\c_xeCJK_ps_fullwidth_tl	1068, 1172	\cline	17, 29, 41
\c_xeCJK_ps_halfwidth_tl	1065, 1090, 1188	\clist_clear:c	1502
\c_xeCJK_ps_margin_kerning_tl	1164	\clist_clear:N	1414, 1422, 1504, 1505, 1864, 1865
\c_xeCJK_ps_mixedwidth_tl	1066, 1091	\clist_gclear_new:c	156, 169
\c_xeCJK_ps_plain_tl	949, 1028, 1153, 1247	\clist_gput_right:cx	182
\c_zero	123, 126, 146, 2116	\clist_gput_right:Nn	911
\c_zero_dim	484, 485, 1031, 1032, 1052, 1060, 1077, 1110, 1211	\clist_gput_right:Nx	174, 802, 1382
\c_zero_skip	991, 2064	\clist_gremove_all:Nn	808
\caption	69, 126	\clist_gremove_duplicates:N	803
\catcode	172	\clist_gset:Nn	1838
\centering	9, 70, 127	\clist_if_empty:cF	1663
\char_set_catcode:nn	2188	\clist_if_empty:NF	899
\char_set_catcode_ignore:n	147	\clist_if_in:NnF	1868
\char_set_catcode_letter:N	2191	\clist_if_in:NnT	1849
\char_set_catcode_other:N	62	\clist_if_in:NnTF	792
\char_set_catcode_other:n	200	\clist_map_break:	1860
\char_set_lccode:nn	57, 58, 59, 60, 61	\clist_map_inline:cn	1940
\char_value_catcode:n	2189	\clist_map_inline:Nn	168, 332, 466, 882, 901, 941, 993, 1140, 1245, 1506, 1680, 1692, 1847, 1866, 2137
\CheckSingle	687	\clist_map_inline:nn	178, 194, 335, 337, 349, 352, 376, 454, 475, 477, 913, 923, 1030, 1599, 1938, 2212
\CJK@hundredmillion	2252	\clist_map_inline:xn	183, 807, 850, 1332
\CJK@ifundefined	2181, 2182	\clist_new:N	41, 42, 43, 170, 817, 938, 1379, 1836
\CJK@tenthousand	2251	\clist_put_left:cx	1336, 1576, 1581
\CJK@UnicodeEnc	2245, 2250	\clist_put_left:NV	1871
\CJKaddEncHook	2246	\clist_put_right:Nn	1869
\CJKecglue	366, 369, 374, 385, 408, 426, 597, 601, 602, 608, 618, 619, 620, 665, 682, 926, 1994, 2208, 2210	\clist_put_right:Nx	1394, 1665, 1670, 1672
xCJKecglue	4	\clist_remove_all:Nn	1682, 1683, 1702, 1851
\CJKfamily	6, 1783, 1783, 1818, 1833, 1896, 1897, 1898, 1899	\clist_remove_duplicates:N	1691
\CJKfamilydefault	7, 1766, 1767, 1855, 1889, 1892, 1893, 1896, 1928, 1929	\clist_set:cx	1429
\CJKfixedspacing	6, 1987, 1987, 2021	\clist_set:Nn	163, 939, 992
\CJKflexiblespacing	6, 1966, 1987, 2006	\clist_set:Nx	1442, 1521, 1845, 1858
\CJKfontspec	6, 1810, 1822, 1873	\cs_end:	73
\CJKglue	400, 433, 453, 526, 596, 596, 1995		
\CJKmath	1909		
\CJKnospace	114, 630, 635, 640		
\CJKnumber	2032		

`\cs_generate_variant:Nn` . 177, 194, 291,
 319, 498, 811, 812, 813, 814, 815, 943,
 1019, 1020, 1021, 1022, 1120, 1139,
 1213, 1378, 1410, 1546, 1555, 1567,
 1568, 1569, 1570, 1571, 1572, 1573, 1742
`\cs_gset_eq:cc` 1739
`\cs_gset_eq:cN` 1750
`\cs_gset_eq:NN` 820, 821, 825
`\cs_gset_protected_nopar:cpx`
 1528, 1534, 1705, 1714
`\cs_if_exist:cF` 666, 683
`\cs_if_exist:NT` 2221, 2237
`\cs_if_exist_use:cF` 1049, 1746
`\cs_if_exist_use:cT` 1749
`\cs_if_free:cT` 870, 1825
`\cs_if_free:cTF` 852, 1777
`\cs_if_free:NT` 28, 1889, 1890, 1891, 1892, 2181
`\cs_if_free:NTF` 2182
`\cs_meaning:c` 69
`\cs_new:Nn` 47, 49, 51
`\cs_new_eq:NN`
 1753, 1878, 1886, 1922, 2101, 2113
`\cs_new_nopar:Nn` 102,
 113, 153, 172, 179, 209, 289, 292, 297,
 308, 320, 322, 330, 388, 396, 453, 480,
 487, 496, 499, 505, 511, 523, 538, 546,
 556, 565, 580, 645, 670, 713, 729, 762,
 772, 790, 818, 823, 848, 880, 919, 998,
