

xeCJK 宏包

ctex.org

2012/09/03 v3.0.8

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 _g T _E X 的字体名查找	8
3.3	CJK 分区字体设置	9
3.4	设置 CJK 字符范围	10
3.5	设置标点宽度和间距	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	61
5.16	兼容性修补	62

6	例子	66
6.1	xeCJK-example-autofake.tex	66
6.2	xeCJK-example-fallback.tex	67
6.3	xeCJK-example-subCJKblock.tex	67
6.4	xeCJK-example-CJKecglue.tex	69
6.5	xeCJK-example-checksingle.tex	70

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</code> \star	<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</code> \star	<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</code> \star	<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 = { \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/> CJKecglue <hr/>	<p>CJKecglue = {⟨glue⟩}</p> <p>设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距,默认值是一个空格。使用这个选项设置的 glue 最好也要用一定的弹性。请注意,这里设置的 glue 只影响 xeCJK 根据需要自动添加的空白,源文件中直接输入的 CJK 文字与西文之间的空格不受影响(直接输出)。有时候 xeCJK 可能不能正确地调整间距,需要手动加空格。</p>
<hr/> xCJKecglue <hr/>	<p>xCJKecglue = {⟨true false glue⟩}</p> <p>缺省状态下,xeCJK 不对源文件中直接输入的 CJK 文字与西文之间的空格进行调整,如果需要调整,请使用这个选项。如果使用这个选项,将使用 CJKecglue 替换源文件中直接输入的 CJK 文字与西文之间的空格。</p>
<hr/> CheckSingle <hr/>	<p>CheckSingle = ⟨true false⟩</p> <p>是否避免单个 CJK 文字单独占一个段落的最后一行。</p>
<hr/> InlineEnv InlineEnv+ InlineEnv- <hr/>	<p>InlineEnv = {env1, env2, env3, ...}</p> <p>在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 L^AT_EX 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^AT_EX 环境没有造成断行,可以使用这个选项来声明它,以便 CheckSingle 能正确识别。</p>
<hr/> AutoFallBack <hr/>	<p>AutoFallBack = ⟨true false⟩</p> <p>当文档中有个别生僻字时,可以使用这个选项,自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。</p>
<hr/> AutoFakeBold ☆ <hr/>	<p>AutoFakeBold = {⟨true false 数字⟩}</p> <p>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。</p>
<hr/> AutoFakeSlant ☆ <hr/>	<p>AutoFakeSlant = {⟨true false 数字⟩}</p> <p>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。</p>
<hr/> EmboldenFactor ☆ <hr/>	<p>EmboldenFactor = {⟨数字 4⟩}</p> <p>设置伪粗体的默认粗细程度。</p>
<hr/> SlantFactor ☆ <hr/>	<p>SlantFactor = {⟨数字 0.167⟩}</p> <p>设置伪斜体的粗细程度,范围是 -0.999 ~ 0.999。</p>
<hr/> PunctStyle <hr/>	<p>PunctStyle = {⟨<i>quanjiao</i> <i>banjiao</i> <i>kaiming</i> <i>hangmoban</i><i>quanjiao</i> CCT plain⟩}</p> <p>设置标点处理格式,有效值分别为</p> <p>quanjiao 全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;</p> <p>banjiao 半角式: 所有标点占半个汉字宽度;</p> <p>kaiming 开明式: 句末点号用全角,其他半角;</p> <p>hangmobanquanjiao 行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</p> <p>CCT CCT 格式;</p> <p>plain 原样(不调整标点间距)。</p>

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-	★	

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-	★	

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-	★	

PunctWidth	★	PunctWidth = {⟨dimen⟩}
------------	---	------------------------

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`。

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

\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。

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。

\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。

\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 <code>CJK</code> 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 (<code>verbatim</code>) 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 <code>CJK</code> 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 <code>CJK</code> 字体族 <code><family></code> 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 <code>CJK</code> 字体族,<code><family></code> 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 <code>CJK</code> 字体族 <code><family></code> 并指定字体,并定义 <code>\<{font-switch}></code>,在文档中可以使用它来切换 <code>CJK</code> 字体族。可以不必指定 <code><family></code>,这时候 <code><family></code> 将等于 <code><{font-switch}></code>。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 <code>CJK</code> 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 <code>CJK</code> 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 <code>CJK</code> 字体缩小为 0.962216。<code>xeCJK</code> 宏包的初始化设置是</p> <pre> \defaultCJKfontfeatures{Script=CJK} </pre>

<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,Εντρονα 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 đâđ,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Εντρονα,Bold,Negrita,
Lihavoitu,Gras,Félkövéř,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
Полужирный,Fet,Kalın,Krepko,đâđ,Lodia
Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,
thườđg,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* {<block>} {<block range>}
\xeCJKDeclareSubCJKBlock* {<block>} {<block range>}
```

其中 `<block range>` 是逗号列表, 可以是 CJK 字符的 Unicode 范围, 也可以是单个字符的 Unicode。例如

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

的形式。需要注意的是, 这里设置的 `<block range>` 除非确实需要(例如某些特殊字体使用了 Unicode 中的私人使用区的情况), 否则不要超出源代码中预设的 **CJK 文字范围**。使用

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

就声明了 `SPUA` 和 `Ext-B` 这两个子分区。同时在 3.2 节介绍的 CJK 字体设置命令的 `` 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。带星号的命令除了设置 CJK 子分区以外, 还重置标点符号所属的字符类。

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

在文档中取消对 CJK 分区的声明。带星号的命令还重置标点符号所属的字符类。

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

在文档中恢复对 CJK 分区的声明。带星号的命令还重置标点符号所属的字符类。

3.4 设置 CJK 字符范围

<code>\xeCJKDeclareCharClass</code> ★	<code>\xeCJKDeclareCharClass {<class>} {<class range>}</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。带星号的命令除了设置字符类别以外, 为了确保标点处理的正确性, 还重置标点符号所属的字符类。

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

用于重置标点符号所属的字符类。

<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 xejk 自动禁止在它之后载入 CJK 宏包。

可以在 xejk 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 xejk 代码实现

```
1 <{*package>
```

5.1 运行环境检查

xeCJK 必须使用 XeTeX 引擎的支持。

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

应该使用较新版本的 expl3 宏包。

```
10 \msg_new:nnn { xejk } { l3-too-old }
11 {
12   Support~package~'expl3'~too~old. \\
13   Please~update~an~up~to~date~version~of~the~bundles\\
14   'l3kernel'~and~'l3packages'\\
15   using~your~TeX~package~manager~or~from~CTAN.\\
16   Loading~xeCJK~will~abort!
17 }
18 \@ifpackagelater { expl3 } { 2012/07/16 } { }
19 { \msg_critical:nn { xejk } { l3-too-old } }
```

```
20 \RequirePackage{xparse}
21 \RequirePackage{l3keys2e}
```

\XeTeXglyphbouds 可以得到一个字符的左右边距,用于标点压缩。

```
22 \msg_new:nnn { xejk } { XeTeX-too-old }
23 {
24   \token_to_str:N \XeTeXglyphbouds\c_space_tl is~not~defined.\\
25   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
26 }
27 \cs_if_free:NT \XeTeXglyphbouds
28 {
29   \msg_error:nn { xejk } { XeTeX-too-old }
30   \AtEndOfPackage { \xeCJK_AfterPreamble:n { \punctstyle { plain } } }
31 }
```

5.2 内部工具

分配临时变量。

```
\l_xeCJK_tmpa_int
\l_xeCJK_tmppb_int 32 \int_new:N \l_xeCJK_tmpa_int
\l_xeCJK_tmppc_int 33 \int_new:N \l_xeCJK_tmppb_int
34 \int_new:N \l_xeCJK_tmppc_int
(End definition for \l_xeCJK_tmpa_int, \l_xeCJK_tmppb_int, and \l_xeCJK_tmppc_int)

\l_xeCJK_tmpa_dim
\l_xeCJK_tmppb_dim 35 \dim_new:N \l_xeCJK_tmpa_dim
\l_xeCJK_tmppc_dim 36 \dim_new:N \l_xeCJK_tmppb_dim
37 \dim_new:N \l_xeCJK_tmppc_dim
(End definition for \l_xeCJK_tmpa_dim, \l_xeCJK_tmppb_dim, and \l_xeCJK_tmppc_dim)

\l_xeCJK_tmpa_tl
\l_xeCJK_tmppb_tl 38 \tl_new:N \l_xeCJK_tmpa_tl
39 \tl_new:N \l_xeCJK_tmppb_tl
(End definition for \l_xeCJK_tmpa_tl and \l_xeCJK_tmppb_tl)

\l_xeCJK_tmpa_clist
\l_xeCJK_tmppb_clist 40 \clist_new:N \l_xeCJK_tmpa_clist
\l_xeCJK_tmppc_clist 41 \clist_new:N \l_xeCJK_tmppb_clist
42 \clist_new:N \l_xeCJK_tmppc_clist
(End definition for \l_xeCJK_tmpa_clist, \l_xeCJK_tmppb_clist, and \l_xeCJK_tmppc_clist)

\XeCJK_no_break:
43 \cs_new_protected_nopar:Npn \XeCJK_no_break: { \tex_penalty:D \c_ten_thousand }
(End definition for \XeCJK_no_break:)

\XeCJK_AtEndPreamble:n 在 \document 前后加上各种钩子。
\XeCJK_AfterPreamble:n 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_protected:Npn \XeCJK_AtEndPreamble:n #1
48 { \tl_gput_right:Nn \g_xeCJK_at_end_preamble_hook {#1} }
49 \cs_new_protected:Npn \XeCJK_AfterPreamble:n #1
50 { \tl_gput_right:Nn \g_xeCJK_after_preamble_hook {#1} }
51 \cs_new_protected:Npn \XeCJK_AfterEndPreamble:n #1
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:Npnn \xeCJK_if_outer_macro:c #1 { 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 l3token 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   \tl_set:Nn \xeCJK_peek_true:w { \group_align_safe_end: #2 }
86   \tl_set:Nn \xeCJK_peek_false:w { \group_align_safe_end: #3 }
87   \group_align_safe_begin:
88   \peek_after:Nw \xeCJK_peek_ignore_spaces_execute_branches:
89 }
90 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NT #1#2
91 { \xeCJK_peek_catcode_ignore_spaces:NTF #1 {#2} { } }
92 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NF #1#2
93 { \xeCJK_peek_catcode_ignore_spaces:NTF #1 { } {#2} }
94 \cs_new_nopar:Npn \xeCJK_peek_ignore_spaces_execute_branches:
95 {
96   \if_meaning:w \l_peek_token \c_space_token
97   \tex_afterassignment:D \xeCJK_peek_ignore_spaces_execute_branches_aux:
98   \exp_after:wN \cs_set_eq:NN \exp_after:wN \xeCJK_peek_tmp:w
99   \else:
100     \exp_after:wN \xeCJK_peek_execute_branches_catcode:
101     \fi:
102 }
103 \cs_new_protected_nopar:Npn \xeCJK_peek_ignore_spaces_execute_branches_aux:
104 { \peek_after:Nw \xeCJK_peek_ignore_spaces_execute_branches: }
105 \cs_new_nopar:Npn \xeCJK_peek_execute_branches_catcode:
106 {
107   \if_catcode:w \exp_not:N \l_peek_token \exp_not:N \l_xeCJK_peek_search_token
108   \exp_after:wN \xeCJK_peek_true:w
109   \else:

```

```

110     \exp_after:wN \xeCJK_peek_false:w
111     \fi:
112 }
(End definition for \xeCJK_peek_catcode_ignore_spaces:NTF)

```

\xeCJK_if_blank_x:p:n 判断是否为空或者仅含一个空格。

```

\xeCJK_if_blank_x:nTF 113 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
114 {
115     \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
116     \prg_return_true:
117     \else:
118     \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
119     \prg_return_true: \else: \prg_return_false: \fi:
120     \fi:
121 }
(End definition for \xeCJK_if_blank_x:n)

```

\xeCJK_if_package_loaded_p:n 判断宏包是否载入,可用于文档中。

```

\xeCJK_if_package_loaded:nTF 122 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
123 {
124     \tl_if_exist:cTF { ver@ #1 . \c_xeCJK_pkg_extension_tl }
125     { \prg_return_true: } { \prg_return_false: }
126 }
127 \tl_const:Nx \c_xeCJK_pkg_extension_tl { \@pkgextension }
(End definition for \xeCJK_if_package_loaded:n)

```

\xeCJK_msg_new:nn 各种信息函数的缩略形式。

```

\xeCJK_error:n 128 \cs_new_nopar:Npn \xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\xeCJK_error:nx 129 \cs_new_nopar:Npn \xeCJK_error:n { \msg_error:nn { xeCJK } }
\xeCJK_warning:nx 130 \cs_new_nopar:Npn \xeCJK_error:nx { \msg_error:nnx { xeCJK } }
\xeCJK_info:nx 131 \cs_new_nopar:Npn \xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\xeCJK_info:nxx 132 \cs_new_nopar:Npn \xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
133 \cs_new_nopar:Npn \xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
134 \cs_new_nopar:Npn \xeCJK_info:nx { \msg_info:nnx { xeCJK } }
135 \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

```

136 \keys_define:nn { xeCJK / options }
137 {
138     xeCJKactive .choice:,
139     xeCJKactive / true .code:n = { \makexeCJKactive } ,
140     xeCJKactive / false .code:n = { \makexeCJKinactive } ,
141     xeCJKactive .default:n = { true },
142 }

```

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

\makexeCJKactive

```

143 \NewDocumentCommand \makexeCJKactive { } { \XeTeXinterchartokenstate = \c_one }
144 \NewDocumentCommand \makexeCJKinactive { } { \XeTeXinterchartokenstate = \c_zero }

```

(End definition for \makexeCJKactive and \makexeCJKinactive)

抑制 BOM。

145 \char_set_catcode_ignore:n { "FEFF }

5.4 字符类别设定

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

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

\xeCJK_Default_class 这五类是 Xe_{La}TeX 预定义的类别。

```

\xeCJK_CJK_class      146 \int_const:Nn \xeCJK_Default_class { 0 }
\xeCJK_FullLeft_class 147 \int_const:Nn \xeCJK_CJK_class { 1 }
\xeCJK_FullRight_class 148 \int_const:Nn \xeCJK_FullLeft_class { 2 }
\xeCJK_Boundary_class 149 \int_const:Nn \xeCJK_FullRight_class { 3 }
                      150 \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 151 \cs_new_protected_nopar:Npn \xeCJK_new_class:n #1
                    152 {
                    153   \exp_args:Nc \newXeTeXintercharclass { xeCJK_#1_class }
                    154   \clist_gclear_new:c { g_xeCJK_#1_range_clist }
                    155   \seq_gput_right:Nv \g_xeCJK_class_seq { xeCJK_#1_class }
                    156 }
                    157 \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 158 \xeCJK_new_class:n { HalfLeft }
\xeCJK_NormalSpace_class 159 \xeCJK_new_class:n { HalfRight }
                        160 \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 161 \clist_set:Nn \g_xeCJK_base_class_clist
\g_xeCJK_CJK_class_prop 162 {
                        163   Default, CJK, FullLeft, FullRight, Boundary,
                        164   HalfLeft, HalfRight, NormalSpace
                        165 }
                        166 \clist_map_inline:Nn \g_xeCJK_base_class_clist

```

```

167 { \clist_gclear_new:c { g_xeCJK_#1_range_clist } }
168 \clist_new:N \g_xeCJK_CJK_class_clist
169 \prop_new:N \g_xeCJK_CJK_class_prop
170 \cs_new_protected_nopar:Npn \xeCJK_save_CJK_class:n #1
171 {
172   \clist_gput_right:Nx \g_xeCJK_CJK_class_clist {#1}
173   \prop_gput:Nvx \g_xeCJK_CJK_class_prop { xeCJK_#1_class } {#1}
174 }
175 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
176 \clist_map_function:nN { CJK, FullLeft, FullRight } \xeCJK_save_CJK_class:n
(End definition for \g_xeCJK_base_class_clist, \g_xeCJK_CJK_class_clist, and \g_xeCJK_CJK_class_prop)

```

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

```

177 \cs_new_nopar:Npn \xeCJK_class_num:n #1 { \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_set_char_class_aux:Nn`

```

178 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
179 {
180   \xeCJK_declare_char_class:nn {#2} {#3}
181   \IfBooleanT {#1} { \xeCJKResetPunctClass }
182 }
183 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
184 {
185   \clist_gput_right:cx { g_xeCJK_#1_range_clist } {#2}
186   \clist_map_inline:xn {#2}
187   {
188     \str_if_eq_x:nnF {##1} { -> }
189     {
190       \xeCJK_set_char_class_aux:Nn \xeCJK_set_char_class:nnn {##1}
191       { \xeCJK_class_num:n {#1} }
192     }
193   }
194 }
195 \NewDocumentCommand \xeCJK_set_char_class_aux:Nn
196 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
197 \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。

```

198 \NewDocumentCommand \xeCJKsetcharclass { s m m m }
199 {
200   \xeCJK_set_char_class:nnn {#2} {#3} {#4}
201   \IfBooleanF {#1} { \xeCJKResetPunctClass }
202 }
203 \cs_new_protected_nopar:Npn \xeCJK_set_char_class:nnn #1#2#3
204 {
205   \xeCJK_check_num_range:nnNN {#1} {#2} \l_xeCJK_tmpe_int \l_xeCJK_tmpe_int
206   \int_set:Nn \l_xeCJK_tmpe_int {#3}
207   \prop_if_in:NVTF \g_xeCJK_CJK_class_prop \l_xeCJK_tmpe_int
208   { \loop \char_set_catcode_other:n \l_xeCJK_tmpe_int }

```



```

209     { \loop }
210     \XeTeXcharclass \l_xeCJK_tmpa_int = \l_xeCJK_tmpe_int
211     \if_int_compare:w \l_xeCJK_tmpa_int < \l_xeCJK_tmpe_int
212         \int_incr:N \l_xeCJK_tmpa_int
213     \repeat
214 }
(End definition for \xeCJKsetcharclass)

```

\xeCJK_check_num_range:nnNN

```

215 \cs_new_protected_nopar:Npn \xeCJK_check_num_range:nnNN #1#2#3#4
216 {
217     \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
218     {
219         \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
220         \int_set_eq:NN #3 #4
221     }
222     {
223         \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
224         \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
225     }
226 }
(End definition for \xeCJK_check_num_range:nnNN)

```

\xeCJKResetPunctClass 用于重置标点符号所属的字符类。

```

227 \NewDocumentCommand \xeCJKResetPunctClass { }
228 {
229     \xeCJK_declare_char_class:nn { HalfLeft }
230     { "28 , "2D , "5B , "60 , "7B }
231     \xeCJK_declare_char_class:nn { HalfRight }
232     { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , }
233     \xeCJK_declare_char_class:nn { FullLeft }
234     {
235         "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
236         "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
237         "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
238         "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6 ,
239     }
240     \xeCJK_declare_char_class:nn { FullRight }
241     {
242         "00B7 , "2019 , "201D , "2014 , "2025 , "2026 , "2030 , "2500 , "3001 ,
243         "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 , "3015 ,
244         "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 , "3047 ,
245         "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C , "309D ,
246         "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
247         "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE , "FE50 ,
248         "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C , "FE5E ,
249         "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B , "FF1F ,
250         "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 , "FF69 ,
251         "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E , "FF9F ,
252     }
253 }

```

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

设置 CJK 文字的范围。

```
254 \xeCJK_declare_char_class:nn { CJK }
```

```
255 {
```

- Hangul Jamo (谚文字母)

```
256 "1100 -> "11FF ,
```

- CJK Radicals Supplement (中日韩部首补充)

```
257 "2E80 -> "2EFF ,
```

- Kangxi Radicals (康熙字典部首)

```
258 "2F00 -> "2FDF ,
```

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

```
259 "2FF0 -> "2FFF ,
```

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

```
260 "3000 -> "303F ,
```

- Hiragana (日文平假名)

```
261 "3040 -> "309F ,
```

- Katakana (日文片假名)

```
262 "30A0 -> "30FF ,
```

- Bopomofo (注音字母)

```
263 "3100 -> "312F ,
```

- Hangul Compatibility Jamo (谚文兼容字母)

```
264 "3130 -> "318F ,
```

- Kanbun (象形字注释标志)

```
265 "3190 -> "319F ,
```

- Bopomofo Extended (注音字母扩展)

```
266 "31A0 -> "31BF ,
```

- CJK Strokes (中日韩笔画)

```
267 "31C0 -> "31EF ,
```

- Katakana Phonetic Extensions (日文片假名语音扩展)

```
268 "31F0 -> "31FF ,
```

- Enclosed CJK Letters and Months (带圈中日韩字母和月份)

```
269 "3200 -> "32FF ,
```

- CJK Compatibility (中日韩兼容)

```
270 "3300 -> "33FF ,
```

- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
271 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
272 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
273 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
274 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
275 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
276 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
277 "AC00 -> "D7AF ,
- Kana Supplement (日文假名增补)
278 "B000 -> "B0FF ,
- Hangul Jamo Extended-B (谚文扩展 B)
279 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
280 "F900 -> "FAFF ,
- CJK Compatibility Forms (中日韩兼容形式)
281 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)
282 "FF00 -> "FFEF ,
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)
283 "20000 -> "2A6DF ,
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)
284 "2A700 -> "2B73F ,
- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)
285 "2B740 -> "2B81F ,
- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)
286 "2F800 -> "2FA1F ,

287 }
重置标点符号的字符类。

288 \xeCJKResetPunctClass

\normalspacedchars 声明前后不加间距的字符。

289 \NewDocumentCommand \normalspacedchars { m }
290 {
291 \tl_map_inline:nn {#1}
292 { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
293 }
294 \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。

\xeCJK_inter_class_toks:nnc 295 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
\xeCJK_inter_class_toks:nxx 296 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = {#3} }
297 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc , nxx }
(End definition for \xeCJK_inter_class_toks:nnn, \xeCJK_inter_class_toks:nnc, and \xeCJK_inter_class_toks:nxx)

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

298 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2
299 { \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} }
(End definition for \xeCJK_get_inter_class_toks:nn)

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

300 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2
301 { \xeCJK_inter_class_toks:nnn {#1} {#2} { \c_empty_tl } }
(End definition for \xeCJK_clear_inter_class_toks:nn)

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

302 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
303 {
304 \xeCJK_inter_class_toks:nxx {#1} {#2}
305 { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
306 }
(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。

\xeCJK_app_inter_class_toks:nnc 307 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
308 {
309 \xeCJK_inter_class_toks:nxx {#1} {#2}
310 { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
311 }
312 \cs_generate_variant:Nn \xeCJK_app_inter_class_toks:nnn { nnc }
(End definition for \xeCJK_app_inter_class_toks:nnn and \xeCJK_app_inter_class_toks:nnc)

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

313 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
314 {
315 \xeCJK_inter_class_toks:nxx {#1} {#2}
316 { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
317 }

(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。

```

318 \cs_new_protected_nopar:Npn \xeCJK_clear_CJK_toks:
319 {
320   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
321     { \xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
322 }
(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	✓	✓	✓	✓	✓		✓	

```

323 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
324 {
325   \clist_map_inline:nn { CJK, FullLeft, FullRight }
326   {
327     \xeCJK_inter_class_toks:nnn {#1} {##1}
328     {
329       \c_group_begin_token
330       \xeCJK_select_font:
331       \xeCJK_clear_inter_class_toks:nn {#1} {##1}
332       \xeCJK_clear_CJK_toks:
333     }
334     \xeCJK_inter_class_toks:nnn {##1} {#1} { \c_group_end_token }
335   }
336   \xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
337   \clist_map_inline:nn { FullLeft, FullRight }
338   { \xeCJK_app_inter_class_toks:nnc {#1} {##1} { xeCJK_CJK_and_##1:N } }
339 }
340 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
341 {
342   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
343   {
344     \bool_if:nTF
345     {
346       \l_xeCJK_xecglue_bool &&
347       \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
348       \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl
349     }
350     {
351       \tex_unskip:D

```

```

352         \bool_if:nTF
353         { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK_space } }
354         { \CJKecglue } { \c_space_token }
355     }
356     {
357         \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
358         { \xeCJK_if_last_node:nT { CJK_space } { \xeCJK_space_or_xecglue: } }
359     }
360 }
361 \str_if_eq_x:nnF {#1} { NormalSpace }
362 { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
363 }
364 \clist_map_inline:nn { Default, HalfRight, NormalSpace }
365 {
366     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
367     {
368         \peek_catcode:NTF \c_space_token
369         { { \xeCJK_make_node:n { default_space } } }
370         { { \xeCJK_make_node:n { default } } }
371     }
372     \str_if_eq_x:nnF {#1} { NormalSpace }
373     { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
374 }
375 \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }

\xeCJK_Boundary_and_CJK:
\xeCJK_check_for_glue: 376 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_CJK:
377 {
378     { \xeCJK_check_for_glue: }
379     \c_group_begin_token
380     \xeCJK_clear_CJK_toks:
381     \xeCJK_select_font:
382     \CJKsymbol
383 }
384 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
385 {
386     \bool_if:nTF
387     { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK_space } }
388     { \CJKglue }
389     {
390         \bool_if:nTF
391         {
392             \xeCJK_if_last_node_p:n { default } ||
393             \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
394             \xeCJK_if_last_node_p:n { default_itcorr }
395         }
396         { \CJKecglue }
397         {
398             \bool_if:nT
399             {
400                 \l_xeCJK_xecglue_bool &&
401                 \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&

```

```

402      ( \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl ||
403        \dim_compare_p:nNn \tex_lastskip:D = \l_xeCJK_ecglue_dim )
404    }
405    {
406      \tex_unskip:D
407      \bool_if:nTF
408        {
409          \xeCJK_if_last_node_p:n { default_space }      ||
410          \int_compare_p:nNn \tex_lastnodetype:D = \c_ten ||
411          \xeCJK_if_last_node_p:n { default }            ||
412          \xeCJK_if_last_node_p:n { default_itcorr }
413        }
414        { \CJKecglue }
415        {
416          \bool_if:nTF
417            {
418              \xeCJK_if_last_node_p:n { CJK }      ||
419              \xeCJK_if_last_node_p:n { CJK_space }
420            }
421            { \bool_if:NTF \l_xeCJK_space_bool \c_space_token \CJKglue }
422            { \c_space_token }
423          }
424        }
425      }
426    }
427  }
428  \tl_set:Nn \l_xeCJK_space_glue_tl
429  {
430    \tex_fontdimen:D \c_two \tex_font:D
431    \@plus \tex_fontdimen:D \c_three \tex_font:D
432    \@minus \tex_fontdimen:D \c_four \tex_font:D
433  }
434  (End definition for \xeCJK_Boundary_and_CJK: and \xeCJK_check_for_glue:)
435  \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
436  {
437    \c_group_end_token
438    { \xeCJK_make_node:n { CJK } }
439    \xeCJK_ignorespaces:
440  }
441  \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }

```

\xeCJK_CJK_and_CJK:N

```

441  \cs_new_protected_nopar:Npn \xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }
442  (End definition for \xeCJK_CJK_and_CJK:N)
443  \clist_map_inline:nn { FullLeft, FullRight }
444  {
445    \xeCJK_inter_class_toks:nnn { Boundary } {#1}
446    {
447      \c_group_begin_token
448      \xeCJK_clear_CJK_toks:
449      \xeCJK_select_font:

```

```

449     \use:c { xeCJK_CJK_and_#1:N }
450   }
451   \xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
452   \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \c_group_end_token \tex_ignorespaces:D }
453 }
454 \clist_map_inline:Nn \g_xeCJK_base_class_clist
455 {
456   \bool_if:nF
457     { \str_if_eq_x_p:nn {#1} { FullLeft } || \str_if_eq_x_p:nn {#1} { FullRight } }
458     {
459       \xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \xeCJK_no_break: }
460       \xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \xeCJK_after_FullRight: }
461     }
462 }
463 \clist_map_inline:nn { CJK, FullLeft, FullRight }
464 {
465   \clist_map_inline:nn { FullLeft, FullRight }
466     { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
467 }

```

`\xeCJK_punct_rule:n` 用于抹去标点符号的左/右空白。

```

468 \cs_new_protected_nopar:Npn \xeCJK_punct_rule:n #1
469 {
470   \tex_vrule:D
471   \@width \dim_eval:c { \l_xeCJK_punct_coor_tl/rule/#1 }
472   \@depth \c_zero_dim
473   \@height \c_zero_dim \scan_stop:
474 }
(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。

```

475 \cs_new_protected_nopar:Npn \xeCJK_punct_glue:n #1
476 {
477   \skip_horizontal:n
478   {
479     \dim_eval:c { \l_xeCJK_punct_coor_tl/glue/#1 }
480     \@plus \dim_eval:n { ( \tl_use:c { \l_xeCJK_punct_coor_tl/glue/#1 } ) /4 }
481     \@minus \dim_eval:n { ( \tl_use:c { \l_xeCJK_punct_coor_tl/glue/#1 } ) /2 }
482   }
483 }
(End definition for \xeCJK_punct_glue:n)

```

`\xeCJK_punct_kern:n` 相邻两个标点之间的间距。

```

484 \cs_new_protected_nopar:Npn \xeCJK_punct_kern:n #1
485 { \tex_kern:D \dim_eval:c { \l_xeCJK_punct_coor_tl/kern/#1 } }
(End definition for \xeCJK_punct_kern:n)

```

`\dim_eval:c`

```

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

```


\xeCJK_before_FullLeft:N

```
487 \cs_new_protected_nopar:Npn \xeCJK_before_FullLeft:N #1
488 {
489   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
490   \xeCJK_punct_rule:n { 1/#1 }
491   \CJKpunctsymbol {#1}
492 }
(End definition for \xeCJK_before_FullLeft:N)
```

\xeCJK_after_FullRight:

```
493 \cs_new_protected_nopar:Npn \xeCJK_after_FullRight:
494 {
495   \xeCJK_get_punct_bounds:nV { r } \l_xeCJK_lastpunct_tl
496   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
497   \xeCJK_punct_glue:n { r/\l_xeCJK_lastpunct_tl }
498 }
(End definition for \xeCJK_after_FullRight:)
```

\xeCJK_CJK_and_FullLeft:N

```
499 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
500 {
501   \xeCJK_get_punct_bounds:nN { l } {#1}
502   \int_compare:nNf \etex_lastnodetype:D = \c_one
503     { \xeCJK_punct_glue:n { 1/#1 } }
504   \xeCJK_before_FullLeft:N {#1}
505   \xeCJK_punct_if_middle:NT {#1}
506   {
507     \xeCJK_punct_rule:n { m/#1 }
508     \xeCJK_punct_glue:n { 1/#1 }
509   }
510 }
(End definition for \xeCJK_CJK_and_FullLeft:N)
```

\xeCJK_CJK_and_FullRight:N

```
511 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
512 {
513   \xeCJK_get_punct_bounds:nN { r } {#1}
514   \xeCJK_punct_if_long:NTF {#1} \CJKglue
515   {
516     \xeCJK_no_break:
517     \xeCJK_punct_if_middle:NT {#1}
518     {
519       \xeCJK_punct_glue:n { r/#1 }
520       \xeCJK_punct_rule:n { m/#1 }
521     }
522   }
523   \tl_gset:Nx \l_xeCJK_lastpunct_tl {#1}
524   \CJKpunctsymbol {#1}
525 }
(End definition for \xeCJK_CJK_and_FullRight:N)
```

\xeCJK_FullLeft_and_FullLeft:N

```
526 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
527 {
528   \xeCJK_no_break:
529   \xeCJK_get_punct_bounds:nN { l } {#1}
530   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
531   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
532   \xeCJK_before_FullLeft:N {#1}
533 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N)
```

\xeCJK_FullLeft_and_FullRight:N

```
534 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
535 {
536   \xeCJK_no_break:
537   \xeCJK_get_punct_bounds:nN { r } {#1}
538   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
539   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
540   \xeCJK_no_break:
541   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
542   \CJKpunctsymbol {#1}
543 }
(End definition for \xeCJK_FullLeft_and_FullRight:N)
```

\xeCJK_FullRight_and_FullLeft:N

```
544 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
545 {
546   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
547   \xeCJK_get_punct_bounds:nN { l } {#1}
548   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
549   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
550   \xeCJK_punct_nobreak:
551   \xeCJK_before_FullLeft:N {#1}
552 }
(End definition for \xeCJK_FullRight_and_FullLeft:N)
```

\xeCJK_FullRight_and_FullRight:N

```
553 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
554 {
555   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
556   \xeCJK_get_punct_bounds:nN { r } {#1}
557   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
558   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
559   \xeCJK_no_break:
560   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
561   \CJKpunctsymbol {#1}
562 }
(End definition for \xeCJK_FullRight_and_FullRight:N)
```

```

\xeCJK_if_last_node:p:n
\xeCJK_if_last_node:nTF 563 \prg_new_conditional:Nppn \xeCJK_if_last_node:n #1 { p , T , F , TF }
564 {
565   \if_dim:w \dim_use:c { c_xeCJK_#1_node_dim } = \tex_lastkern:D
566   \prg_return_true: \else: \prg_return_false: \fi:
567 }
(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 568 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
569 {
570   \dim_if_exist:cTF { c_xeCJK_#1_node_dim }
571   { \dim_gset:cn } { \dim_const:cn }
572   { c_xeCJK_#1_node_dim } { #2 }
573 }
574 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
575 {
576   \tex_kern:D - \dim_use:c { c_xeCJK_#1_node_dim }
577   \tex_kern:D \dim_use:c { c_xeCJK_#1_node_dim }
578 }
579 \xeCJK_def_node:nn { CJK } { 11 sp }
580 \xeCJK_def_node:nn { CJK_space } { 12 sp }
581 \xeCJK_def_node:nn { default } { 13 sp }
582 \xeCJK_def_node:nn { default_space } { 14 sp }
583 \xeCJK_def_node:nn { default_itscorr } { 15 sp }
(End definition for \xeCJK_def_node:nn and \xeCJK_make_node:n)

```

CJKglue CJK 文字之间插入的 glue。

```

584 \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 585 \keys_define:nn { xeCJK / options }
586 {
587   CJKecglue .code:n =
588   {
589     \cs_set:Npn \CJKecglue { #1 }
590     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
591   },
592   xCJKecglue .choice:,
593   xCJKecglue / true .code:n =
594   {
595     \bool_set_true:N \l_xeCJK_xecglue_bool
596     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
597   },
598   xCJKecglue / false .code:n =
599   {
600     \bool_set_false:N \l_xeCJK_xecglue_bool
601     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_token
602   },
603   xCJKecglue / unknown .code:n =
604   {

```

```

605     \bool_set_true:N \l_xeCJK_xecglue_bool
606     \cs_set:Npn \CJKecglue {#1}
607     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
608     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
609   },
610   xCJKecglue .default:n = { true } ,
611 }
612 \dim_new:N \l_xeCJK_ecglue_dim
613 \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。

```

614 \keys_define:nn { xeCJK / options }
615 {
616   CJKspace .choice:,
617   CJKspace / true .code:n = { \CJKspace },
618   CJKspace / false .code:n = { \CJKnospace },
619   CJKspace .default:n = { true },
620   space .meta:n = { CJKspace = true },
621   nospace .meta:n = { CJKspace = false },
622 }
(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 623 \NewDocumentCommand \CJKspace { }
624 {
625   \bool_set_true:N \l_xeCJK_space_bool
626   \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_peek_math:
627 }
628 \NewDocumentCommand \CJKnospace { }
629 {
630   \bool_set_false:N \l_xeCJK_space_bool
631   \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_ignore_spaces:
632 }
(End definition for \CJKspace and \CJKnospace)

```

\xeCJK_ignore_spaces:

```

\xeCJK_peek_math: 633 \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:
634 {
635   \peek_catcode:NTF \c_space_token
636   {
637     \xeCJK_if_last_node:nT { CJK }
638     { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK_space } } }
639     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
640     { \xeCJK_space_or_xecglue: }
641     {
642       \bool_if:nT
643       {
644         \xeCJK_if_outer_macro_p:c { l_peek_token } ||
645         \token_if_cs_p:c { l_peek_token }
646       }
647       { \xeCJK_space_or_xecglue: }

```

```

648     }
649   }
650   {
651     \xeCJK_if_outer_macro:cF { l_peek_token }
652     {
653       \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
654       { \cs_if_exist:cF { l_peek_token } \exp_not:N }
655     }
656   }
657 }
658 \cs_new_protected_nopar:Npn \xeCJK_peek_math:
659 {
660   \peek_catcode:NTF \c_space_token
661   {
662     \xeCJK_if_last_node:nT { CJK }
663     { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK_space } } }
664     \xeCJK_peek_catcode_ignore_spaces:NF \c_space_token
665     { \xeCJK_space_or_xecglue: }
666   }
667   {
668     \xeCJK_if_outer_macro:cF { l_peek_token }
669     {
670       \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
671       { \cs_if_exist:cF { l_peek_token } \exp_not:N }
672     }
673   }
674 }
(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。

```

675 \keys_define:nn { xeCJK / options }
676 {
677   CheckSingle .choice:,
678   CheckSingle / true .code:n = { \xeCJKenablechecksingle },
679   CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
680   CheckSingle .default:n = { true },
681   CJKchecksingle .meta:n = { CheckSingle = true } ,
682 }
(End definition for CheckSingle This function is documented on page 4.)