 1005, 1011, 1024, 1148, 1162, 1214,
 1229, 1255, 1297, 1326, 1380, 1411,
 1432, 1446, 1494, 1547, 1557, 1562,
 1574, 1579, 1584, 1586, 1619, 1643,
 1652, 1659, 1677, 1689, 1732, 1744,
 1754, 1923, 1970, 1979, 2102, 2134, 2158
`\cs_new_nopar:Npn` . 71, 130, 131, 132, 133,
 134, 135, 136, 137, 1440, 2080, 2081, 2099
`\cs_new_protected:Npn` 82, 98, 100
`\cs_new_protected_nopar:Nn`
 .. 111, 586, 991, 1516, 2196, 2201, 2206
`\cs_new_protected_nopar:Npx` 1818
`\cs_set:Npn` 601, 618
`\cs_set:Npx` 85, 90
`\cs_set_eq:NN` 84, 106, 200, 201, 608,
 613, 620, 638, 643, 701, 702, 710, 988,
 990, 1285, 1286, 1294, 1900, 1994,
 1995, 2086, 2119, 2169, 2182, 2225, 2226
`\cs_set_nopar:cpn` 2246
`\cs_set_nopar:Npn`
 2170, 2227, 2232, 2239, 2246, 2247
`\cs_set_nopar:Nx` 2186
`\cs_set_protected_nopar:Npn` 2114
`\cs_to_str:N` 1815

D

`\DeclareSymbolFont` 1934
`\def` 105, 153, 173, 174, 175
`\defaultCJKfontfeatures` 6, 1836, 1837, 1839
`\dim_add:Nc` 1020, 1167, 1169
`\dim_add:Nn` 1021, 1095, 1102, 1103, 1236
`\dim_compare:nNnT` 1180, 1196
`\dim_compare:nNnTF` 1060, 1077
`\dim_compare_p:nNn` 415
`\dim_const:cn` 583
`\dim_eval:c` 483, 491, 497, 498
`\dim_eval:n` 492, 493, 498, 1238
`\dim_gset:cn` 583
`\dim_gset:Nn` 1983
`\dim_if_exist:cTF` 582
`\dim_new:N` 36, 37, 38, 624, 1963
`\dim_set:Nc` 1020,
 1044, 1045, 1156, 1181, 1191, 1198, 1200
`\dim_set:Nn` 1020,
 1047, 1058, 1074, 1079, 1192, 1216, 1257
`\dim_set_eq:NN` 1086, 1107, 1233
`\dim_set_max:Nn` 1110, 1202, 1209, 1211
`\dim_set_min:Nc` 1020, 1199, 1201
`\dim_set_min:Nn` 1022, 1087, 1096, 1104
`\dim_to_fp:n` 1975
`\dim_use:c` 577, 588, 589
`\dim_use:N` 1113, 1115, 1250
`\dim_zero:N` 1039, 1040, 1041, 1152
`\document` 54, 55
`\documentclass` 2, 52, 93, 150, 169

E

`\E` 62
`xeCJKactive` 3
`\else:` 74, 107, 117, 125, 127, 578
`\EmboldenFactor` 1347
`\end` 48, 49, 88, 89,
 90, 145, 146, 147, 166, 180, 191, 784, 785
`\etex_fontcharwd:D` 1238
`\etex_lastnodetype:D` 359, 405, 413, 422, 514
`\etex_protected:D` ... 1896, 1897, 1898, 1899
`\exp_after:wN` .. 106, 108, 116, 118, 723, 1418
`\exp_args:Nc` 155, 1611
`\exp_args:NNc` 722, 723
`\exp_args:Nnnx` 2141, 2143
`\exp_args:No` 722, 1416, 1434
`\exp_args:Nv` 1833
`\exp_args:Nx` 1322, 2011
`\exp_last_unbraced:Nf` 68
`\exp_last_unbraced:NNo` 1436
`\exp_last_unbraced:NNV` 1750

\exp_not:N ...	87, 92, 115, 666, 683, 1532, 1533, 1537, 1538, 1539, 1709, 1710, 1712, 1713, 1717, 1718, 1719, 1818, 1873	\g_xeCJK_exspace_dim	1962, 1963, 1983, 2000, 2001
\exp_not:n	88, 93, 303, 315, 1530, 1535, 1715, 2188	\g_xeCJK_family_name_prop .	1532, 1551, 1674, 1675, 1709, 1712, 1737, 1779, 1932
\exp_not:V	954	\g_xeCJK_family_options_prop	1674, 1676, 1686, 1700, 1703, 1737, 1863
\ExplFileDate	2	\g_xeCJK_family_prop	1317, 1330, 1549, 1674, 1674, 1685, 1698, 1699, 1737, 1842, 1846, 1852, 1854, 1951
\ExplFileDescription	2	\g_xeCJK_family_spec_tl ..	1828, 1830, 1831
\ExplFileName	2	\g_xeCJK_features_id_prop	1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1499, 1585, 1661
F			
\F	61	\g_xeCJK_fontspec_int	1497, 1820, 1821, 1827, 1829
\f@series	1743	\g_xeCJK_indent_bool	2032, 2036, 2084
\f@shape	1743	\g_xeCJK_inline_env_clist	792, 799, 802, 803, 808