```

\xeCJKenablechecksingle 孤字处理功能开关。

```

\xeCJKdisablechecksingle \bool_new:N \l_xeCJK_checksingl_bool
683 \NewDocumentCommand \xeCJKenablechecksingle { }
684 {
685   \bool_if:NF \l_xeCJK_checksingl_bool
686   {
687     \bool_set_true:N \l_xeCJK_checksingl_bool
688     \cs_set_eq:NN \xeCJK_checksingl_save_CJKsymbol:N \xeCJK_CJK_and_CJK:N
689     \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingl:N
690   }
691 }

```

```

692 }
693 \NewDocumentCommand \xeCJKdisablechecksingle { }
694 {
695   \bool_if:NT \l_xeCJK_checksingle_bool
696   {
697     \bool_set_false:N \l_xeCJK_checksingle_bool
698     \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle_save_CJKsymbol:N
699   }
700 }

```

(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle)

```

\xeCJK_checksingle:N
\xeCJK_checksingle:NN 701 \cs_new_protected_nopar:Npn \xeCJK_checksingle:N #1
\xeCJK_checksingle:NNN 702 {
703   \peek_catcode:NTF \c_catcode_other_token
704   { \xeCJK_checksingle:NN {#1} }
705   {
706     \bool_if:nTF
707     {
708       ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } )      &&
709       \xeCJK_if_blank_x_p:n { \token_get_arg_spec:c { l_peek_token } } &&
710       \exp_args:NNc \exp_args:No \tl_if_single_token_p:n { l_peek_token } &&
711       \exp_args:NNc \exp_after:wN \token_if_other_p:N { l_peek_token }
712     }
713     { \xeCJK_checksingle:NN {#1} }
714     { \xeCJK_checksingle_save_CJKsymbol:N {#1} }
715   }
716 }
717 \cs_new_protected_nopar:Npn \xeCJK_checksingle:NN #1#2
718 {
719   \peek_catcode:NTF \c_catcode_other_token
720   { \xeCJK_checksingle_save_CJKsymbol:N {#1} #2 }
721   {
722     \bool_if:nTF
723     {
724       ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
725       ( \token_if_space_p:c { l_peek_token } ||
726         \token_if_math_toggle_p:c { l_peek_token } ||
727         \token_if_cs_p:c { l_peek_token } )
728     }
729     {
730       \bool_if:nTF { \token_if_space_p:c { l_peek_token } }
731       {
732         \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_other_token
733         { \xeCJK_checksingle_save_CJKsymbol:N {#1} #2 \c_space_token }
734         {
735           \bool_if:nTF
736           {
737             ! ( \xeCJK_if_outer_macro_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         { \xeCJK_checksingl:NNN {#1} {#2} \c_space_token }
742         { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
743     }
744 }
745 { \xeCJK_checksingl:NNN {#1} {#2} { } }
746 }
747 { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
748 }
749 }
750 \cs_new_protected_nopar:Npn \xeCJK_checksingl:NNN #1#2#3
751 {
752     \bool_if:nTF
753     {
754         \token_if_eq_meaning_p:NN \l_peek_token \par ||
755         \token_if_eq_meaning_p:NN \l_peek_token \[ % \]
756     }
757     { \CJKsymbol {#1} \xeCJK_no_break: #2 }
758     { \xeCJK_checksingl_env:NNN {#1} {#2} {#3} }
759 }

```

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

\xeCJK_checksingl_env:NNN

\xeCJK_checksingl_env:NNNNn

```

760 \cs_new_protected_nopar:Npn \xeCJK_checksingl_env:NNN #1#2#3
761 {
762     \peek_catcode_remove:NTF \c_math_toggle_token
763     {
764         \peek_catcode:NTF \c_math_toggle_token
765         { \CJKsymbol {#1} \xeCJK_no_break: #2 \c_math_toggle_token }
766         { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 \c_math_toggle_token }
767     }
768     {
769         \peek_meaning_remove:NTF \begin
770         { \xeCJK_checksingl_env:NNNNn {#1} {#2} {#3} \begin }
771         {
772             \peek_meaning_remove:NTF \end
773             { \xeCJK_checksingl_env:NNNNn {#1} {#2} {#3} \end }
774             { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 }
775         }
776     }
777 }
778 \cs_new_protected_nopar:Npn \xeCJK_checksingl_env:NNNNn #1#2#3#4#5
779 {
780     \clist_if_in:NnTF \g_xeCJK_inline_env_clist {#5}
781     { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 }
782     { \CJKsymbol {#1} \xeCJK_no_break: #2 }
783     \scan_stop: #4 {#5}
784 }

```

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

InlineEnv

InlineEnv+ 785 \keys_define:nn { xeCJK / options }

InlineEnv- 786 {

```

787     InlineEnv .clist_gset:N = \g_xeCJK_inline_env_clist ,
788     InlineEnv+ .code:n =
789     {
790         \clist_gput_right:Nx \g_xeCJK_inline_env_clist {#1}
791         \clist_gremove_duplicates:N \g_xeCJK_inline_env_clist
792     },
793     InlineEnv- .code:n =
794     {
795         \clist_map_inline:xn {#1}
796         { \clist_gremove_all:Nn \g_xeCJK_inline_env_clist {##1} }
797     },
798 }

```

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

```

799 \cs_generate_variant:Nn \token_if_cs_p:N { c }
800 \cs_generate_variant:Nn \token_if_space_p:N { c }
801 \cs_generate_variant:Nn \token_get_arg_spec:N { c }
802 \cs_generate_variant:Nn \token_if_math_toggle_p:N { c }
803 \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
804 \tl_new:N \XeCJK_UL_subclass_patch_tl
805 \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:
806 \cs_new_protected_nopar:Npn \XeCJK_erase_CJKsymbol:
807 {
808     \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
809     \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
810 }
811 \cs_new_protected_nopar:Npn \XeCJK_restore_CJKsymbol:
812 {
813     \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block
814     \CJKsymbol
815 }

```

(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。

```

816 \NewDocumentCommand \XeCJKDeclareSubCJKBlock { s m m }
817 {
818     \XeCJKDeclareSubCharClass { CJK } {#2} {#3}
819     \IfBooleanT {#1} { \XeCJKResetPunctClass }
820 }
821 \@onlypreamble \XeCJKDeclareSubCJKBlock

```

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

\XeJKCancelSubCJKBlock 取消和恢复对 CJK 子区的声明。

```

\XeCJKRestoreSubCJKBlock
822 \bool_new:N \l_xeCJK_sub_cancel_bool
823 \NewDocumentCommand \XeJKCancelSubCJKBlock { s m }

```



```

824 {
825   \bool_if:NF \l_xeCJK_sub_cancel_bool
826   {
827     \bool_set_true:N \l_xeCJK_sub_cancel_bool
828     \xeCJK_sub_restore_or_cancel:n {#2}
829     \IfBooleanT {#1} { \xeCJKResetPunctClass }
830   }
831 }
832 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
833 {
834   \bool_if:NT \l_xeCJK_sub_cancel_bool
835   {
836     \bool_set_false:N \l_xeCJK_sub_cancel_bool
837     \xeCJK_sub_restore_or_cancel:n {#2}
838     \IfBooleanT {#1} { \xeCJKResetPunctClass }
839   }
840 }
841 \cs_new_protected_nopar:Npn \xeCJK_sub_restore_or_cancel:n #1
842 {
843   \clist_map_inline:xn {#1}
844   {
845     \cs_if_free:cTF { xeCJK_ CJK/##1 _class }
846     { \xeCJK_error:nx { SubBlock-undefined } {##1} }
847     {
848       \xeCJK_declare_char_class:nn
849       { CJK \bool_if:NF \l_xeCJK_sub_cancel_bool { /##1 } }
850       { \tl_use:c { g_xeCJK_CJK/##1_range_clist } }
851     }
852   }
853 }
854 \xeCJK_msg_new:nn { SubBlock-undefined }
855 {
856   The~CJK~sub~block~'#1'~is~undefined.\\\
857   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \c_space_tl
858   to~declare~it.
859 }

```

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

\xeCJKDeclareSubCharClass

```

860 \NewDocumentCommand \xeCJKDeclareSubCharClass
861 { > { \TrimSpaces } m > { \TrimSpaces } m m }
862 {
863   \cs_if_free:cT { xeCJK_ #1/#2 _class }
864   {
865     \xeCJK_new_class:n { #1/#2 }
866     \xeCJK_set_sub_class_toks:nn {#1} {#2}
867     \xeCJK_new_sub_key:n {#2}
868     \xeCJK_UL_subclass_patch:nn {#1} {#2}
869   }
870   \xeCJK_declare_char_class:nn { #1/#2 } {#3}
871 }
872 \@onlypreamble \xeCJKDeclareSubCharClass

```

(End definition for \xeCJKDeclareSubCharClass)

\xeCJK_set_sub_class_toks:nn

```
873 \cs_new_protected_nopar:Npn \xeCJK_set_sub_class_toks:nn #1#2
874 {
875   \clist_map_inline:Nn \g_xeCJK_base_class_clist
876   {
877     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
878     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
879     \str_if_eq_x:nnTF {##1} { CJK }
880     {
881       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
882       { \xeCJK_switch_font:nn {#1} {#2} }
883     }
884     {
885       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
886       { \xeCJK_erase_CJKsymbol: }
887       \xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
888       { \xeCJK_switch_font:nn {#1} {#2} \xeCJK_restore_CJKsymbol: }
889     }
890   }
891   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
892   \clist_if_empty:NF \g_xeCJK_CJK_subclass_clist
893   {
894     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
895     {
896       \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
897       \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
898       \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
899       { \xeCJK_switch_font:nn {#2} {##1} }
900       \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
901       { \xeCJK_switch_font:nn {##1} {#2} }
902     }
903   }
904   \clist_gput_right:Nn \g_xeCJK_CJK_subclass_clist {#2}
905   \xeCJK_save_CJK_class:n { #1/#2 }
906   \clist_map_inline:nn { CJK, FullLeft, FullRight }
907   {
908     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
909     { \xeCJK_switch_font:nn {#2} {#1} }
910   }
911 }
```

(End definition for \xeCJK_set_sub_class_toks:nn)

\xeCJK_UL_subclass_patch:nn

```
912 \cs_new_protected_nopar:Npn \xeCJK_UL_subclass_patch:nn #1#2
913 {
914   \tl_put_right:Nn \xeCJK_UL_subclass_patch_tl
915   {
916     \clist_map_inline:nn { Default, HalfLeft, HalfRight }
917     {
918       \str_if_eq_x:nnTF {##1} { HalfLeft }
```

```

919         { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKecglue \CJKsymbol } }
920         { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
921     \xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
922     {
923         \xeCJK_erase_CJKsymbol:
924         \xeCJK_Boundary_and_CJK:
925         \xeCJK_switch_font:nn {#1} {#2}
926         \xeCJK_restore_CJKsymbol:
927     }
928 }
929 }
930 }
(End definition for \xeCJK_UL_subclass_patch:nn)

```

5.8 标点处理

`\g_xeCJK_punctstyle_clist` 标点处理方式。

```

931 \clist_new:N \g_xeCJK_punctstyle_clist
932 \clist_set:Nn \g_xeCJK_punctstyle_clist
933 { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
934 \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
935 { \tl_const:cn { c_xeCJK_ps_#1_tl } {#1} }
(End definition for \g_xeCJK_punctstyle_clist)

```

`\keys_define:nx`

```

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

```

PunctStyle

```

937 \keys_define:nn { xeCJK / options }
938 {
939     PunctStyle .choice_code:n =
940     {
941         \tl_set:Nx \l_xeCJK_punctstyle_tl { \l_keys_choice_tl }
942         \tl_if_eq:NNT \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
943         { \xeCJKallowbreakbetweenpuncts }
944     },
945 }
946 \keys_define:nx { xeCJK / options }
947 { PunctStyle .generate_choices:n = { \exp_not:V \g_xeCJK_punctstyle_clist } }
948 \keys_define:nn { xeCJK / options }
949 {
950     PunctStyle / banjiao .meta:n = { PunctStyle = halfwidth },
951     PunctStyle / quanjiao .meta:n = { PunctStyle = fullwidth },
952     PunctStyle / kaiming .meta:n = { PunctStyle = mixedwidth },
953     PunctStyle / hangmobanjiao .meta:n = { PunctStyle = marginkerning },
954     PunctStyle / unknown .code:n =
955     { \xeCJK_error:nx { PunctStyle-undefined } \l_keys_value_tl },
956 }
957 \xeCJK_msg_new:nn { PunctStyle-undefined }
958 {
959     Punctstyle~"~#1~"~is~not~available. \\\

```

```

960     You~can~only~use~one~of\\
961     "\g_xeCJK_punctstyle_clist".\\
962   }

```

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

AllowBreakBetweenPuncts 相关选项声明。

```

KaiMingPunct 963 \keys_define:nn { xeCJK / options }
LongPunct    964 {
MiddlePunct  965   AllowBreakBetweenPuncts .choice:,
PunctWidth   966   AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
              967   AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
              968   AllowBreakBetweenPuncts .default:n = { true },
              969   KaiMingPunct .code:n = { \xeCJK_set_special_punct:nn { mixedwidth } {#1} },
              970   KaiMingPunct+ .code:n = { \xeCJK_add_special_punct:nn { mixedwidth } {#1} },
              971   KaiMingPunct- .code:n = { \xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
              972   LongPunct .code:n = { \xeCJK_set_special_punct:nn { long } {#1} },
              973   LongPunct+ .code:n = { \xeCJK_add_special_punct:nn { long } {#1} },
              974   LongPunct- .code:n = { \xeCJK_sub_special_punct:nn { long } {#1} },
              975   MiddlePunct+ .code:n = { \xeCJK_add_special_punct:nn { middle } {#1} },
              976   MiddlePunct .code:n = { \xeCJK_set_special_punct:nn { middle } {#1} },
              977   MiddlePunct- .code:n = { \xeCJK_sub_special_punct:nn { middle } {#1} },
              978   PunctWidth .tl_gset:N = \g_xeCJK_punct_width_tl ,
              979 }

```

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

```

\xeCJKallowbreakbetweenpuncts
\xeCJKnobreakbetweenpuncts 980 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
                             981 { \cs_set_eq:NN \xeCJK_punct_nobreak: \xeCJK_zero_glue: }
                             982 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
                             983 { \cs_set_eq:NN \xeCJK_punct_nobreak: \xeCJK_no_break: }
                             984 \cs_new_protected_nopar:Npn \xeCJK_zero_glue: { \skip_horizontal:N \c_zero_skip }

```

(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts)

```

\g_xeCJK_special_ps_clist
\xeCJK_set_special_punct:nn 985 \clist_set:Nn \g_xeCJK_special_ps_clist { mixedwidth , long , middle }
\xeCJK_add_special_punct:nn 986 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
\xeCJK_sub_special_punct:nn 987 {
                             988   \tl_new:c { l_xeCJK_#1_punct_tl }
                             989   \prop_new:c { l_xeCJK_#1_punct_prop }
                             990 }
                             991 \cs_new_protected_nopar:Npn \xeCJK_set_special_punct:nn #1#2
                             992 {
                             993   \tl_set:cx { l_xeCJK_#1_punct_tl } {#2}
                             994   \prop_clear:c { l_xeCJK_#1_punct_prop }
                             995   \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
                             996     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
                             997 }
                             998 \cs_new_protected_nopar:Npn \xeCJK_add_special_punct:nn #1#2
                             999 {
1000   \tl_put_right:cx { l_xeCJK_#1_punct_tl } {#2}
1001   \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
1002     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }

```

```

1003 }
1004 \cs_new_protected_nopar:Npn \xeCJK_sub_special_punct:nn #1#2
1005 {
1006   \tl_map_inline:xn {#2}
1007   {
1008     \tl_remove_all:cn { l_xeCJK_#1_punct_tl } {##1}
1009     \prop_remove:cn { l_xeCJK_#1_punct_prop } {##1}
1010   }
1011 }
1012 \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 1013 \cs_generate_variant:Nn \dim_set:Nn { Nc }
\dim_set_min:Nc 1014 \cs_generate_variant:Nn \dim_add:Nn { Nc }
1015 \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_coortl` 记录当前字体状态下的标点处理方式。

```

1016 \tl_set:Nn \l_xeCJK_punct_coortl { \l_xeCJK_font_coortl/\l_xeCJK_punctstyle_tl }
(End definition for \l_xeCJK_punct_coortl)

```

`\xeCJK_get_punct_bounds:nN` #1 为 l/r, #2 为标点字符, 返回标点的实际左/右空白的负值和根据标点处理方式决定添加的左/右空白。全角方式直接添加标点的实际左/右空白; 半角方式添加标点左右空白中的最小值; 开明方式与半角相同, 只在开明标点再增加实际左/右空白的一半。