\f@size	1743, 1975, 1984	\g_xeCJK_math_bool	1910, 1953
\FallBack	1445	\g_xeCJK_monoscale_fp	1491, 1962, 1962, 1968, 1974
\fi:	74, 109, 119, 127, 128, 578	\g_xeCJK_number_bool	2035, 2243
\file_input:n	2192	\g_xeCJK_punct_width_tl ...	985, 1051, 1053
\font_glyph_if_exist:NnTF	1299	\g_xeCJK_punctstyle_clist	938, 938, 939, 941, 954, 968, 1245
\fontencoding	1537, 1717, 2175	\g_xeCJK_SlantFactor_fp	1346, 1368, 1372, 1481, 1514
\fontfamily	1538, 1718, 1973, 1982, 2175	\g_xeCJK_special_ps_clist	992, 992, 993, 1140
\fontseries	2175	\g_xeCJK_sub_key_clist	1379, 1379, 1382, 1506
\fontshape	2175	\g_xeCJK_unknown_family_seq	1788, 1790, 1796
\fontspec_set_family:Nnn	1530	\group_align_safe_begin:	95
\foo	174, 177	\group_align_safe_end:	87, 92
\fp_gset:Nn	1974	\group_begin:	54, 56, 1328, 1518, 1679, 1844, 1972, 1981, 2167
\fp_new:N	1345, 1346, 1962	\group_end:	54, 65, 1341, 1544, 1687, 1873, 1976, 1985, 2167
\fp_set:Nn	1357, 1368, 1470, 1479	H	
\fp_set_eq:NN	1513, 1514	CheckSingle	4
\fp_use:N	1491, 1577, 1582, 1968	\hline	11, 23, 35, 47
G			
\g_fontspec_encoding_tl	1537, 1717, 1934, 1936	\hrulefill	162, 187
\g_xeCJK_after_end_preamble_hook	46, 52, 55	\hsize	183
\g_xeCJK_after_preamble_hook ...	45, 50, 53	I	
\g_xeCJK_at_end_preamble_hook ..	44, 48, 54	MiddlePunct	5
\g_xeCJK_AutoFakeBold_bool	1343, 1351, 1353, 1356, 1511	MiddlePunct+	5
\g_xeCJK_AutoFakeSlant_bool	1344, 1362, 1364, 1367, 1512	MiddlePunct-	5
\g_xeCJK_base_class_clist	163, 163, 168, 466, 882	\if_catcode:w	115
\g_xeCJK_CJK_class_clist	163, 170, 174, 332, 2137	\if_cs_exist:w	73
\g_xeCJK_CJK_class_prop .	163, 171, 175, 199	\if_dim:w	577
\g_xeCJK_CJK_subclass_clist	816, 817, 899, 901, 911, 1680, 1847	\if_int_compare:w	123, 126, 205, 1918
\g_xeCJK_class_seq	153, 157, 159, 2123, 2129	\if_meaning:w	104
\g_xeCJK_default_features_clist	1524, 1836, 1838	\IfBooleanTF	2024, 2027
\g_xeCJK_EmboldenFactor_fp	1345, 1357, 1371, 1472, 1513		

\IfNoValueTF	217, 218, 1814	\l_xeCJK_AutoFakeBold_bool	1466, 1469, 1511, 1622, 1635
\IfValueT	2185	\l_xeCJK_AutoFakeSlant_bool	1475, 1478, 1512, 1624
\indentfirst	2032	\l_xeCJK_checksingle_bool	695, 698, 700, 707, 709
\InlineEnv	797	\l_xeCJK_CJKfontspec_id_tl	1824, 1825, 1831, 1833
\InlineEnv+	797	\l_xeCJK_current_coor_tl	1748, 1750
\InlineEnv-	797	\l_xeCJK_ecglue_dim	415, 602, 619, 624
\int_compare:nNnF	514, 2122	\l_xeCJK_EmboldenFactor_fp	1470, 1513, 1577
\int_compare:nNnTF	2116	\l_xeCJK_fallback_bool	1279, 1282, 1284, 1291, 1293
\int_compare_p:nNn	359, 405, 413, 422	\l_xeCJK_family_fb_tl	1329, 1330, 1334, 1335, 1336, 1337, 1338, 1339
\int_const:Nn	148, 149, 150, 151, 152	\l_xeCJK_family_tl	1391, 1520, 1527, 1528, 1533, 1534, 1685, 1686, 1694, 1698, 1700, 1707, 1710, 1815, 1816, 1817, 1818
\int_eval:n	1266, 2248	\l_xeCJK_fixed_spacing_bool	1989, 1991, 2008, 2010, 2020
\int_gdecr:N	1821	\l_xeCJK_fixed_xecglue_bool	1993, 2016
\int_gincr:N	1497, 1827	\l_xeCJK_flexible_punctstyle_tl	1992, 2013
\int_incr:N	206, 1919, 2128	\l_xeCJK_font_coor_tl	1023, 1249, 1743, 1743, 1746, 1748
\int_max:nn	218	\l_xeCJK_fontname_bf_tl	1602
\int_min:nn	217	\l_xeCJK_fontname_tl	1330, 1331, 1427, 1428, 1522, 1531, 1685
\int_new:N	33, 34, 35, 1820, 2157	\l_xeCJK_fontoptions_clist	1521, 1682, 1683, 1686
\int_set:Nn	198, 213, 217, 218, 1914, 2136	\l_xeCJK_fontspec_family_tl	1530, 1533, 1538
\int_set_eq:NN	214, 2125	\l_xeCJK_lastpunct_tl	501, 507, 508, 509, 535, 542, 543, 550, 551, 553, 558, 560, 561, 567, 569, 570, 572
\int_to_hexadecimal:n	1318	\l_xeCJK_math_family_tl	1932, 1935, 1937
\int_until_do:nNnn	2126	\l_xeCJK_monoscale_tl	1486, 1491, 1496, 1671, 1672
\int_use:N	1829	\l_xeCJK_pass_features_clist	1504, 1524, 1531, 1665, 1670, 1672
\ItalicFeatures	1446	\l_xeCJK_peek_search_token	81, 84, 115
\ItalicFont	1446	\l_xeCJK_punct_coor_tl	483, 491, 492, 493, 497, 1023, 1023, 1026, 1031, 1032, 1035, 1044, 1046, 1072, 1082, 1112, 1114, 1116, 1125, 1132, 1133, 1134, 1135, 1150, 1159, 1167, 1169, 1181, 1191, 1198, 1199, 1200, 1201, 1219, 1221, 1223, 1224, 1226, 1232, 1235, 1237
\itshape	14, 15, 20, 21, 26, 27, 32, 33, 38, 39, 44, 45, 154, 156	\l_xeCJK_punctstyle_tl	948, 949, 1023, 1028, 1063, 1088, 1153, 1164, 1172, 1188, 1992
J			
\J	61	\l_xeCJK_SlantFactor_fp	1479, 1514, 1582
CJKecglue	4	\l_xeCJK_space_bool	433, 637, 642
CJKglue	3	\l_xeCJK_space_glue_tl	360, 414, 440
CJKmath	3		
CJKnumber	3		
CJKspace	3		
K			
\K	61		
\KaiMingPunct	970		
\keys_define:nn	138, 596, 597, 626, 687, 797, 943, 944, 955, 970, 1271, 1347, 1383, 1456, 1462, 1910, 2033		
\keys_define:nx	943, 953		
\keys_set:nn	2062, 2091		
\keys_set_known:nnN	1378		
\keys_set_known:nxN	1378, 1523		
L			
\l_keys_choice_tl	948		
\l_keys_key_tl	2053, 2054		
\l_keys_module_tl	2059		
\l_keys_value_tl	962, 1357, 1368, 1470, 1479		
\l_peek_token	104, 115, 766, 767		
\l_xeCJK_add_font_prop	1498, 1559, 1564, 1669		

<code>\str_if_eq_x_p:nn</code>	469, 1242, 1616, 1628, 1629	<code>\tl_if_in:nnTF</code>	1322
T		<code>\tl_if_single_token_p:n</code>	722
<code>\T</code>	61	<code>\tl_map_function:nN</code>	62
<code>\t</code>	57	<code>\tl_map_inline:cn</code>	1002, 1008
<code>\TeX</code>	154, 156	<code>\tl_map_inline:nn</code>	61, 285, 1019, 1736, 2162
<code>\tex_afterassignment:D</code>	105	<code>\tl_map_inline:xn</code>	1013, 1122
<code>\tex_char:D</code>	2248	<code>\tl_new:c</code>	76, 77, 78, 79, 995, 1893
<code>\tex_font:D</code>	442, 443, 444, 1238, 1299, 1750, 1975, 1984	<code>\tl_new:N</code>	39, 40, 44, 45, 46, 816
<code>\tex_fontdimen:D</code>	442, 443, 444, 1975, 1984	<code>\tl_put_left:Nn</code>	54, 2159
<code>\tex_ignorespaces:D</code>	55, 464, 1794, 1834, 1876, 2092, 2199	<code>\tl_put_right:cx</code>	1007
<code>\tex_kern:D</code>	497, 588, 589	<code>\tl_put_right:Nn</code>	55, 921, 1304, 1334, 1896, 1897, 1898, 1899, 2021, 2159
<code>\tex_lastkern:D</code>	577	<code>\tl_put_right:Nx</code>	1391, 1763
<code>\tex_lastskip:D</code>	360, 414, 415	<code>\tl_remove_all:cn</code>	1015
<code>\tex_the:D</code>	303, 314, 321	<code>\tl_remove_all:Nn</code>	1759
<code>\tex_unkern:D</code>	650, 675, 2107	<code>\tl_replace_all:Nnx</code>	1428
<code>\tex_unskip:D</code>	363, 418	<code>\tl_set:cn</code>	2180
<code>\tex_vrule:D</code>	482	<code>\tl_set:cx</code>	1000, 1430
<code>\textbf</code>	63, 155, 157	<code>\tl_set:Nn</code>	440, 1023, 1436, 1743, 1889, 1890, 1891, 1892, 2245, 2251, 2252