```

1017 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:nN #1#2
1018 {
1019   \tl_if_exist:cF { \l_xeCJK_punct_coortl/rule/#1/#2 }
1020   {
1021     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1022     {
1023       \clist_map_inline:nn { glue , rule , bound }
1024       { \tl_gset:cV { \l_xeCJK_punct_coortl/##1/#1/#2 } \c_zero_dim }
1025       \tl_gset:cV { \l_xeCJK_punct_coortl/rule/m/#2 } \c_zero_dim
1026     }
1027     {
1028       \tl_if_exist:cF { \l_xeCJK_punct_coortl/bound/#1/#2 }
1029       { { \xeCJK_select_font: \xeCJK_get_punct_dimen:N {#2} } }
1030       \xeCJK_punct_if_long:NTF {#2}
1031       {
1032         \dim_zero:N \l_xeCJK_tmpa_dim
1033         \dim_zero:N \l_xeCJK_tmppb_dim
1034         \dim_zero:N \l_xeCJK_tmppc_dim
1035       }
1036       {
1037         \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coortl/bound/#1/#2 }
1038         \dim_set:Nc \l_xeCJK_tmppb_dim
1039         { \l_xeCJK_punct_coortl/bound/\str_if_eq:x:nnTF {#1} lrl/#2 }
1040         \dim_set:Nn \l_xeCJK_tmppc_dim
1041         {
1042           \cs_if_exist_use:cF { g_xeCJK_punct_width/#2 }
1043           {

```

```

1044         \xeCJK_if_blank_x:nTF \g_xeCJK_punct_width_tl
1045         { \c_zero_dim }
1046         { \g_xeCJK_punct_width_tl }
1047     }
1048 }
1049 \xeCJK_punct_if_middle:NTF {#2}
1050 {
1051     \dim_set:Nn \l_xeCJK_tmpc_dim
1052     {
1053         \dim_compare:nNnTF \l_xeCJK_tmpc_dim > \c_zero_dim
1054         { \l_xeCJK_tmpc_dim }
1055         {
1056             \tl_case:Nnn \l_xeCJK_punctstyle_tl
1057             {
1058                 \c_xeCJK_ps_halfwidth_tl { .5 em }
1059                 \c_xeCJK_ps_mixedwidth_tl { .5 em }
1060                 \c_xeCJK_ps_CCT_tl { .7 em }
1061                 \c_xeCJK_ps_fullwidth_tl { 1 em }
1062             }
1063             { 1 em }
1064         }
1065         - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/#2 }
1066     }
1067     \dim_set:Nn \l_xeCJK_tmpc_dim { .5 \l_xeCJK_tmpc_dim }
1068 }
1069 {
1070     \dim_compare:nNnTF \l_xeCJK_tmpc_dim > \c_zero_dim
1071     {
1072         \dim_set:Nn \l_xeCJK_tmpc_dim
1073         {
1074             \l_xeCJK_tmpc_dim - \l_xeCJK_tmpb_dim
1075             - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/#2 }
1076         }
1077     }
1078     {
1079         \dim_set_eq:NN \l_xeCJK_tmpc_dim \l_xeCJK_tmpa_dim
1080         \dim_set_min:Nn \l_xeCJK_tmpc_dim \l_xeCJK_tmpb_dim
1081         \tl_case:Nnn \l_xeCJK_punctstyle_tl
1082         {
1083             \c_xeCJK_ps_halfwidth_tl { \prg_do_nothing: }
1084             \c_xeCJK_ps_mixedwidth_tl
1085             {
1086                 \xeCJK_punct_if_mixedwidth:NT {#2}
1087                 {
1088                     \dim_add:Nn \l_xeCJK_tmpc_dim { .5 \l_xeCJK_tmpa_dim }
1089                     \dim_set_min:Nn \l_xeCJK_tmpc_dim \l_xeCJK_tmpa_dim
1090                 }
1091             }
1092             \c_xeCJK_ps_CCT_tl
1093             {
1094                 \xeCJK_punct_if_mixedwidth:NTF {#2}
1095                 { \dim_add:Nn \l_xeCJK_tmpc_dim { .5 \l_xeCJK_tmpa_dim } }

```

```

1096             { \dim_add:Nn \l_xeCJK_tmpr_dim { .3 \l_xeCJK_tmpr_dim } }
1097             \dim_set_min:Nn \l_xeCJK_tmpr_dim \l_xeCJK_tmpr_dim
1098         }
1099     }
1100     { \dim_set_eq:NN \l_xeCJK_tmpr_dim \l_xeCJK_tmpr_dim }
1101 }
1102 }
1103 \dim_set_max:Nn \l_xeCJK_tmpr_dim \c_zero_dim
1104 }
1105 \tl_gset:cx { \l_xeCJK_punct_coortl/rule/#1/#2 }
1106 { - \dim_use:N \l_xeCJK_tmpr_dim }
1107 \tl_gset:cx { \l_xeCJK_punct_coortl/rule/m/#2 }
1108 { - \dim_use:N \l_xeCJK_tmpr_dim }
1109 \tl_gset:cV { \l_xeCJK_punct_coortl/glue/#1/#2 } \l_xeCJK_tmpr_dim
1110 }
1111 }
1112 }
1113 \cs_generate_variant:Nn \xeCJK_get_punct_bounds:nN { nV }
(End definition for \xeCJK_get_punct_bounds:nN)

```

\xeCJKsetwidth

```

1114 \NewDocumentCommand \xeCJKsetwidth { m m }
1115 { \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 1116 \prg_new_conditional:Npnn \xeCJK_punct_if_right:N #1 { p , T , F , TF }
\xeCJK_punct_if_long_p:N 1117 {
\xeCJK_punct_if_middle_p:N 1118 \tl_if_exist:cTF { \l_xeCJK_punct_coortl/glue/r/#1 }
\xeCJK_punct_if_mixedwidth_p:N 1119 \prg_return_true: \prg_return_false:
\xeCJK_punct_if_right:NNTF 1120 }
\xeCJK_punct_if_both_left_right:NNTF 1121 \prg_new_conditional:Npnn \xeCJK_punct_if_both_left_right:NN #1#2 { p , T , F , TF }
\xeCJK_punct_if_long:NNTF 1122 {
\xeCJK_punct_if_middle:NNTF 1123 \bool_if:nTF
\xeCJK_punct_if_mixedwidth:NNTF 1124 {
1125 ( \tl_if_exist_p:c { \l_xeCJK_punct_coortl/glue/r/#1 } &&
1126 \tl_if_exist_p:c { \l_xeCJK_punct_coortl/glue/r/#2 } ) ||
1127 ( \tl_if_exist_p:c { \l_xeCJK_punct_coortl/glue/l/#1 } &&
1128 \tl_if_exist_p:c { \l_xeCJK_punct_coortl/glue/l/#2 } )
1129 }
1130 \prg_return_true: \prg_return_false:
1131 }
1132 \cs_generate_variant:Nn \prg_new_conditional:Npnn { c }
1133 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
1134 {
1135 \prg_new_conditional:cpnn { xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1136 {
1137 \prop_if_in:cnTF { l_xeCJK_#1_punct_prop } {##1}
1138 \prg_return_true: \prg_return_false:
1139 }
1140 }
(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。

```

1141 \cs_new_protected_nopar:Npn \xeCJK_get_kern:NN #1#2
1142 {
1143   \tl_if_exist:cF { \l_xeCJK_punct_coors_tl/kern/#1-#2 }
1144   {
1145     \dim_zero:N \l_xeCJK_tmpa_dim
1146     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1147     {
1148       \tl_if_exist:cTF { g_xeCJK_punct/kern/#1-#2 }
1149       { \dim_set:Nc \l_xeCJK_tmpa_dim { g_xeCJK_punct/kern/#1-#2 } }
1150       { \xeCJK_calc_kern:NN {#1} {#2} }
1151     }
1152     \tl_gset:cV { \l_xeCJK_punct_coors_tl/kern/#1-#2 } \l_xeCJK_tmpa_dim
1153   }
1154 }
1155 \cs_new_protected_nopar:Npn \xeCJK_calc_kern:NN #1#2
1156 {
1157   \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_marginkerning_tl
1158   {
1159     \xeCJK_punct_if_right:NT {#1}
1160     { \dim_add:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coors_tl/bound/r/#1 } }
1161     \xeCJK_punct_if_right:NF {#2}
1162     { \dim_add:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coors_tl/bound/l/#2 } }
1163   }
1164   {
1165     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_fullwidth_tl
1166     { \xeCJK_calc_kern:NnNN \l_xeCJK_tmpa_dim { 1.5 em } {#1} {#2} }
1167     {
1168       \xeCJK_punct_if_both_left_right:NNTF {#1} {#2}
1169       {
1170         \xeCJK_punct_if_mixedwidth:NT {#1}
1171         {
1172           \xeCJK_calc_kern:NnNN \l_xeCJK_tmpa_dim { 1 em } {#1} {#2}
1173           \dim_compare:nNnT \l_xeCJK_tmpa_dim < { .1 em }
1174           { \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coors_tl/bound/l/#1 } }
1175         }
1176       }
1177       {
1178         \bool_if:nTF
1179         {
1180           \xeCJK_punct_if_mixedwidth_p:N {#1} &&
1181           ! ( \tl_if_eq_p:NN \l_xeCJK_punctstyle_tl \c_xeCJK_ps_halfwidth_tl )
1182         }
1183         {
1184           \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coors_tl/glue/r/#1 }
1185           \dim_set:Nn \l_xeCJK_tmpa_dim { .7 \l_xeCJK_tmpa_dim }
1186         }
1187         {
1188           \xeCJK_calc_kern:NnNN \l_xeCJK_tmpa_dim { 1 em } {#1} {#2}
1189           \dim_compare:nNnT \l_xeCJK_tmpa_dim < { .1 em }
1190           {

```



```

1191         \dim_set:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#1 }
1192         \dim_set_min:Nc \l_xeCJK_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/r/#1 }
1193         \dim_set:Nc \l_xeCJK_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/l/#2 }
1194         \dim_set_min:Nc \l_xeCJK_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/r/#2 }
1195         \dim_set_max:Nn \l_xeCJK_tmpa_dim \l_xeCJK_tmpb_dim
1196     }
1197 }
1198 }
1199 }
1200 \bool_if:nT
1201 { \xeCJK_punct_if_long_p:N {#1} || \xeCJK_punct_if_long_p:N {#2} }
1202 { \dim_set_max:Nn \l_xeCJK_tmpa_dim { .1 em } }
1203 }
1204 \dim_set_max:Nn \l_xeCJK_tmpa_dim \c_zero_dim
1205 }
1206 \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。

```

1207 \cs_new_protected_nopar:Npn \xeCJK_calc_kern:NnNN #1#2#3#4
1208 {
1209     \dim_set:Nn #1
1210     {
1211         #2
1212         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1213             \xeCJK_punct_if_right:NTF {#3} { bound } { glue } /l/#3 }
1214         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1215             \xeCJK_punct_if_right:NTF {#4} { glue } { bound } /r/#4 }
1216         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#3 }
1217         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#4 }
1218         \xeCJK_punct_if_both_left_right:NNT {#3} {#4}
1219         { + \tl_use:c { \l_xeCJK_punct_coord_tl/rule/m/#4 } }
1220     }
1221 }
(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。

```

1222 \cs_new_protected_nopar:Npn \xeCJK_get_punct_dimen:N #1
1223 {
1224     \xeCJK_get_glyph_bounds:nNN { left } {#1} \l_xeCJK_tmpa_dim
1225     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/l/#1 } \l_xeCJK_tmpa_dim
1226     \dim_set_eq:Nn \l_xeCJK_tmpb_dim \l_xeCJK_tmpa_dim
1227     \xeCJK_get_glyph_bounds:nNN { right } {#1} \l_xeCJK_tmpa_dim
1228     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/r/#1 } \l_xeCJK_tmpa_dim
1229     \dim_add:Nn \l_xeCJK_tmpb_dim \l_xeCJK_tmpa_dim
1230     \tl_gset:cx { \l_xeCJK_punct_coord_tl/dimen/#1 }
1231     { \dim_eval:n { \etex_fontcharwd:D \tex_font:D `#1 - \l_xeCJK_tmpb_dim } }
1232     \bool_if:nT
1233     {
1234         \xeCJK_punct_if_long_p:N {#1} &&

```

```

1235      ! ( \str_if_eq_x_p:nn {#1} { ... } || \str_if_eq_x_p:nn {#1} { .. } )
1236    }
1237  {
1238    \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
1239      {
1240        \str_if_eq_x:nnF \c_xeCJK_ps_plain_tl {##1}
1241          {
1242            \tl_gset:cx { \l_xeCJK_font_coors_tl/##1/kern/#1-#1 }
1243              { - \dim_use:N \l_xeCJK_tmpr_dim }
1244          }
1245      }
1246  }
1247 }
(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。

```

1248 \cs_new_protected_nopar:Npn \xeCJK_get_glyph_bounds:nNN #1#2#3
1249 {
1250   \dim_set:Nn #3
1251   {
1252     \XeTeXglyphbounds
1253     \str_case_x:nnn {#1}
1254       {
1255         { left } { \c_one }    { top } { \c_two }
1256         { right } { \c_three } { bottom } { \c_four }
1257       }
1258       { \c_three }
1259     \int_eval:n { \XeTeXcharglyph `#2 }
1260   }
1261 }
(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。

```

1262 \NewDocumentCommand \xeCJKsetkern { m m m }
1263 { \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。

```

1264 \keys_define:nn { xeCJK / options }
1265 {
1266   AutoFallBack .choice:,
1267   AutoFallBack / true .code:n = { \xeCJKenablefallback },
1268   AutoFallBack / false .code:n = { \xeCJKdisablefallback },
1269   AutoFallBack .default:n = { true },
1270   fallback .meta:n = { AutoFallBack = true },
1271 }
(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 1272 \bool_new:N \l_xeCJK_fallback_bool
1273 \NewDocumentCommand \xeCJKenablefallback { }
1274 {
1275   \bool_if:NF \l_xeCJK_fallback_bool
1276   {
1277     \bool_set_true:N \l_xeCJK_fallback_bool
1278     \cs_set_eq:NN \xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1279     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_testsymbol:N
1280   }
1281 }
1282 \NewDocumentCommand \xeCJKdisablefallback { }
1283 {
1284   \bool_if:NT \l_xeCJK_fallback_bool
1285   {
1286     \bool_set_false:N \l_xeCJK_fallback_bool
1287     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_save_CJKsymbol:N
1288   }
1289 }
```

(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。

```
1290 \cs_new_protected_nopar:Npn \xeCJK_fallback_testsymbol:N #1
1291 {
1292   \font_glyph_if_exist:NnTF \tex_font:D {'#1}
1293   { \xeCJK_fallback_save_CJKsymbol:N {'#1} }
1294   {
1295     \xeCJK_family_if_exist:xTF { \xeCJK@family/FallBack }
1296     { {
1297       \tl_put_right:Nn \xeCJK@family { /FallBack }
1298       \xeCJK_select_font:
1299       \xeCJK_fallback_testsymbol:N {'#1}
1300     } }
1301     {
1302       \xeCJK_warning:nx { fallback } {'#1}
1303       \xeCJK_fallback_save_CJKsymbol:N {'#1}
1304     }
1305   }
1306 }
1307 \xeCJK_msg_new:nn { fallback }
1308 {
1309   CJKfamily~'\xeCJK@family'~
1310   ( \prop_get:NV \g_xeCJK_family_prop \xeCJK@family )~
1311   does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {'#1}).\\
1312 }
```

(End definition for \xeCJK_fallback_testsymbol:N)

`\setCJKfallbackfamilyfont`

```
\xeCJK_set_family_fallback:nnn 1313 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1314 {
1315   \exp_args:Nx \tl_if_in:nnTF {#3} { , , }
1316   { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} }
```

```

1317     { \xeCJK_set_family:nnn { #1/FallBack } {#2} {#3} }
1318   }
1319   \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnn #1#2#3
1320   {
1321     \group_begin:
1322     \tl_set:Nx \l_xeCJK_family_fb_tl {#1}
1323     \prop_get:NVNF \g_xeCJK_family_prop \l_xeCJK_family_fb_tl \l_xeCJK_fontname_tl
1324     { \tl_clear:N \l_xeCJK_fontname_tl }
1325     \clist_map_inline:xn {#3}
1326     {
1327       \tl_put_right:Nn \l_xeCJK_family_fb_tl { /FallBack }
1328       \xeCJK_get_sub_features:nn \l_xeCJK_family_fb_tl {##1}
1329       \clist_put_left:cx { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist } {#2}
1330       \xeCJK_set_family:ncc \l_xeCJK_family_fb_tl
1331       { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist }
1332       { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontname_tl }
1333     }
1334     \group_end:
1335   }

```

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

5.10 CJK 字体族声明方式

```

1336 \bool_new:N \g_xeCJK_AutoFakeBold_bool
1337 \bool_new:N \g_xeCJK_AutoFakeSlant_bool
1338 \fp_new:N \g_xeCJK_EmboldenFactor_fp
1339 \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

```

1340 \keys_define:nn { xeCJK / options }
1341 {
1342   AutoFakeBold .choice:,
1343   AutoFakeBold / true .code:n =
1344     { \bool_set_true:N \g_xeCJK_AutoFakeBold_bool },
1345   AutoFakeBold / false .code:n =
1346     { \bool_set_false:N \g_xeCJK_AutoFakeBold_bool },
1347   AutoFakeBold / unknown .code:n =
1348     {
1349       \bool_set_true:N \g_xeCJK_AutoFakeBold_bool
1350       \fp_set:Nn \g_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1351     },
1352   AutoFakeBold .default:n = { true },
1353   AutoFakeSlant .choice:,
1354   AutoFakeSlant / true .code:n =
1355     { \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool },
1356   AutoFakeSlant / false .code:n =
1357     { \bool_set_false:N \g_xeCJK_AutoFakeSlant_bool },
1358   AutoFakeSlant / unknown .code:n =
1359     {
1360       \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool
1361       \fp_set:Nn \g_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1362     },
1363   AutoFakeSlant .default:n = { true },

```

```

1364     EmboldenFactor .fp_set:N = \g_xeCJK_EmboldenFactor_fp,
1365     SlantFactor    .fp_set:N = \g_xeCJK_SlantFactor_fp,
1366     BoldFont      .meta:n = { AutoFakeBold = true },
1367     boldfont      .meta:n = { AutoFakeBold = true },
1368     SlantFont     .meta:n = { AutoFakeSlant = true },
1369     slantfont     .meta:n = { AutoFakeSlant = true },
1370   }