<code>\textcentereddot</code>	2164	<code>\tl_set:Nx</code>	501, 553, 572, 948, 1329, 1413, 1421, 1425, 1427, 1443, 1491, 1520, 1522, 1694, 1748, 1762, 1786, 1815, 1816, 1824, 1856, 1926, 1929
<code>\textellipsis</code>	2164	<code>\tl_set_eq:NN</code>	1930, 1992
<code>\textemdash</code>	2164	<code>\tl_to_lowercase:n</code>	63
<code>\textit</code>	155, 157	<code>\tl_to_str:N</code>	1881
<code>\textperiodcentered</code>	2164	<code>\tl_use:c</code>	179, 492, 493, 857, 1072, 1082, 1219, 1221, 1223, 1224, 1226, 1653, 1666
<code>\textquotedblleft</code>	2165	<code>\token_get_arg_spec:c</code>	721
<code>\textquotedblright</code>	2165	<code>\token_get_arg_spec:N</code>	813
<code>\textquoteleft</code>	2165	<code>\token_if_cs_p:c</code>	657, 739, 751
<code>\textquoteright</code>	2165	<code>\token_if_cs_p:N</code>	811
<code>\textsl</code>	65	<code>\token_if_eq_meaning_p:NN</code>	766, 767
<code>\tipaencoding</code>	2168	<code>\token_if_math_toggle:cTF</code>	665, 682
<code>\tl_case:Nnn</code>	1063, 1088	<code>\token_if_math_toggle:NTF</code>	815
<code>\tl_clear:c</code>	1501	<code>\token_if_math_toggle_p:c</code>	738, 750
<code>\tl_clear:N</code>	1331, 1486, 1496	<code>\token_if_math_toggle_p:N</code>	814
<code>\tl_const:cn</code>	942	<code>\token_if_other_p:N</code>	723
<code>\tl_const:Nn</code>	1909	<code>\token_if_space_p:c</code>	737, 742
<code>\tl_gput_right:Nn</code>	48, 50, 52	<code>\token_if_space_p:N</code>	812
<code>\tl_gset:cn</code>	1122, 1270	<code>\token_new:Nn</code>	81
<code>\tl_gset:cV</code>	1031, 1032, 1116, 1159, 1232, 1235	<code>\token_to_str:N</code>	25, 864, 1803, 1804, 1805, 1807, 1959
<code>\tl_gset:cx</code>	1112, 1114, 1237, 1249, 1831	<code>\TrimSpaces</code>	180, 868
<code>\tl_gset:Nx</code>	535, 1828	<code>\ttdefault</code>	1973, 1982
<code>\tl_if_empty:xT</code>	2145, 2150	<code>\ttfamily</code>	36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 71, 128, 1899
<code>\tl_if_empty_p:x</code>	721	U	
<code>\tl_if_eq:NNF</code>	1153	<code>\U</code>	62
<code>\tl_if_eq:NNT</code>	949	<code>\UL@hook</code>	2221, 2223
<code>\tl_if_eq:NNTF</code>	1028, 1164, 1172	<code>PunctStyle</code>	4
<code>\tl_if_eq_p:NN</code>	1188		
<code>\tl_if_exist:cF</code>	1026, 1035, 1150		
<code>\tl_if_exist:cTF</code>	1125, 1155		
<code>\tl_if_exist_p:c</code>	1132, 1133, 1134, 1135		
<code>\tl_if_head_eq_charcode:nNTF</code>	1416		
<code>\tl_if_head_group:nT</code>	1434		

PunctWidth	5	\xeCJK_calc_kern:NnNN	1173, 1179, 1195, 1214, 1214
\Unicode	2247	\xeCJK_check_family:n	1547, 1547, 1555
\upshape	12,	\xeCJK_check_family:V	1527, 1697
	13, 18, 19, 24, 25, 30, 31, 36, 37, 42, 43	\xeCJK_check_for_glue:	388, 390, 396
\use:c	461, 1779, 1917, 2250	\xeCJK_check_num_range:nnNN	197, 209, 209, 1913
\use:x	299, 310, 324, 1872	\xeCJK_checksingl:N	702, 713, 713
\use_none:n	201	\xeCJK_checksingl:NN ...	713, 716, 725, 729
\usepackage	3, 53, 54, 55, 94, 95, 151, 170	\xeCJK_checksingl:NNN ..	713, 753, 757, 762
AutoFakeBold	4, 5	\xeCJK_checksingl_env:NNN ..	770, 772, 772
AutoFakeSlant	4, 5	\xeCJK_checksingl_env:NNNN	772, 782, 785, 790
AutoFallBack	4	\xeCJK_checksingl_save_CJKsymbol:N ..	701,
			710, 726, 732, 745, 754, 759, 778, 786, 793
V			
\verb	12, 14, 16, 18, 20, 22,	\xeCJK_CJK_and_CJK:N	452, 453, 453, 701, 702, 710
	24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46	\xeCJK_CJK_and_FullLeft:N	511, 511
\verbatim@font	2021	\xeCJK_CJK_and_FullRight:N	523, 523
X			
\xCJKecglue	597	\xeCJK_CJK_class	148, 149
\xe@alloc@intercharclass	2123, 2126	\xeCJK_class_num:n	179, 179, 188,
\xeCJK@family ...	1302, 1304, 1316, 1317,		286, 290, 295, 301, 304, 312, 314, 315, 321
	1743, 1749, 1759, 1762, 1763, 1764,	\xeCJK_clear_CJK_toks: 330, 330, 344, 392, 459	
	1767, 1768, 1786, 1842, 1846, 1852, 1863	\xeCJK_clear_inter_class_toks:nn ...	292, 292, 333, 343
\xeCJK@setfont	1753	\xeCJK_copy_family:nn	1732, 1732, 1742
\xeCJK_add_fake_bold:n 1574, 1574, 1622, 1639		\xeCJK_copy_family:xx	1767, 1768
\xeCJK_add_fake_slant:n ..	1574, 1579, 1631	\xeCJK_copy_inter_class_toks:nnnn ...	322, 322, 884,
\xeCJK_add_font:nn	1557, 1557, 1567		885, 898, 903, 904, 2139, 2140, 2147, 2152
\xeCJK_add_font:nV	1602	\xeCJK_def_node:nn	580, 580, 591, 592, 593, 594, 595
\xeCJK_add_font:nv	1590	\xeCJK_Default_class	148, 148
\xeCJK_add_font:nx	1647	\xeCJK_erase_CJKsymbol: ..	818, 818, 893, 930
\xeCJK_add_font_if_new:nn	1557, 1562, 1568, 1641	\xeCJK_error:n	130, 131
\xeCJK_add_font_if_new:nv	1594	\xeCJK_error:nx	130, 132, 853, 962, 2053
\xeCJK_add_font_if_new:nx	1649	\xeCJK_fallback_save_CJKsymbol:N ...	1285, 1294, 1300, 1310
\xeCJK_add_special_punct:nn	977, 980, 982, 992, 1005	\xeCJK_fallback_testsymbol:N	1286, 1297, 1297, 1306
\xeCJK_add_sub_block:	1543, 1689, 1689	\xeCJK_family_if_exist:x	1775
\xeCJK_after_FullRight:	472, 505, 505	\xeCJK_family_if_exist:xF	1764
\xeCJK_AfterEndPreamble:n ...	44, 51, 2120	\xeCJK_family_if_exist:xF	1707, 1734
\xeCJK_AfterPreamble:n	31, 44, 49, 2021, 2160, 2219	\xeCJK_family_if_exist:xF	1302, 1766, 1775, 1785, 1925, 1928
\xeCJK_app_inter_class_toks:nnc ..	308, 350	\xeCJK_fix_itcorr:	2102, 2102, 2117
\xeCJK_app_inter_class_toks:nnn	308, 308, 319, 348, 374, 894, 2141	\xeCJK_flexible_cjkg glue:	1995, 2014
\xeCJK_AtEndPreamble:n	44, 47, 1948	\xeCJK_flexible_ecglue:	1994, 2015
\xeCJK_aux_r	2169, 2170	\xeCJK_FullLeft_and_FullLeft:N ..	538, 538
\xeCJK_before_FullLeft:N	499, 499, 516, 544, 563	\xeCJK_FullLeft_and_FullRight:N ..	546, 546
\xeCJK_Boundary_and_CJK:	387, 388, 388, 931, 2211	\xeCJK_FullLeft_class	148, 150
\xeCJK_Boundary_class	148, 152		
\xeCJK_calc_kern:NN	1148, 1157, 1162		

\xeCJK_FullRight_and_FullLeft:N .	556 , 556	\xeCJK_inter_class_toks:nnn	289 , 289 , 291 , 326 , 339 , 346 , 354 , 378 ,
\xeCJK_FullRight_and_FullRight:N	565 , 565		387 , 446 , 452 , 456 , 463 , 464 , 926 , 927 ,
\xeCJK_FullRight_class	148 , 151		928 , 2208 , 2209 , 2210 , 2211 , 2214 , 2215
\xeCJK_get_glyph_bounds:nNN	1231 , 1234 , 1255 , 1255	\xeCJK_itcorr_aux . . .	2108 , 2111 , 2113 , 2117
\xeCJK_get_inter_class_toks:nn	320 , 320 , 327 , 2142 , 2144 , 2145 , 2150	\xeCJK_make_node:n	381 , 382 , 449 , 580 , 586 , 650 , 675 , 2109 , 2230
\xeCJK_get_kern:NN	1148 , 1148 , 1213	\xeCJK_map_features_id:n .	1446 , 1560 , 1565