```

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

\keys_set_known:nxN

```

1371 \cs_generate_variant:Nn \keys_set_known:nnN { nx }

```

(End definition for \keys_set_known:nxN)

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

```

\g_xeCJK_sub_key_clist 1372 \clist_new:N \g_xeCJK_sub_key_clist
1373 \cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
1374 {
1375   \clist_gput_right:Nx \g_xeCJK_sub_key_clist {#1}
1376   \keys_define:nn { xeCJK / features }
1377   {
1378     #1 .code:n =
1379     {
1380       \xeCJK_if_blank_x:nTF {##1}
1381       {
1382         \bool_set_false:c { l_xeCJK_copy_#1_bool }
1383         \bool_set_false:c { l_xeCJK_add_#1_bool }
1384         \tl_put_right:Nx \l_xeCJK_family_tl { /#1 }
1385       }
1386       {
1387         \clist_put_right:Nx \l_xeCJK_sub_key_clist {#1}
1388         \str_if_eq_x:nnTF {##1} *
1389         {
1390           \bool_set_true:c { l_xeCJK_copy_#1_bool }
1391           \bool_set_false:c { l_xeCJK_add_#1_bool }
1392         }
1393         {
1394           \bool_set_false:c { l_xeCJK_copy_#1_bool }
1395           \bool_set_true:c { l_xeCJK_add_#1_bool }
1396           \xeCJK_get_sub_features:nn {#1} {##1}
1397         }
1398       }
1399     },
1400     #1 .default:n = \c_empty_tl,
1401   }
1402 }
1403 \cs_generate_variant:Nn \xeCJK_new_sub_key:n { x }
1404 \cs_new_protected_nopar:Npn \xeCJK_get_sub_features:nn #1#2
1405 {
1406   \tl_set:Nx \l_xeCJK_tmpa_tl {#2}
1407   \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1408   \xeCJK_gobble_brace:N \l_xeCJK_tmpa_tl
1409   \exp_args:No \tl_if_head_eq_charcode:nNTF \l_xeCJK_tmpa_tl [ % ]

```

```

1410     {
1411         \exp_after:wN \xeCJK_get_sub_features_aux:n \l_xeCJK_tmpa_tl \c_empty_tl
1412         \xeCJK_if_blank_x:nT \l_xeCJK_sub_fontname_tl
1413         {
1414             \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_tmpa_tl
1415             \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1416         }
1417     }
1418     { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_tmpa_tl }
1419     \xeCJK_if_blank_x:nTF \l_xeCJK_sub_fontname_tl
1420     { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_fontname_tl }
1421     { \tl_replace_all:Nnx \l_xeCJK_sub_fontname_tl * \l_xeCJK_fontname_tl }
1422     \clist_set:cx { \l_xeCJK_#1_fontfeat_clist } \l_xeCJK_sub_fontfeat_clist
1423     \tl_set:cx { \l_xeCJK_#1_fontname_tl } \l_xeCJK_sub_fontname_tl
1424 }

```

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

\xeCJK_gobble_brace:N

\xeCJK_get_sub_features:nn

\xeCJK_get_sub_features_aux:n

```

1425 \cs_new_protected_nopar:Npn \xeCJK_gobble_brace:N #1
1426 {
1427     \exp_args:No \tl_if_head_is_group:nT #1
1428     {
1429         \exp_last_unbraced:NNo \tl_set:Nn #1 #1
1430         \xeCJK_gobble_brace:N #1
1431     }
1432 }
1433 \cs_new_protected_nopar:Npn \xeCJK_get_sub_features_aux:n [#1] #2
1434 {
1435     \clist_set:Nx \l_xeCJK_sub_fontfeat_clist {#1}
1436     \tl_set:Nx \l_xeCJK_sub_fontname_tl {#2}
1437 }

```

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

FallBack

```

1438 \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 1439 \cs_new_nopar:Npn \xeCJK_map_features_id:n #1
    ItalicFont 1440 { \prop_get:Nn \g_xeCJK_features_id_prop {#1} }
    BoldItalicFont 1441 \prop_new:N \g_xeCJK_features_id_prop
    SlantedFont 1442 \prop_put:Nnn \g_xeCJK_features_id_prop { bf } { Bold }
    BoldSlantedFont 1443 \prop_put:Nnn \g_xeCJK_features_id_prop { it } { Italic }
    BoldFeatures 1444 \prop_put:Nnn \g_xeCJK_features_id_prop { bfit } { BoldItalic }
    ItalicFeatures 1445 \prop_put:Nnn \g_xeCJK_features_id_prop { sl } { Slanted }
    BoldItalicFeatures 1446 \prop_put:Nnn \g_xeCJK_features_id_prop { bfsl } { BoldSlanted }
    SlantedFeatures 1447 \prop_map_inline:Nn \g_xeCJK_features_id_prop
    BoldSlantedFeatures 1448 {
1449     \keys_define:nn { xeCJK / features }
1450     {
1451         #2Font .tl_set_x:c = { \l_xeCJK_fontname_#1_tl },
1452         #2Features .tl_set_x:c = { \l_xeCJK_fontfeat_#1_clist } ,

```

```

1453     }
1454 }
(End definition for \g_xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

Mono

```

1455 \keys_define:nn { xeCJK / features }
1456 {
1457   AutoFakeBold .choice:,
1458   AutoFakeBold / false .code:n =
1459     { \bool_set_false:N \l_xeCJK_AutoFakeBold_bool },
1460   AutoFakeBold / unknown .code:n =
1461     {
1462       \bool_set_true:N \l_xeCJK_AutoFakeBold_bool
1463       \fp_set:Nn \l_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1464     },
1465   AutoFakeBold .default:n = { \g_xeCJK_EmboldenFactor_fp },
1466   AutoFakeSlant .choice:,
1467   AutoFakeSlant / false .code:n =
1468     { \bool_set_false:N \l_xeCJK_AutoFakeSlant_bool },
1469   AutoFakeSlant / unknown .code:n =
1470     {
1471       \bool_set_true:N \l_xeCJK_AutoFakeSlant_bool
1472       \fp_set:Nn \l_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1473     },
1474   AutoFakeSlant .default:n = { \g_xeCJK_SlantFactor_fp },
1475   Mono .choice:,
1476   Mono / Exspace .code:n =
1477     {
1478       \xeCJK_set_monoexspace:
1479       \tl_clear:N \l_xeCJK_monoscale_tl
1480     },
1481   Mono / Scale .code:n =
1482     {
1483       \xeCJK_set_monoscale:
1484       \tl_set:Nx \l_xeCJK_monoscale_tl { Scale = { \fp_use:N \g_xeCJK_monoscale_fp } }
1485     }
1486 }

```

(End definition for AutoFakeBold, AutoFakeSlant, and Mono)

\xeCJK_set_init:

```

1487 \cs_new_protected_nopar:Npn \xeCJK_set_init:
1488 {
1489   \tl_clear:N \l_xeCJK_monoscale_tl
1490   \int_gincr:N \g_xeCJK_fontspec_int
1491   \prop_clear:N \l_xeCJK_add_font_prop
1492   \prop_map_inline:Nn \g_xeCJK_features_id_prop
1493     {
1494       \tl_clear:c { l_xeCJK_fontname_##1_tl }
1495       \clist_clear:c { l_xeCJK_fontfeat_##1_clist }
1496     }
1497   \clist_clear:N \l_xeCJK_pass_features_clist
1498   \clist_clear:N \l_xeCJK_sub_key_clist

```

```

1499 \clist_map_inline:Nn \g_xeCJK_sub_key_clist
1500 {
1501     \bool_set_false:c { l_xeCJK_copy_##1_bool }
1502     \bool_set_false:c { l_xeCJK_add_##1_bool }
1503 }
1504 \bool_set_eq:NN \l_xeCJK_AutoFakeBold_bool \g_xeCJK_AutoFakeBold_bool
1505 \bool_set_eq:NN \l_xeCJK_AutoFakeSlant_bool \g_xeCJK_AutoFakeSlant_bool
1506 \fp_set_eq:NN \l_xeCJK_EmboldenFactor_fp \g_xeCJK_EmboldenFactor_fp
1507 \fp_set_eq:NN \l_xeCJK_SlantFactor_fp \g_xeCJK_SlantFactor_fp
1508 }

```

(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。

```

1509 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
1510 {
1511     \group_begin:
1512     \xeCJK_set_init:
1513     \tl_set:Nx \l_xeCJK_family_tl {#1}
1514     \clist_set:Nx \l_xeCJK_fontoptions_clist {#2}
1515     \tl_set:Nx \l_xeCJK_fontname_tl {#3}
1516     \keys_set_known:nxN { xeCJK / features }
1517     { \g_xeCJK_default_features_clist, #2 } \l_xeCJK_pass_features_clist
1518     \xeCJK_parse_features:
1519     \xeCJK_pass_features:
1520     \xeCJK_check_family:V \l_xeCJK_family_tl
1521     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1522     {
1523         \exp_not:n { \fontspec_set_family:Nnn \l_xeCJK_fontspec_family_tl }
1524         { \l_xeCJK_pass_features_clist } { \l_xeCJK_fontname_tl }
1525         \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1526         { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_fontspec_family_tl
1527         \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1528         \exp_not:n
1529         { {
1530             \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1531             \exp_not:N \fontfamily { \l_xeCJK_fontspec_family_tl }
1532             \exp_not:N \selectfont
1533         } }
1534     }
1535     \xeCJK_save_family_info:
1536     \xeCJK_add_sub_block:
1537     \group_end:
1538 }

```

\cs_generate_variant:Nn \xeCJK_set_family:nnn { ncc }

(End definition for \xeCJK_set_family:nnn)

\xeCJK_check_family:n

```

1540 \cs_new_protected_nopar:Npn \xeCJK_check_family:n #1
1541 {
1542     \prop_get:NnNT \g_xeCJK_family_prop {#1} \l_xeCJK_tmpa_tl
1543     {
1544         \prop_gpop:NnNT \g_xeCJK_family_name_prop {#1} \l_xeCJK_tmpa_tl { }

```



```

1545         \xeCJK_warning:nxx { CJKfamily-redef } {#1} \l_xeCJK_tmpa_tl
1546     }
1547 }
1548 \cs_generate_variant:Nn \xeCJK_check_family:n { V }
1549 \xeCJK_msg_new:nn { CJKfamily-redef } { Redefining~CJKfamily~`#1'~(#2). }
(End definition for \xeCJK_check_family:n)

```

```

\l_xeCJK_add_font:nn
\l_xeCJK_add_font_if_new:nn
1550 \cs_new_protected_nopar:Npn \l_xeCJK_add_font:nn #1#2
1551 {
1552     \prop_put:Nxx \l_xeCJK_add_font_prop
1553     { \xeCJK_map_features_id:n {#1} Font } {#2}
1554 }
1555 \cs_new_protected_nopar:Npn \l_xeCJK_add_font_if_new:nn #1#2
1556 {
1557     \prop_put_if_new:Nxx \l_xeCJK_add_font_prop
1558     { \xeCJK_map_features_id:n {#1} Font } {#2}
1559 }
1560 \cs_generate_variant:Nn \l_xeCJK_add_font:nn { nx, nc, nV, nv }
1561 \cs_generate_variant:Nn \l_xeCJK_add_font_if_new:nn { nx, nc, nV, nv }
(End definition for \l_xeCJK_add_font:nn and \l_xeCJK_add_font_if_new:nn)

1562 \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1563 \cs_generate_variant:Nn \prop_get:Nn { NV }
1564 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1565 \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1566 \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

```

\l_xeCJK_add_fake_bold:n
\l_xeCJK_add_fake_slant:n
1567 \cs_new_protected_nopar:Npn \l_xeCJK_add_fake_bold:n #1
1568 {
1569     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1570     { FakeBold = { \fp_use:N \l_xeCJK_EmboldenFactor_fp } }
1571 }
1572 \cs_new_protected_nopar:Npn \l_xeCJK_add_fake_slant:n #1
1573 {
1574     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1575     { FakeSlant = { \fp_use:N \l_xeCJK_SlantFactor_fp } }
1576 }
(End definition for \l_xeCJK_add_fake_bold:n and \l_xeCJK_add_fake_slant:n)

```

```

\l_xeCJK_parse_features:
\l_xeCJK_parse_features:n
1577 \cs_new_protected_nopar:Npn \l_xeCJK_parse_features:
1578 { \prop_map_inline:Nn \g_xeCJK_features_id_prop { \l_xeCJK_parse_features:n {##1} } }
1579 \cs_new_protected_nopar:Npn \l_xeCJK_parse_features:n #1
1580 {
1581     \xeCJK_if_font_select:nTF {#1}
1582     {
1583         \xeCJK_add_font:nv {#1} { l_xeCJK_fontname_#1_tl }
1584         \xeCJK_if_it_or_sl:nTF {#1}
1585         {
1586             \xeCJK_if_font_select:nF {bf#1}
1587             { \xeCJK_add_font_if_new:nv {bf#1} { l_xeCJK_fontname_#1_tl } }

```

```

1588     }
1589     {
1590         \str_if_eq_x:nnT {#1} {bf}
1591         {
1592             \clist_map_inline:nn { it , sl }
1593             {
1594                 \xeCJK_if_font_select:nF {bf##1}
1595                 { \xeCJK_add_font:nV {bf##1} \l_xeCJK_fontname_bf_tl }
1596             }
1597         }
1598     }
1599 }
1600 { \xeCJK_set_fake:n {#1} }
1601 }

```

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

```

\xeCJK_if_font_select_p:n
  \xeCJK_if_it_or_sl_p:n 1602 \prg_new_conditional:Npnn \xeCJK_if_font_select:n #1 { p, T, F, TF }
\xeCJK_if_font_select:nTF 1603 {
  \xeCJK_if_it_or_sl:nTF 1604   \exp_args:Nc \xeCJK_if_blank_x:nTF { \l_xeCJK_fontname_#1_tl }
  \prg_return_false: \prg_return_true:
1606 }
1607 \prg_new_conditional:Npnn \xeCJK_if_it_or_sl:n #1 { p, T, F, TF }
1608 {
1609   \bool_if:nTF { \str_if_eq_x_p:nn { it } {#1} || \str_if_eq_x_p:nn { sl } {#1} }
1610   \prg_return_true: \prg_return_false:
1611 }

```

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

```

\xeCJK_set_fake:n
1612 \cs_new_protected_nopar:Npn \xeCJK_set_fake:n #1
1613 {
1614   \str_if_eq_x:nnTF {#1} { bf }
1615   { \bool_if:NT \l_xeCJK_AutoFakeBold_bool { \xeCJK_add_fake_bold:n {#1} } }
1616   {
1617     \bool_if:NTF \l_xeCJK_AutoFakeSlant_bool
1618     {
1619       \bool_if:nT
1620       { \xeCJK_if_it_or_sl_p:n {#1} ||
1621         ( \str_if_eq_x_p:nn {#1} {bf} ) && ! ( \xeCJK_if_font_select_p:n {it} ) ) ||
1622         ( \str_if_eq_x_p:nn {#1} {bfs} ) && ! ( \xeCJK_if_font_select_p:n {sl} ) )
1623       }
1624       { \xeCJK_add_fake_slant:n {#1} }
1625     }
1626     { \xeCJK_if_it_or_sl:nT {#1} { \xeCJK_map_it_sl:n {#1} } }
1627     \bool_if:NT
1628     { \l_xeCJK_AutoFakeBold_bool
1629       && ! ( \xeCJK_if_it_or_sl_p:n {#1} )
1630       && ! ( \xeCJK_if_font_select_p:n {bf} )
1631     }
1632     { \xeCJK_add_fake_bold:n {#1} }
1633   }
}

```

```

1634     \xeCJK_add_font_if_new:nn {#1} *
1635   }
  (End definition for \xeCJK_set_fake:n)

```

\xeCJK_map_it_sl:n

```

1636 \cs_new_protected_nopar:Npn \xeCJK_map_it_sl:n #1
1637 {
1638   \xeCJK_if_map_font_select:nT {#1}
1639   {
1640     \xeCJK_add_font:nx {#1} { \xeCJK_get_map_font:n {#1} }
1641     \xeCJK_if_font_select:nF {bf#1}
1642     { \xeCJK_add_font_if_new:nx {bf#1} { \xeCJK_get_map_font:n {#1} } }
1643   }
1644 }
  (End definition for \xeCJK_map_it_sl:n)

```

\xeCJK_get_map_font:n

```

1645 \cs_new_nopar:Npn \xeCJK_get_map_font:n #1
1646 { \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 1647 \prg_new_conditional:Npnn \xeCJK_if_map_font_select:n #1 { p, T, F, TF }
1648 {
1649   \xeCJK_if_blank_x:nTF { \xeCJK_get_map_font:n {#1} }
1650   \prg_return_false: \prg_return_true:
1651 }
  (End definition for \xeCJK_if_map_font_select:n)

```

\xeCJK_pass_features:

```

1652 \cs_new_protected_nopar:Npn \xeCJK_pass_features:
1653 {
1654   \prop_map_inline:Nn \g_xeCJK_features_id_prop
1655   {
1656     \clist_if_empty:cF { l_xeCJK_fontfeat_##1_clist }
1657     {
1658       \clist_put_right:Nx \l_xeCJK_pass_features_clist
1659       { ##2Features = { \tl_use:c { l_xeCJK_fontfeat_##1_clist } } }
1660     }
1661   }
1662   \prop_map_inline:Nn \l_xeCJK_add_font_prop
1663   { \clist_put_right:Nx \l_xeCJK_pass_features_clist { ##1 = { ##2 } } }
1664   \xeCJK_if_blank_x:nF \l_xeCJK_monoscale_tl
1665   { \clist_put_right:Nx \l_xeCJK_pass_features_clist \l_xeCJK_monoscale_tl }
1666 }
  (End definition for \xeCJK_pass_features:)

```

\g_xeCJK_family_prop

```

\g_xeCJK_family_name_prop 1667 \prop_new:N \g_xeCJK_family_prop
\g_xeCJK_family_options_prop 1668 \prop_new:N \g_xeCJK_family_name_prop
1669 \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:

```
1670 \cs_new_protected_nopar:Npn \xeCJK_save_family_info:
1671 {
1672   \group_begin:
1673   \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1674   {
1675     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist {##1}
1676     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist { ##1 = }
1677   }
1678   \prop_gput:NVV \g_xeCJK_family_prop \l_xeCJK_family_tl \l_xeCJK_fontname_tl
1679   \prop_gput:NVV \g_xeCJK_family_options_prop \l_xeCJK_family_tl \l_xeCJK_fontoptions_clist
1680   \group_end:
1681 }
(End definition for \xeCJK_save_family_info:)
```

\xeCJK_add_sub_block:

```
1682 \cs_new_protected_nopar:Npn \xeCJK_add_sub_block:
1683 {
1684   \clist_remove_duplicates:N \l_xeCJK_sub_key_clist
1685   \clist_map_inline:Nn \l_xeCJK_sub_key_clist
1686   {
1687     \tl_set:Nx \l_xeCJK_sub_family_tl { \l_xeCJK_family_tl/##1 }
1688     \bool_if:cT { l_xeCJK_copy_##1_bool }
1689     {
1690       \xeCJK_check_family:V \l_xeCJK_sub_family_tl
1691       \prop_get:NVNT \g_xeCJK_family_prop \l_xeCJK_family_tl \l_xeCJK_tmpa_tl
1692       { \prop_gput:NVV \g_xeCJK_family_prop \l_xeCJK_sub_family_tl \l_xeCJK_tmpa_tl }
1693       \prop_get:NVNT \g_xeCJK_family_options_prop \l_xeCJK_family_tl \l_xeCJK_tmpa_clist
1694       {
1695         \clist_remove_all:Nn \l_xeCJK_tmpa_clist { ##1 = * }
1696         \prop_gput:NVV \g_xeCJK_family_options_prop \l_xeCJK_sub_family_tl \l_xeCJK_tmpa_clist
1697       }
1698       \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_sub_family_tl }
1699       {
1700         \xeCJK_family_if_exist:xT { \l_xeCJK_family_tl }
1701         {
1702           \prop_get:NnNT \exp_not:N \g_xeCJK_family_name_prop
1703           { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_tmpa_tl
1704           {
1705             \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1706             { \l_xeCJK_sub_family_tl } \exp_not:N \l_xeCJK_tmpa_tl
1707             \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_sub_family_tl }
1708             \exp_not:n
1709             { {
1710               \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1711               \exp_not:N \fontfamily { \l_xeCJK_tmpa_tl }
1712               \exp_not:N \selectfont
1713             } }
1714           }
1715         }
1716       }
1717     }
  }
```

```

1718     \bool_if:cT { l_xeCJK_add_##1_bool }
1719     {
1720         \xeCJK_set_family:ncc \l_xeCJK_sub_family_tl
1721         { l_xeCJK_##1_fontfeat_clist } { l_xeCJK_##1_fontname_tl }
1722     }
1723 }
1724 }
(End definition for \xeCJK_add_sub_block:)

```

\xeCJK_copy_family:nn

```

1725 \cs_new_protected_nopar:Npn \xeCJK_copy_family:nn #1#2
1726 {
1727     \xeCJK_family_if_exist:xT {#2}
1728     {
1729         \tl_map_inline:nn
1730         { \g_xeCJK_family_prop \g_xeCJK_family_options_prop \g_xeCJK_family_name_prop }
1731         { \prop_get:NnNT ##1 {#2} \l_xeCJK_tmpa_tl { \prop_gput:NnV ##1 {#1} \l_xeCJK_tmpa_tl } }
1732         \cs_gset_eq:cc { xeCJK/family/#1 } { xeCJK/family/#2 }
1733     }
1734 }
1735 \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: 1736 \tl_set:Nn \l_xeCJK_font_coor_tl { xeCJK/\xeCJK@family/\f@series/\f@shape/\f@size }
\l_xeCJK_switch_font:nn 1737 \cs_new_protected_nopar:Npn \xeCJK_select_font:
1738 {
1739     \cs_if_exist_use:cF { \l_xeCJK_font_coor_tl }
1740     {
1741         \tl_set:Nx \l_xeCJK_current_coor_tl \l_xeCJK_font_coor_tl
1742         \xeCJK_family_if_exist:xTF { \xeCJK@family }
1743         { \use:c { xeCJK/family/\xeCJK@family } }
1744         { \use_none:nnnn }
1745         \exp_last_unbraced:NNV \cs_gset_eq:cN \l_xeCJK_current_coor_tl \tex_font:D
1746     }
1747 }
1748 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
1749 \cs_new_protected_nopar:Npn \xeCJK_switch_font:nn #1#2
1750 {
1751     \str_if_eq_x:nnF {#1} {#2}
1752     {
1753         \xeCJK_info:nxx { CJK-block } {#1} {#2}
1754         \tl_remove_all:Nn \xeCJK@family { /#1 }
1755         \str_if_eq_x:nnF {#2} { CJK }
1756         {
1757             \tl_set:Nx \l_xeCJK_tmpa_tl \xeCJK@family
1758             \tl_put_right:Nx \xeCJK@family { /#2 }
1759             \xeCJK_family_if_exist:xF \xeCJK@family
1760             {
1761                 \xeCJK_family_if_exist:xTF { \CJKfamilydefault/#2 }

```

```

1762         { \xeCJK_copy_family:xx \xeCJK@family { \CJKfamilydefault/#2 } }
1763         { \xeCJK_copy_family:xx \xeCJK@family \l_xeCJK_tmpa_tl }
1764     }
1765 }
1766 }
1767 \xeCJK_select_font:
1768 }
1769 \xeCJK_msg_new:nn { CJK-block } { Switch~from~block~'#1'~to~'#2'. }
(End definition for \l_xeCJK_font_coor_tl, \xeCJK_select_font:, and \xeCJK_switch_font:nn)

```

`\xeCJK_family_if_exist:xTF`

```

1770 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T, F, TF }
1771 {
1772     \cs_if_free:cTF { xeCJK/family/#1 }
1773     { \prg_return_false: }
1774     {
1775         \prop_get:NxNF \g_xeCJK_family_name_prop {#1} \l_xeCJK_family_name_tl
1776         { \use:c { xeCJK/family/#1 } }
1777         \prg_return_true:
1778     }
1779 }
1780 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
(End definition for \xeCJK_family_if_exist:xTF)

```

\CJKfamily 用于切换 CJK 字体族。

```

1781 \NewDocumentCommand \CJKfamily { m }
1782 {
1783     \cs_if_exist:cTF { xeCJK/family/#1 }
1784     { \tl_set:Nx \xeCJK@family {#1} }
1785     {
1786         \seq_if_in:NxF \g_xeCJK_unknown_family_seq {#1}
1787         {
1788             \seq_gput_right:Nx \g_xeCJK_unknown_family_seq {#1}
1789             \xeCJK_warning:nx { CJKfamily-Unknown } {#1}
1790         }
1791     }
1792     \tex_ignorespaces:D
1793 }
1794 \seq_new:N \g_xeCJK_unknown_family_seq
1795 \xeCJK_msg_new:nn { CJKfamily-Unknown }
1796 {
1797     Unknown~CJK~family~'#1'~is~ignored.\\\
1798     Try~to~use~
1799     \str_case:x:nnn {#1}
1800     {
1801         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1802         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1803         \CJKttdefault { \token_to_str:N \setCJKmonofont }
1804     }
1805     { \token_to_str:N \setCJKfamilyfont {#1} }
1806     [...]{}~to~define~it.
1807 }

```

(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。
\newCJKfontfamily 1808 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
\CJKfontspec 1809 { \xeCJK_set_family:nnn {#1} {#2} {#3} }
1810 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1811 {
1812   \IfNoValueTF {#1}
1813     { \tl_set:Nx \l_xeCJK_family_tl { \cs_to_str:N #2 } }
1814     { \tl_set:Nx \l_xeCJK_family_tl {#1} }
1815   \xeCJK_set_family:nnn \l_xeCJK_family_tl {#3} {#4}
1816   \cs_new_protected_nopar:Npx #2 { \exp_not:N \CJKfamily { \l_xeCJK_family_tl } }
1817 }
1818 \int_new:N \g_xeCJK_fontspec_int
1819 \int_gdecr:N \g_xeCJK_fontspec_int
1820 \NewDocumentCommand \CJKfontspec { O{} m }
1821 {
1822   \tl_set:Nx \l_xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1823   \cs_if_free:cT \l_xeCJK_CJKfontspec_id_tl
1824   {
1825     \int_gincr:N \g_xeCJK_fontspec_int
1826     \tl_gset:Nx \g_xeCJK_family_spec_tl
1827       { CJKfontspec ( \int_use:N \g_xeCJK_fontspec_int ) }
1828     \xeCJK_set_family:nnn \g_xeCJK_family_spec_tl {#1} {#2}
1829     \tl_gset:cx \l_xeCJK_CJKfontspec_id_tl \g_xeCJK_family_spec_tl
1830   }
1831   \exp_args:Nv \CJKfamily \l_xeCJK_CJKfontspec_id_tl
1832   \tex_ignorespaces:D
1833 }
```

(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 1834 \clist_new:N \g_xeCJK_default_features_clist
1835 \NewDocumentCommand \defaultCJKfontfeatures { m }
1836 { \clist_gset:Nn \g_xeCJK_default_features_clist {#1} }
1837 \defaultCJKfontfeatures { Script = CJK }
1838 \@onlypreamble \defaultCJKfontfeatures
1839 \NewDocumentCommand \addCJKfontfeatures { m }
1840 {
1841   \prop_if_in:NVTF \g_xeCJK_family_prop \xeCJK@family
1842   {
1843     \group_begin:
1844     \clist_set:Nx \l_xeCJK_tmpa_clist {#1}
1845     \prop_get:NvN \g_xeCJK_family_prop \xeCJK@family \l_xeCJK_tmpa_tl
1846     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1847     {
1848       \clist_if_in:NnT \l_xeCJK_tmpa_clist {##1}
1849       {
1850         \clist_remove_all:Nn \l_xeCJK_tmpa_clist {##1}
1851         \prop_get:NxNF \g_xeCJK_family_prop { \xeCJK@family/##1 } \l_xeCJK_tmpb_tl
1852         {
1853           \prop_get:NxNF \g_xeCJK_family_prop
```

```

1854         { \CJKfamilydefault/##1 } \l_xeCJK_tmpb_tl
1855         { \tl_set:Nx \l_xeCJK_tmpb_tl \l_xeCJK_tmpa_tl }
1856     }
1857     \clist_set:Nx \l_xeCJK_tmpa_clist
1858     { ##1 = { [ \l_xeCJK_tmpa_clist ] { \l_xeCJK_tmpb_tl } } }
1859     \clist_map_break:
1860 }
1861 }
1862 \prop_get:NVNF \g_xeCJK_family_options_prop \xeCJK@family \l_xeCJK_tmpb_clist
1863 { \clist_clear:N \l_xeCJK_tmpb_clist }
1864 \clist_clear:N \l_xeCJK_tmpc_clist
1865 \clist_map_inline:Nn \l_xeCJK_tmpb_clist
1866 {
1867     \clist_if_in:NnF \l_xeCJK_tmpa_clist {##1}
1868     { \clist_put_right:Nn \l_xeCJK_tmpc_clist {##1} }
1869 }
1870 \clist_put_left:NV \l_xeCJK_tmpa_clist \l_xeCJK_tmpc_clist
1871 \use:x
1872 { \group_end: \exp_not:N \CJKfontspec [ \l_xeCJK_tmpa_clist ] { \l_xeCJK_tmpa_tl } }
1873 }
1874 { \xeCJK_warning:n { addCJKfontfeature-ignored } }
1875 \tex_ignorespaces:D
1876 }
1877 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
1878 \xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1879 {
1880     \tl_to_str:N \addCJKfontfeature (s)~ignored.\\\
1881     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1882 }

```

(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 1883 \NewDocumentCommand \setCJKmainfont { O{} m }
1884 { \xeCJK_set_family:nnn \CJKrmdefault {#1} {#2} }
1885 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1886 \NewDocumentCommand \setCJKsansfont { O{} m }
1887 { \xeCJK_set_family:nnn \CJKsfdefault {#1} {#2} }

```

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

\CJKrmdefault

\CJKsfdefault

\CJKttdefault

\CJKfamilydefault

```

1888 \cs_if_free:NT \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
1889 \cs_if_free:NT \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
1890 \cs_if_free:NT \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1891 \cs_if_free:NT \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1892 \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。

```

1893 \AtEndOfPackage
1894 {
1895     \tex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1896     \tex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }

```



```

1897 \etex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1898 \etex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1899 \cs_set_eq:NN \reset@font \normalfont
1900 }

```

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。

```

1901 \@onlypreamble \setCJKmainfont
1902 \@onlypreamble \setCJKmathfont
1903 \@onlypreamble \setCJKsansfont
1904 \@onlypreamble \setCJKmonofont
1905 \@onlypreamble \setCJKromanfont

```

5.12 数学字体设置

\setCJKmathfont 设置 CJK 数学字体。

```

1906 \NewDocumentCommand \setCJKmathfont { O{} m }
1907 { \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。

```

1908 \tl_const:Nn \c_xeCJK_math_tl { CJKmath }
1909 \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

```

1910 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1911 {
1912   \xeCJK_check_num_range:nnNN {#1} {#2} \l_xeCJK_tmpa_int \l_xeCJK_tmpb_int
1913   \int_set:Nn \l_xeCJK_tmppc_int { \xeCJK_math_type:n {#3} }
1914   \loop
1915     \XeTeXmathcode
1916     \l_xeCJK_tmpa_int = \l_xeCJK_tmppc_int \use:c { sym #4 } \l_xeCJK_tmpa_int
1917     \if_int_compare:w \l_xeCJK_tmpa_int < \l_xeCJK_tmpb_int
1918       \int_incr:N \l_xeCJK_tmpa_int
1919   \repeat
1920 }
1921 \cs_new_eq:NN \xeCJK_math_type:n \mathchar@type

```

(End definition for \xeCJKsetmathcode)

\xeCJK_set_mathfont:

```

1922 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
1923 {
1924   \xeCJK_family_if_exist:xF \c_xeCJK_math_tl
1925   { \tl_set:Nx \l_xeCJK_tmpa_tl \c_xeCJK_math_tl }
1926   {
1927     \xeCJK_family_if_exist:xF \CJKfamilydefault
1928     { \tl_set:Nx \l_xeCJK_tmpa_tl \CJKfamilydefault }
1929     { \tl_set_eq:NN \l_xeCJK_tmpa_tl \q_no_value }
1930   }
1931   \prop_get:NVNT \g_xeCJK_family_name_prop \l_xeCJK_tmpa_tl \l_xeCJK_math_family_tl
1932   {
1933     \DeclareSymbolFont \c_xeCJK_math_tl \g_fontspec_encoding_tl

```

```

1934 \l_xeCJK_math_family_tl \mddefault \shapedefault
1935 \SetSymbolFont \c_xeCJK_math_tl { bold } \g_fontspec_encoding_tl
1936 \l_xeCJK_math_family_tl \bfdefault \shapedefault
1937 \clist_map_inline:nn { CJK, FullLeft, FullRight }
1938 {
1939   \clist_map_inline:cn { g_xeCJK_##1_range_clist }
1940   {
1941     \xeCJK_set_char_class_aux:Nn \xeCJKsetmathcode {####1}
1942     \mathalpha \c_xeCJK_math_tl
1943   }
1944 }
1945 }
1946 }

```

(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。

```

1947 \xeCJK_AtEndPreamble:n
1948 {
1949   \normalfont
1950   \prop_if_empty:NTF \g_xeCJK_family_prop
1951   { \xeCJK_warning:n { no-CJKfamily } }
1952   { \bool_if:NT \g_xeCJK_math_bool { \xeCJK_set_mathfont: } }
1953 }
1954 \xeCJK_msg_new:nn { no-CJKfamily }
1955 {
1956   It~seems~that~you~have~not~declare~a~CJKfamily.\\
1957   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
1958   \token_to_str:N \xeCJKmainfont[...]{...}\\
1959   in~the~preamble~to~declare~the~main~CJKfamily.\\
1960 }

```

5.13 处理等宽字体和抄录环境

\g_xeCJK_monoscale_fp

\g_xeCJK_exspace_dim

```

1961 \fp_new:N \g_xeCJK_monoscale_fp
1962 \dim_new:N \g_xeCJK_exspace_dim
(End definition for \g_xeCJK_monoscale_fp and \g_xeCJK_exspace_dim)

```

\setCJKmonoscale

```

1963 \NewDocumentCommand \setCJKmonoscale { }
1964 {
1965   \CJKflexiblespacing
1966   \xeCJK_set_monoscale:
1967   \addCJKfontfeatures { Scale = \fp_use:N \g_xeCJK_monoscale_fp }
1968 }
1969 \cs_new_protected_nopar:Npn \xeCJK_set_monoscale:
1970 {
1971   \group_begin:
1972   \fontfamily \ttdefault \selectfont
1973   \fp_gset:Nn \g_xeCJK_monoscale_fp
1974   { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } / ( \f@size ) }
1975   \group_end:
1976 }

```

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

\setCJKmonoexspace

```
1977 \NewDocumentCommand \setCJKmonoexspace { } { \xeCJK_set_monoexspace: }
1978 \cs_new_protected_nopar:Npn \xeCJK_set_monoexspace:
1979 {
1980   \group_begin:
1981     \fontfamily \ttdefault \selectfont
1982     \dim_gset:Nn \g_xeCJK_exspace_dim
1983       { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size \p@ }
1984   \group_end:
1985 }
```

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

\CJKfixedspacing

\CJKflexiblespacing