\xeCJK_get_kern:VN	542 , 550 , 560 , 569	\xeCJK_map_it_sl:n	1633 , 1643 , 1643
\xeCJK_get_map_font:n	1647 , 1649 , 1652 , 1652 , 1656	\xeCJK_math_type:n	1914 , 1922
\xeCJK_get_punct_bounds:nN	513 , 525 , 541 , 549 , 559 , 568 , 1024 , 1024 , 1120	\xeCJK_msg_new:nn . . .	130 , 130 , 861 , 964 , 1314 , 1556 , 1774 , 1797 , 1879 , 1955 , 2057
\xeCJK_get_punct_bounds:nV	507	\xeCJK_new_class:n 153 , 153 , 160 , 161 , 162 , 872	
\xeCJK_get_punct_dimen:N .	1036 , 1229 , 1229	\xeCJK_new_sub_key:n	874 , 1379 , 1380 , 1410 , 1445
\xeCJK_get_sub_features:nn	1335 , 1403 , 1411 , 1432	\xeCJK_NormalSpace_class	160
\xeCJK_get_sub_features_aux:n	1418 , 1432 , 1440	\xeCJK_Others_class	2120 , 2136 , 2157
\xeCJK_gobble_brace:N 1415 , 1432 , 1432 , 1437		\xeCJK_parse_features: . . .	1525 , 1584 , 1584
\xeCJK_HalfLeft_class	160	\xeCJK_parse_features:n . .	1584 , 1585 , 1586
\xeCJK_HalfRight_class	160	\xeCJK_pass_features: . . .	1526 , 1659 , 1659
\xeCJK_if_font_select:n	1609	\xeCJK_patch:Nnn	2158 , 2158 , 2167 , 2168
\xeCJK_if_font_select:nF .	1593 , 1601 , 1648	\xeCJK_peek_catcode_ignore_spaces:Nf	100 , 676
\xeCJK_if_font_select:nTF	1588 , 1609	\xeCJK_peek_catcode_ignore_spaces:NT .	98
\xeCJK_if_font_select_p:n	1609 , 1628 , 1629 , 1637	\xeCJK_peek_catcode_ignore_spaces:Nf	81 , 82 , 99 , 101 , 651 , 744
\xeCJK_if_it_or_sl:n	1614	\xeCJK_peek_execute_branches_catcode:	108 , 113
\xeCJK_if_it_or_sl:nT	1633	\xeCJK_peek_false:w	90 , 118
\xeCJK_if_it_or_sl:nTF	1591 , 1609	\xeCJK_peek_ignore_spaces_execute_branches:	96 , 102 , 112
\xeCJK_if_it_or_sl_p:n . . .	1609 , 1627 , 1636	\xeCJK_peek_ignore_spaces_execute_branches_aux:	105 , 111
\xeCJK_if_last_node:n	575	\xeCJK_peek_math:	638 , 645 , 670
\xeCJK_if_last_node:nT	370 , 649 , 674	\xeCJK_peek_tmp:w	106
\xeCJK_if_last_node:nTF	369 , 575 , 2105	\xeCJK_peek_true:w	85 , 116
\xeCJK_if_last_node_p:n	365 , 399 , 404 , 406 , 421 , 423 , 424 , 430 , 431 , 575	\xeCJK_pre_inter_class_toks:nnn	297 , 297 , 385 , 471 , 472 , 888 , 892 , 905 , 907 , 915 , 2143
\xeCJK_if_map_font_select:n	1654	\xeCJK_punct_glue:n	487 , 487 , 509 , 515 , 520 , 531
\xeCJK_if_map_font_select:nT	1645	\xeCJK_punct_if_both_left_right:NN .	1128
\xeCJK_if_map_font_select:nTF	1654	\xeCJK_punct_if_both_left_right:NNT	1225
\xeCJK_if_map_font_select_p:n	1654	\xeCJK_punct_if_both_left_right:NNTf	1123 , 1175
\xeCJK_if_outer_macro:c	66	\xeCJK_punct_if_both_left_right_p:NN	1123
\xeCJK_if_outer_macro:cF	663 , 680	\xeCJK_punct_if_long:Nf . .	526 , 1037 , 1123
\xeCJK_if_outer_macro:cTF	56	\xeCJK_punct_if_long_p:N .	1123 , 1208 , 1241
\xeCJK_if_outer_macro_aux:w	68 , 71	\xeCJK_punct_if_middle:NT	517 , 529
\xeCJK_if_outer_macro_p:c	56 , 656 , 720 , 736 , 749	\xeCJK_punct_if_middle:Nf . . .	1056 , 1123
\xeCJK_ignore_spaces:	643 , 645 , 645	\xeCJK_punct_if_middle_p:N	1123
\xeCJK_ignorespaces: 450 , 638 , 643 , 2204 , 2230		\xeCJK_punct_if_mixedwidth:NT .	1093 , 1177
\xeCJK_info:nx	130 , 136		
\xeCJK_info:nxx	130 , 137 , 1758		
\xeCJK_inter_class_toks:nnc	289 , 478		

\xeCJK_punct_if_mixedwidth:N	1101, 1123	\xeCJK_UL_CJKsymbol	2225, 2229
\xeCJK_punct_if_mixedwidth_p:N	1123, 1187	\xeCJK_UL_subclass_patch:nn	875, 919, 919