```
1986 \NewDocumentCommand \CJKfixedspacing { }
1987 {
1988   \bool_if:NF \l_xeCJK_fixed_spacing_bool
1989   {
1990     \bool_set_true:N \l_xeCJK_fixed_spacing_bool
1991     \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1992     \bool_set_eq:NN \l_xeCJK_fixed_xecglue_bool \l_xeCJK_xecglue_bool
1993     \cs_set_eq:NN \xeCJK_flexible_ecglue: \CJKecglue
1994     \cs_set_eq:NN \xeCJK_flexible_cjkglue: \CJKglue
1995     \xeCJK_set_monoexspace:
1996     \xeCJKsetup
1997     {
1998       PunctStyle = plain ,
1999       CJKglue    = { \skip_horizontal:N \g_xeCJK_exspace_dim } ,
2000       CJKecglue  = { \skip_horizontal:n { .5\g_xeCJK_exspace_dim } } ,
2001       xCJKecglue = false ,
2002     }
2003   }
2004 }
2005 \NewDocumentCommand \CJKflexiblespacing { }
2006 {
2007   \bool_if:NT \l_xeCJK_fixed_spacing_bool
2008   {
2009     \bool_set_false:N \l_xeCJK_fixed_spacing_bool
2010     \exp_args:Nx \xeCJKsetup
2011     {
2012       PunctStyle = { \l_xeCJK_flexible_punctstyle_tl } ,
2013       CJKglue    = { \xeCJK_flexible_cjkglue: } ,
2014       CJKecglue  = { \xeCJK_flexible_ecglue: } ,
2015       xCJKecglue = { \bool_if:NTF \l_xeCJK_fixed_xecglue_bool { true } { false } } ,
2016     }
2017   }
2018 }
2019 \bool_new:N \l_xeCJK_fixed_spacing_bool
2020 \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。

```
2021 \NewDocumentCommand \setCJKmonofont { s t+ O{ } m }
2022 {
2023   \IfBooleanTF {#1}
2024   { \xeCJK_set_family:nnn \CJKttdefault { Mono = Scale, #3 } {#4} }
2025   {
2026     \IfBooleanTF {#2}
2027     { \xeCJK_set_family:nnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
2028     { \xeCJK_set_family:nnn \CJKttdefault {#3} {#4} }
2029   }
2030 }
```

(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

```
2031 \bool_new:N \g_xeCJK_indent_bool
2032 \keys_define:nn { xeCJK / options }
2033 {
2034   CJKnumber      .bool_set:N = \g_xeCJK_number_bool ,
2035   indentfirst    .bool_set:N = \g_xeCJK_indent_bool ,
2036   normalindentfirst .meta:n = { indentfirst = false } ,
2037   quiet .code:n =
2038   {
2039     \msg_redirect_module:nnn { xeCJK } { warning } { info }
2040     \msg_redirect_module:nnn { xeCJK } { info } { none }
2041     \xeCJK_if_package_loaded:nF { fontspec }
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     \xeCJK_if_package_loaded:nF { fontspec }
2049     { \PassOptionsToPackage { silent } { fontspec } }
2050   },
2051   unknown .code:n =
2052   {
2053     \xeCJK_if_package_loaded:nTF { fontspec }
2054     { \xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2055     { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2056   },
2057 }
2058 \xeCJK_msg_new:nn { key-unknown }
2059 {
2060   Sorry,~but~\l_keys_module_tl\c_space_tl does~not~have~a~key~called~'#1'.\\
2061   The~key~'#1'~is~being~ignored.
2062 }
```

(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。

```
2063 \keys_set:nn { xeCJK / options }
2064 {
2065   CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
2066   CJKecglue      = \c_space_token ,
2067   xCJKecglue     = false ,
2068   CheckSingle    = false ,
2069   CJKspace       = false ,
2070   CJKmath        = false ,
2071   xeCJKactive    = true ,
2072   indentfirst    = true ,
2073   EmboldenFactor = 4 ,
2074   SlantFactor    = .167 ,
2075   PunctStyle     = quanjiao ,
2076   KaiMingPunct   = { . 。 ? ! } ,
2077   LongPunct      = { — … ‣ — } ,
2078   MiddlePunct    = { · · · } ,
2079   AllowBreakBetweenPuncts = false ,
2080 }
```

\CJKsymbol

```
\CJKpunctsymbol 2081 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2082 \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。

```
2083 \ProcessKeysOptions { xeCJK / options }
2084 \RequirePackage { fontspec } [ 2011/09/13 ]

2085 \bool_if:NT \g_xeCJK_indent_bool
2086 {
2087   \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
2088   \@afterindenttrue
2089 }
```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```
2090 \NewDocumentCommand \xeCJKsetup { m }
2091 {
2092   \keys_set:nn { xeCJK / options } {#1}
2093   \tex_ignorespaces:D
2094 }
```

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

\xeCJKsetemboldenfactor

\xeCJKsetslantfactor

```
2095 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
2096 { \xeCJKsetup { EmboldenFactor = {#1} } }
2097 \NewDocumentCommand \xeCJKsetslantfactor { m }
2098 { \xeCJKsetup { SlantFactor = {#1} } }
```

(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor)

```

\punctstyle
\XeCJKplainchr 2099 \NewDocumentCommand \punctstyle { m } { \XeCJKsetup { PunctStyle = {#1} } }
2100 \NewDocumentCommand \XeCJKplainchr { } { \punctstyle { plain } }
(End definition for \punctstyle and \XeCJKplainchr)

```

```

\CJKsetecglue
2101 \NewDocumentCommand \CJKsetecglue { m } { \XeCJKsetup { CJKecglue = {#1} } }
2102 \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。

```

2103 \cs_new_protected_nopar:Npn \XeCJK_fix_itcorr:
2104 {
2105   \scan_stop:
2106   \XeCJK_if_last_node:nTF { default }
2107   {
2108     \tex_unkern:D \tex_unkern:D
2109     \XeCJK_itcorr_aux
2110     { \XeCJK_make_node:n { default_itcorr } }
2111   }
2112   { \XeCJK_itcorr_aux }
2113 }
2114 \cs_new_eq:NN \XeCJK_itcorr_aux \
2115 \cs_set_protected_nopar:Npn \
2116 {
2117   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
2118   { \XeCJK_fix_itcorr: } { \XeCJK_itcorr_aux }
2119 }
2120 \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 2121 \XeCJK_AfterEndPreamble:n
2122 {
2123   \int_compare:nNnF
2124   { \c_three + ( \seq_count:N \g_xeCJK_class_seq ) } = \xe@alloc@intercharclass
2125   {
2126     \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
2127     {
2128       \seq_if_in:NnF \g_xeCJK_class_seq {#1}
2129       { \XeCJK_set_others_toks:n {#1} }
2130     }
2131   }
2132 }
2133 \cs_new_protected_nopar:Npn \XeCJK_set_others_toks:n #1
2134 {
2135   \int_set:Nn \XeCJK_Others_class {#1}
2136   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
2137   {

```

```

2138 \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2139 \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2140 \exp_args:Nnnx \xeCJK_app_inter_class_toks:nnn {##1} { Others }
2141 { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
2142 \exp_args:Nnnx \xeCJK_pre_inter_class_toks:nnn { Others } {##1}
2143 { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
2144 \xeCJK_if_blank_x:nT { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2145 {
2146 \xeCJK_copy_inter_class_toks:nnnn
2147 { Others } { Boundary } { Default } { Boundary }
2148 }
2149 \xeCJK_if_blank_x:nT { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2150 {
2151 \xeCJK_copy_inter_class_toks:nnnn
2152 { Boundary } { Others } { Boundary } { Default }
2153 }
2154 }
2155 }
2156 \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。

```

2157 \cs_new_protected_nopar:Npn \xeCJK_patch:Nnn #1#2#3
2158 { \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。

```

2159 \xeCJK_AfterPreamble:n
2160 {
2161 \tl_map_inline:nn
2162 {
2163 \textellipsis \textemdash \textperiodcentered \textcentereddot
2164 \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2165 }
2166 {
2167 \xeCJK_patch:Nnn #1
2168 { \xeCJK_group_begin: \makexeCJKinactive }
2169 { \xeCJK_group_end: }
2170 }
2171 \xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2172 \cs_set_eq:NN \xeCJK_aux_r \r
2173 \cs_set_protected_nopar:Npn \r #1 { { \makexeCJKinactive \xeCJK_aux_r {#1} } }
2174 \xeCJK_if_package_loaded:nT { pifont }
2175 {
2176 \RenewDocumentCommand \Pifont { m }
2177 {
2178 \fontencoding {U} \fontfamily {#1} \fontseries {m} \fontshape {n}
2179 \selectfont \makexeCJKinactive
2180 }
2181 }
2182 }
2183 \cs_new_eq:NN \xeCJK_group_begin: \group_begin:

```

```

2184 \cs_new_eq:NN \xeCJK_group_end: \group_end:
      简单处理与 hyperref 宏包的兼容问题。
2185 \xeCJK_AfterEndPreamble:n
2186 {
2187   \bool_if:nT
2188   {
2189     \xeCJK_if_package_loaded_p:n { hyperref } &&
2190     \tl_if_exist_p:N \pdfstringdefPreHook
2191   }
2192   {
2193     \tl_put_right:Nn \pdfstringdefPreHook
2194     {
2195       \cs_set_eq:NN \makeCJKinactive \prg_do_nothing:
2196       \cs_set_eq:NN \xeCJK_group_begin: \prg_do_nothing:
2197       \cs_set_eq:NN \xeCJK_group_end: \prg_do_nothing:
2198       \cs_set_eq:NN \CJKfamily \use_none:n
2199     }
2200   }
2201 }
      禁止在 xeCJK 宏包后再载入 CJK 宏包。
2202 \tl_set:cn { ver@CJK.sty } { 2050/01/01 }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

2203 \cs_if_free:NT \CJK@ifundefined
2204 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2205 \NewDocumentCommand \xeCJKcaption { o m }
2206 {
2207   \IfValueT {#1} { \XeTeXdefaultencoding "#1" }
2208   \cs_set_nopar:Npx \xeCJK_reset_at_catcode:
2209   {
2210     \exp_not:n { \char_set_catcode:nn { \@ } }
2211     { \char_value_catcode:n { \@ } }
2212   }
2213   \char_set_catcode_letter:N \@
2214   \file_input:n { #2.cpx }
2215   \xeCJK_reset_at_catcode:
2216   \XeTeXdefaultencoding "UTF-8"
2217 }
      (End definition for \xeCJKcaption)

```

```

\XeCJK_ULprepunctchar:n
\XeCJK_ULpostpunctchar:n
\XeCJK_ULroutines:
2218 \cs_new_protected_nopar:Npn \xeCJK_ULprepunctchar:n #1
2219 {
2220   { \makeCJKinactive \CJKpunctsymbol {#1} \xeCJK_no_break: }
2221   \tex_ignorespaces:D
2222 }
2223 \cs_new_protected_nopar:Npn \xeCJK_ULpostpunctchar:n #1
2224 {
2225   { \makeCJKinactive \CJKpunctsymbol {#1} }
2226   \xeCJK_ignorespaces:
2227 }

```



```

2228 \cs_new_protected_nopar:Npn \xeCJK_ULroutines:
2229 {
2230   \xeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKecglue \CJKsymbol }
2231   \xeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2232   \xeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKecglue \CJKsymbol }
2233   \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }
2234   \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2235   {
2236     \xeCJK_inter_class_toks:nnn {##1} { FullLeft } { \xeCJK_ULprepunctchar:n }
2237     \xeCJK_inter_class_toks:nnn {##1} { FullRight } { \xeCJK_ULpostpunctchar:n }
2238   }
2239   \xeCJK_UL_subclass_patch_tl
2240 }

```

(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。

```

2241 \xeCJK_AfterPreamble:n
2242 {
2243   \cs_if_exist:NT \UL@hook
2244   {
2245     \addto@hook \UL@hook
2246     {
2247       \cs_set_eq:NN \xeCJK_UL_CJKsymbol \CJKsymbol
2248       \cs_set_eq:NN \xeCJK_UL_CJKpunctsymbol \CJKpunctsymbol
2249       \cs_set_protected_nopar:Npn \CJKsymbol #1
2250       {
2251         { \xeCJK_select_font: \xeCJK_UL_CJKsymbol {#1} }
2252         { \xeCJK_make_node:n { CJK } } \xeCJK_ignorespaces:
2253       }
2254       \cs_set_protected_nopar:Npn \CJKpunctsymbol #1
2255       { { \xeCJK_select_font: \xeCJK_UL_CJKpunctsymbol {#1} } }
2256       \xeCJK_ULroutines:
2257     }
2258   }
2259   \cs_if_exist:NT \XeTeX@CJKfntef@hook
2260   {
2261     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2262     { \xeCJK_select_font: \makexeCJKinactive }
2263   }
2264 }

```

使用 CJKnumb 宏包。

```

2265 \bool_if:NT \g_xeCJK_number_bool
2266 {
2267   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2268   \cs_set_protected_nopar:Npn \CJKaddEncHook #1#2
2269   { \cs_set_nopar:cpn { xeCJK_enc_#1 } {#2} }
2270   \cs_set_protected_nopar:Npn \Unicode #1#2
2271   { \tex_char:D \int_eval:n { (#1) * \c_two_hundred_fifty_six + (#2) } }
2272   \RequirePackage { CJKnumb }
2273   \use:c { xeCJK_enc_ \CJK@UnicodeEnc }
2274   \tl_set:Nn \CJK@tenthousand { 万 }
2275   \tl_set:Nn \CJK@hundredmillion { 亿 }
2276 }

```

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]{simsum.ttc}
5  \setCJKsansfont[AutoFakeSlant=false,
6    BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsum.ttc}
7  \setCJKmonofont[ItalicFont=simkai.ttf]{simsum.ttc}
8  \begin{document}
9  \centering
10 \begin{tabular}{lllll}
11 \hline
12 {\bf rm} & md & up & \verb|\rmfamily\mdseries\upshape| & &
13           & & & {\rmfamily\mdseries\upshape English 中文字体} & \\\
14           & md & it & \verb|\rmfamily\mdseries\itshape| & &
15           & & & {\rmfamily\mdseries\itshape English 中文字体} & \\\
16           & md & sl & \verb|\rmfamily\mdseries\slshape| & &
17           & & & {\rmfamily\mdseries\slshape English 中文字体} & \\\ \cline{2-5}
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 止 & 3400 & 𠂇 & 3401 & 𠂈 & 3402 & 个 & 3403 & \\
73 𠂉 & 3404 & 𠂊 & 3405 & 𠂋 & 3406 & 𠂌 & 3407 & \\
74 𠂍 & 3408 & 𠂎 & 3409 & 𠂏 & 340A & 𠂐 & 340B & \\
75 𠂑 & 340C & 𠂒 & 340D & 𠂓 & 340E & 𠂔 & 341F & \\
76 𠂕 & 3410 & 𠂖 & 3411 & 𠂗 & 3412 & 𠂘 & 3413 & \\
77 𠂙 & 3414 & 𠂚 & 3415 & 𠂛 & 3416 & 𠂜 & 3417 & \\
78 𠂝 & 3418 & 𠂞 & 3419 & 𠂟 & 341A & 𠂠 & 341B & \\
79 𠂡 & 341C & 𠂢 & 341D & 𠂣 & 341E & 𠂤 & 341F & \\
80 𠂥 & 20000 & 𠂦 & 20001 & 𠂧 & 20002 & 𠂨 & 20003 & \\
81 𠂩 & 20004 & 𠂪 & 20005 & 𠂫 & 20006 & 𠂬 & 20007 & \\
82 𠂭 & 20008 & 𠂮 & 20009 & 𠂯 & 2000A & 𠂰 & 2000B & \\
83 𠂱 & 2000C & 𠂲 & 2000D & 𠂳 & 2000E & 𠂴 & 2000F & \\
84 𠂵 & 20010 & 𠂶 & 20011 & 𠂷 & 20012 & 𠂸 & 20013 & \\
85 𠂹 & 20014 & 𠂺 & 20015 & 𠂻 & 20016 & 𠂼 & 20017 & \\
86 𠂽 & 20018 & 𠂾 & 20019 & 𠂿 & 2001A & 𠃀 & 2001B & \\
87 𠃁 & 2001C & 𠃂 & 2001D & 𠃃 & 2001E & 𠃄 & 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（英語：CJKV Unified Ideographs），旧称中日韩统一表意文字（英
語：CJK Unified Ideographs），也称统一汉字（英語：UniHan），目的是要把分别来自中
文、日文、韩文、越文、壮文中，对于相同起源、本义相同、形状一样或稍异的表意文字
（主要为汉字，但也有仿汉字如：方块壮字、日文汉字（かんじ / kanji）、韩文汉字（한
자 / hanja）、越南的喃字（Ch Nôm）与越文汉字（Ch Nho，在越南也称作儒字），应赋予
其在 ISO 10646 及统一码标准中有相同编码。此计划原本只包含中文、日文及韩文中所使用的
汉字，是以旧称中日韩统一表意文字（CJK）。后来，此计划加入了越文的喃字，所以合称中
日韩越统一表意文字（CJKV）。
107
108 CJK 統合漢字（シージェーケーとうごうかんじ、CJK Unified Ideographs）は、ISO/IEC 10646 (Uni-
versal Multiple-Octet Coded Character Set, 略称 UCS) および Unicode にて採用されてい
る符号化用漢字集合およびその符号表である。CJK 統合漢字の名称は、中国語（Chinese）、
日本語（Japanese）、韓国語（Korean）で使われている漢字をひとまとめにしたことから
きている。CJK 統合漢字の初版である Unified Repertoire and Ordering (URO) 第二版は
1992 年に制定されたが、1994 年にベトナムで使われていた漢字も含めることにしたため、
CJKV と呼ばれる事もある。CJKV は、中国語・日本語・韓国語・ベトナム語（英語：Chi-
nese-Japanese-Korean-Vietnamese）の略。特に、その 4 言語で共通して使われる、または
使われていた文字体系である漢字（チュノムを含む）のこと。ソフトウェアの国際化、中
でも文字コードに関する分野で用いられる。
109
110 \CJKspace
111 한중일월 통합 한자 (또는 한중일 통합 한자) 는 유니코드에 담겨 있는 한자들의 집합으
로, 한국, 중국, 일본에서 쓰이는 한자를 묶은 것이기 때문에 머리 글자를 따서 한중일
(CJK) 통합 한자라고 불렀는데, 최근에는 베트남에서 쓰이는 한자도 추가되었기에 한중일
월 (CJKV) 통합 한자로 부르게 되었다.
112
113 처음에 유니코드에는 65,536( $2^{16}$ ) 자만 들어갈 수 있었기 때문에, 가장 많은 문
자가 배당되는 한자를 위해서 한국, 중국, 일본에서 사용하는 한자 중에 모양이 유사하
며 그 뜻이 같은 글자를 같은 코드로 통합했다. 따라서 문자 코드만으로 그 한자가 사용
되는 언어를 알아 낼 수 없는데, 다만 중국의 간체자나 번체자, 일본의 구자체나 신자
체 등 분명하게 모양이 다른 글자는 별도의 부호를 할당하고 있다. 이런 문자 할당 정책
에 반발하여 TRON 과 같은 인코딩이 만들어지기도 했으나, 실제로 통합된 한자의 차이
가 별로 크지 않기 때문에 문제가 되지 않는다는 의견도 있다.
114 \CJKnospace}
115 \showtext
116
117 \bigskip
118 \xeCJKCancelSubCJKBlock{Kana, Hangul}