\xeCJK_punct_if_right:N	1123	\xeCJK_UL_subclass_patch_tl	816, 816, 921, 2217
\xeCJK_punct_if_right:N	1168	\xeCJK_ULpostpunctchar:n	2196, 2201, 2215
\xeCJK_punct_if_right:NT	1166	\xeCJK_ULprepunctchar:n	2196, 2196, 2214
\xeCJK_punct_if_right:N	1123, 1220, 1222	\xeCJK_ULroutines:	2196, 2206, 2234
\xeCJK_punct_if_right_p:N	1123	\xeCJK_warning:n	133, 1875, 1952
\xeCJK_punct_kern:n	496, 496, 543, 551, 561, 570	\xeCJK_warning:nx	130, 134, 1309, 1791
\xeCJK_punct_nobreak:	562, 988, 990	\xeCJK_warning:nxx	135, 1552
\xeCJK_punct_rule:n	480, 480, 502, 508, 519, 532, 558, 567	\xeCJK_zero_glue:	988, 991
\xeCJK_reset_at_catcode:	2186, 2193	\xeCJKactive	138
\xeCJK_restore_CJKsymbol:	818, 823, 895, 933	\xeCJKallowbreakbetweenpuncts	950, 973, 987, 987
\xeCJK_save_CJK_class:n	172, 178, 912	\xeCJKCancelSubCJKBlock	10, 118, 831, 832
\xeCJK_save_family_info:	1542, 1677, 1677	\xeCJKcaption	2181, 2183
\xeCJK_select_font:	342, 393, 460, 1036, 1305, 1743, 1744, 1753, 1772, 2229, 2233, 2240	\xeCJKDeclareCharClass	180, 180, 223, 225, 227, 234, 248, 855, 877
\xeCJK_set_catcode:n	200, 201, 203	\xeCJKDeclareSubCharClass	829, 867, 867, 879
\xeCJK_set_fake:n	1607, 1619, 1619	\xeCJKDeclareSubCJKBlock	9, 96, 97, 98, 99, 828, 828, 830, 864
\xeCJK_set_family:ncc	1337, 1727	\xeCJKdisablechecksingle	691, 695, 705
\xeCJK_set_family:nnn	1324, 1516, 1516, 1546, 1811, 1817, 1830, 1885, 1888, 1908, 2025, 2028, 2029	\xeCJKdisablefallback	1275, 1279, 1289
\xeCJK_set_family_fallback:nnn	1320, 1323, 1326	\xeCJKenablechecksingle	690, 695, 696
\xeCJK_set_init:	1494, 1494, 1519	\xeCJKenablefallback	1274, 1279, 1280
\xeCJK_set_mathfont:	1923, 1923, 1953	\xeCJKmainfont	1959
\xeCJK_set_monoexspace:	1485, 1978, 1979, 1996	\xeCJKnobreakbetweenpuncts	974, 987, 989
\xeCJK_set_monoscale:	1490, 1967, 1970	\xeCJKplainchr	2098, 2099
\xeCJK_set_others_toks:n	2120, 2130, 2134	\xeCJKResetPunctClass	10, 221, 221, 282
\xeCJK_set_special_punct:nn	976, 979, 983, 992, 998	\xeCJKRestoreSubCJKBlock	10, 122, 831, 840
\xeCJK_set_sub_class_toks:nn	873, 880, 880	\xeCJKsetcharclass	187, 195, 195
\xeCJK_setcharclass_aux:Nn	180, 187, 192, 1942	\xeCJKsetecglue	2101
\xeCJK_space_or_xecglue:	370, 608, 613, 620, 652, 659, 677	\xeCJKsetemboldenfactor	2094, 2094
\xeCJK_sub_restore_or_cancel:n	837, 845, 848	\xeCJKsetkern	10, 1269, 1269
\xeCJK_sub_special_punct:nn	978, 981, 984, 992, 1011	\xeCJKsetmathcode	1911, 1911, 1942
\xeCJK_switch_font:nn	889, 895, 906, 908, 916, 932, 1743, 1754	\xeCJKsetslantfactor	2094, 2096
\xeCJK_tl_if_blank:x	121	\xeCJKsetup	3, 164, 189, 1997, 2011, 2089, 2089, 2095, 2097, 2098, 2100
\xeCJK_tl_if_blank:x	1671	\xeCJKsetwidth	10, 1121, 1121
\xeCJK_tl_if_blank:x	1419	\XeTeX@CJKfntef@hook	2237, 2239
\xeCJK_tl_if_blank:xTF	121, 213, 1051, 1387, 1426, 1611, 1656	\xetex_if_engine:F	10
\xeCJK_tl_if_blank_p:x	121, 211	\XeTeXcharclass	204, 286
\xeCJK_UL_CJKpunctsymbol	2226, 2233	\XeTeXcharglyph	1266
		\XeTeXdefaultencoding	2185, 2194
		\XeTeXglyphbounds	25, 28, 1259
		\XeTeXinterchartokenstate	145, 146, 2116
		\XeTeXinterchartoks	290, 294, 301, 303, 312, 314, 321
		\XeTeXmathcode	1916
		Y	
		\Y	58, 62

	Z	
$\backslash Z$	57, 62
$\backslash o$	172