```

```

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

```

6.4 xeCJK-example-CJKecglue.tex

```

149 <*ex-ecglue>
150 \documentclass{minimal}
151 \usepackage{xeCJK}
152 \setCJKmainfont[BoldFont=SimHei]{SimSun}
153 \long\def\showtext{%
154 这是 English 中文 {\itshape Chinese} 中文 \TeX\
155 间隔 \textit{Italic} 中文\textbf{字体} a 数学 $b$ 数学 $c$ $d$\par
156 这是English中文{\itshape Chinese}中文\TeX\
157 间隔\textit{Italic}中文\textbf{字体}a数学$b$数学$c$ $d$\par
158 This is an example. 这是一个例子}
159 \begin{document}
160 \showtext
161
162 \hrulefill\bigskip
163
164 \xeCJKsetup{xCJKecglue=\quad}
165 \showtext
166 \end{document}
167 </ex-ecglue>

```

6.5 xeCJK-example-checksingle.tex

```

168 <*ex-single>
169 \documentclass{minimal}
170 \usepackage{xeCJK}
171 \setCJKmainfont{SimSun}
172 \catcode'\。=\active
173 \def。{. }
174 \def\foo{一}
175 \long\def\showtext{一二三四五六七八九十十一二三四五六七八九十十一二三四五六七八九十。}
176 $$x^2+y^2$$
177 一二三四五六七八九十十一二三四五六七八九十十一二三四五六七八九十\foo
178 \begin{equation}
179 x^2+y^2
180 \end{equation}
181 一二三四五六七八九十十一二三四五六七八九十十一二三四五六七八九十。}
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 </ex-single>

```

代码索引

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

Symbols	A
\/ 60, 62, 2114, 2115, 2120	\addCJKfontfeature 1877, 1880
\@@italiccorr 2120	\addCJKfontfeatures 7, <u>1834</u> , 1839, 1877, 1967
\@afterindentfalse 2087	\addto@hook 2245
\@afterindenttrue 2087, 2088	KaiMingPunct 5
\@depth 472	KaiMingPunct+ 5
\@height 473	KaiMingPunct- 5
\@ifpackagelater 18	FallBack 7
\@minus 432, 481	\AllowBreakBetweenPuncts <u>963</u>
\@onlypreamble 821, 872, 1838, 1901, 1902, 1903, 1904, 1905	\AtBeginDocument 53
\@pkgextension 127	\AtEndOfPackage 30, 1893
\@plus 431, 480, 2065	\AutoFakeBold <u>1340</u> , <u>1455</u>
\@width 471	\AutoFakeSlant <u>1340</u> , <u>1455</u>
\[..... 755	\AutoFallBack <u>1264</u>
\] 755	B
	\baselineskip 2065
	\begin 769, 770

\bfdefault	1936	\c_xeCJK_ps_marginkerning_tl	1157
\BoldFeatures	1439	\c_xeCJK_ps_mixedwidth_tl	1059, 1084
\BoldFont	1439	\c_xeCJK_ps_plain_tl	942, 1021, 1146, 1240
\BoldItalicFeatures	1439	\c_zero	115, 118, 144, 2117
\BoldItalicFont	1439	\c_zero_dim	472, 473, 1024, 1025, 1045, 1053, 1070, 1103, 1204
\BoldSlantedFeatures	1439	\c_zero_skip	984, 2065
\BoldSlantedFont	1439	\char_set_catcode:nn	2210
\bool_if:cT	1688, 1718	\char_set_catcode_ignore:n	145
\bool_if:Nf	686, 825, 849, 1275, 1988	\char_set_catcode_letter:N	2213
\bool_if:nF	456	\char_set_catcode_other:N	62
\bool_if:NT	695, 834, 1284, 1615, 1952, 2007, 2085, 2265	\char_set_catcode_other:n	208
\bool_if:nT	398, 642, 1200, 1232, 1619, 1627, 2187	\char_set_lccode:nn	57, 58, 59, 60, 61
\bool_if:NTF	421, 1617, 2015	\char_value_catcode:n	2211
\bool_if:nTF	217, 344, 352, 386, 390, 407, 416, 706, 722, 730, 735, 752, 1123, 1178, 1609	\CheckSingle	675
\bool_new:N	613, 683, 822, 1272, 1336, 1337, 2019, 2031	\CJK@hundredmillion	2275
\bool_set_eq:NN	1504, 1505, 1992	\CJK@ifundefined	2203, 2204
\bool_set_false:c	1382, 1383, 1391, 1394, 1501, 1502	\CJK@tenthousand	2274
\bool_set_false:N	600, 630, 697, 836, 1286, 1346, 1357, 1459, 1468, 2009	\CJK@UnicodeEnc	2267, 2273
\bool_set_true:c	1390, 1395	\CJKaddEncHook	2268
\bool_set_true:N	595, 605, 625, 688, 827, 1277, 1344, 1349, 1355, 1360, 1462, 1471, 1990	\CJKkecglue	354, 357, 362, 373, 396, 414, 585, 589, 590, 596, 606, 607, 608, 653, 670, 919, 1993, 2230, 2232
C			
\c_catcode_other_token	703, 719, 732	xCJKkecglue	4
\c_eleven	347, 401	\CJKfamily	6, 1781, 1781, 1816, 1831, 1895, 1896, 1897, 1898, 2198
\c_empty_tl	115, 301, 1400, 1411	\CJKfamilydefault	7, 1761, 1762, 1854, 1888, 1891, 1892, 1895, 1927, 1928
\c_four	432, 1256, 2126	\CJKfixedspacing	6, 1986, 1986, 2020
\c_group_begin_token	329, 379, 446	\CJKflexiblespacing	6, 1965, 1986, 2005
\c_group_end_token	334, 436, 452	\CJKfontspec	6, 1808, 1820, 1872
\c_math_toggle_token	639, 762, 764, 765, 766	\CJKglue	388, 421, 441, 514, 584, 584, 1994
\c_one	143, 502, 1255, 2126	\CJKmath	1908
\c_space_tl	24, 118, 857, 2060	\CJKnospace	618, 623, 628
\c_space_token	96, 354, 368, 421, 422, 601, 635, 660, 664, 733, 741, 742, 2066	\CJKnumber	2031
\c_ten	393, 410	\CJKpunctsymbol	491, 524, 542, 561, 2081, 2082, 2220, 2225, 2248, 2254
\c_ten_thousand	43	\CJKrmdefault	7, 1801, 1884, 1888, 1888, 1891, 1896
\c_three	431, 1256, 1258, 2124	\CJKsetecglue	2101, 2101, 2102
\c_two	430, 1255, 1974, 1983	\CJKsfdefault	7, 1802, 1887, 1888, 1889, 1897
\c_two_hundred_fifty_six	2271	\CJKspace	614, 617, 623, 623
\c_xeCJK_math_tl	1907, 1908, 1924, 1925, 1933, 1935, 1942	\CJKsymbol	336, 382, 441, 451, 757, 765, 782, 808, 809, 813, 814, 919, 920, 1278, 1279, 1287, 2081, 2081, 2230, 2231, 2232, 2247, 2249
\c_xeCJK_pkg_extension_tl	124, 127	\CJKsymbol_Block	808, 813
\c_xeCJK_ps_CCT_tl	1060, 1092	\CJKttdefault	7, 1803, 1888, 1890, 1898, 2024, 2027, 2028
\c_xeCJK_ps_fullwidth_tl	1061, 1165	\clist_clear:c	1495
\c_xeCJK_ps_halfwidth_tl	1058, 1083, 1181	\clist_clear:N	1407, 1415, 1497, 1498, 1863, 1864

\clist_gclear_new:c	154, 167	\cs_if_free:N	2204
\clist_gput_right:cx	185	\cs_meaning:c	69
\clist_gput_right:Nn	904	\cs_new_eq:NN	1748, 1877, 1885, 1921, 2102, 2114, 2183, 2184
\clist_gput_right:Nx	172, 790, 1375	\cs_new_nopar:Npn	71, 94, 105, 128, 129, 130, 131, 132, 133, 134, 135, 177, 298, 1439, 1645, 2081, 2082
\clist_gremove_all:Nn	796	\cs_new_protected:Npn	47, 49, 51, 82, 90, 92
\clist_gremove_duplicates:N	791	\cs_new_protected_nopar:Npn	43, 103, 151, 170, 183, 203, 215, 295, 300, 302, 307, 313, 318, 376, 384, 441, 468, 475, 484, 487, 493, 499, 511, 526, 534, 544, 553, 568, 574, 633, 658, 701, 717, 750, 760, 778, 806, 811, 841, 873, 912, 984, 991, 998, 1004, 1017, 1141, 1155, 1207, 1222, 1248, 1290, 1319, 1373, 1404, 1425, 1433, 1487, 1509, 1540, 1550, 1555, 1567, 1572, 1577, 1579, 1612, 1636, 1652, 1670, 1682, 1725, 1737, 1749, 1922, 1969, 1978, 2103, 2133, 2157, 2218, 2223, 2228
\clist_gset:Nn	1836	\cs_new_protected_nopar:Npx	1816
\clist_if_empty:cF	1656	\cs_set:Npn	589, 606
\clist_if_empty:N	892	\cs_set_eq:NN	84, 98, 596, 601, 608, 626, 631, 689, 690, 698, 981, 983, 1278, 1279, 1287, 1899, 1993, 1994, 2087, 2120, 2172, 2195, 2196, 2197, 2198, 2204, 2247, 2248
\clist_if_in:NnF	1867	\cs_set_nopar:cpn	2269
\clist_if_in:NnT	1848	\cs_set_nopar:Npn	2261
\clist_if_in:NnTF	780	\cs_set_nopar:Npx	2208
\clist_map_break:	1859	\cs_set_protected_nopar:Npn	2115, 2173, 2249, 2254, 2268, 2270
\clist_map_function:nN	176	\cs_to_str:N	1813
\clist_map_inline:cn	1939		
\clist_map_inline:Nn	166, 320, 454, 875, 894, 934, 986, 1133, 1238, 1499, 1673, 1685, 1846, 1865, 2136		
\clist_map_inline:nn	197, 323, 325, 337, 340, 364, 442, 463, 465, 906, 916, 1023, 1592, 1937, 2234		
\clist_map_inline:xn	186, 795, 843, 1325		
\clist_new:N	40, 41, 42, 168, 805, 931, 1372, 1834		
\clist_put_left:cx	1329, 1569, 1574		
\clist_put_left:N	1870		
\clist_put_right:Nn	1868		
\clist_put_right:Nx	1387, 1658, 1663, 1665		
\clist_remove_all:Nn	1675, 1676, 1695, 1850		
\clist_remove_duplicates:N	1684		
\clist_set:cx	1422		
\clist_set:Nn	161, 932, 985		
\clist_set:Nx	1435, 1514, 1844, 1857		
\cs_end:	73		
\cs_generate_variant:Nn	175, 197, 297, 312, 486, 799, 800, 801, 802, 803, 936, 1012, 1013, 1014, 1015, 1113, 1132, 1206, 1371, 1403, 1539, 1548, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1735, 1780		
\cs_gset_eq:cc	1732		
\cs_gset_eq:cN	1745		
\cs_gset_eq:NN	808, 809, 813		
\cs_gset_protected_nopar:cpx	1521, 1527, 1698, 1707		
\cs_if_exist:cF	654, 671		
\cs_if_exist:cTF	1783		
\cs_if_exist:NT	2243, 2259		
\cs_if_exist_use:cF	1042, 1739		
\cs_if_free:cT	863, 1823		
\cs_if_free:cTF	845, 1772		
\cs_if_free:NT	27, 1888, 1889, 1890, 1891, 2203		
		D	
		\DeclareSymbolFont	1933
		\defaultCJKfontfeatures	6, 1834, 1835, 1837, 1838
		\dim_add:Nc	1013, 1160, 1162
		\dim_add:Nn	1014, 1088, 1095, 1096, 1229
		\dim_compare:nNnT	1173, 1189
		\dim_compare:nNnTF	1053, 1070
		\dim_compare_p:nNn	403
		\dim_const:cn	571
		\dim_eval:c	471, 479, 485, 486
		\dim_eval:n	480, 481, 486, 1231
		\dim_gset:cn	571
		\dim_gset:Nn	1982
		\dim_if_exist:cTF	570
		\dim_new:N	35, 36, 37, 612, 1962
		\dim_set:Nc	1013, 1037, 1038, 1149, 1174, 1184, 1191, 1193

\dim_set:Nn	1013, 1040, 1051, 1067, 1072, 1185, 1209, 1250	\fp_use:N	1484, 1570, 1575, 1967
\dim_set_eq:NN	1079, 1100, 1226	G	
\dim_set_max:Nn	1103, 1195, 1202, 1204	\g_fontspec_encoding_tl	1530, 1710, 1933, 1935
\dim_set_min:Nc	1013, 1192, 1194	\g_xeCJK_after_end_preamble_hook	46, 52, 55
\dim_set_min:Nn	1015, 1080, 1089, 1097	\g_xeCJK_after_preamble_hook	45, 50, 53
\dim_to_fp:n	1974	\g_xeCJK_at_end_preamble_hook	44, 48, 54
\dim_use:c	565, 576, 577	\g_xeCJK_AutoFakeBold_bool	1336, 1344, 1346, 1349, 1504
\dim_use:N	1106, 1108, 1243	\g_xeCJK_AutoFakeSlant_bool	1337, 1355, 1357, 1360, 1505
\dim_zero:N	1032, 1033, 1034, 1145	\g_xeCJK_base_class_clist	161, 161, 166, 454, 875
\document	54, 55	\g_xeCJK_CJK_class_clist	161, 168, 172, 320, 2136
E		\g_xeCJK_CJK_class_prop	161, 169, 173, 207
xeCJKactive	3	\g_xeCJK_CJK_subclass_clist	804, 805, 892, 894, 904, 1673, 1846
\else:	74, 99, 109, 117, 119, 566	\g_xeCJK_class_seq	151, 155, 157, 2124, 2128
\EmboldenFactor	1340	\g_xeCJK_default_features_clist	1517, 1834, 1836
\end	772, 773	\g_xeCJK_EmboldenFactor_fp	1338, 1350, 1364, 1465, 1506
\etex_fontcharwd:D	1231	\g_xeCJK_exspace_dim	1961, 1962, 1982, 1999, 2000
\etex_lastnodetype:D	347, 393, 401, 410, 502	\g_xeCJK_family_name_prop	1525, 1544, 1667, 1668, 1702, 1705, 1730, 1775, 1931
\etex_protected:D	1895, 1896, 1897, 1898	\g_xeCJK_family_options_prop	1667, 1669, 1679, 1693, 1696, 1730, 1862
\exp_after:wN	98, 100, 108, 110, 711, 1411	\g_xeCJK_family_prop	1310, 1323, 1542, 1667, 1667, 1678, 1691, 1692, 1730, 1841, 1845, 1851, 1853, 1950
\exp_args:Nc	153, 1604	\g_xeCJK_family_spec_tl	1826, 1828, 1829
\exp_args:NNc	710, 711	\g_xeCJK_features_id_prop	1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1492, 1578, 1654
\exp_args:Nnnx	2140, 2142	\g_xeCJK_fontspec_int	1490, 1818, 1819, 1825, 1827
\exp_args:No	710, 1409, 1427	\g_xeCJK_indent_bool	2031, 2035, 2085
\exp_args:Nv	1831	\g_xeCJK_inline_env_clist	780, 787, 790, 791, 796
\exp_args:Nx	1315, 2010	\g_xeCJK_math_bool	1909, 1952
\exp_last_unbraced:Nf	68	\g_xeCJK_monoscale_fp	1484, 1961, 1961, 1967, 1973
\exp_last_unbraced:NNo	1429	\g_xeCJK_number_bool	2034, 2265
\exp_last_unbraced:NNV	1745	\g_xeCJK_punct_width_tl	978, 1044, 1046
\exp_not:N	107, 654, 671, 1525, 1526, 1530, 1531, 1532, 1702, 1703, 1705, 1706, 1710, 1711, 1712, 1816, 1872	\g_xeCJK_punctstyle_clist	931, 931, 932, 934, 947, 961, 1238
\exp_not:n	305, 310, 1523, 1528, 1708, 2210	\g_xeCJK_SlantFactor_fp	1339, 1361, 1365, 1474, 1507
\exp_not:V	947	\g_xeCJK_special_ps_clist	985, 985, 986, 1133
F		\g_xeCJK_sub_key_clist	1372, 1372, 1375, 1499
\f@series	1736		
\f@shape	1736		
\f@size	1736, 1974, 1983		
\FallBack	1438		
\fi:	74, 101, 111, 119, 120, 566		
\file_input:n	2214		
\font_glyph_if_exist:NnTF	1292		
\fontencoding	1530, 1710, 2178		
\fontfamily	1531, 1711, 1972, 1981, 2178		
\fontseries	2178		
\fontshape	2178		
\fontspec_set_family:Nnn	1523		
\fp_gset:Nn	1973		
\fp_new:N	1338, 1339, 1961		
\fp_set:Nn	1350, 1361, 1463, 1472		
\fp_set_eq:NN	1506, 1507		

\g_xeCJK_unknown_family_seq	1786, 1788, 1794	CJKspace	3
\group_align_safe_begin:	87		
\group_align_safe_end:	85, 86		
\group_begin:	54, 56,		
	1321, 1511, 1672, 1843, 1971, 1980, 2183		
\group_end:	54, 65,		
	1334, 1537, 1680, 1872, 1975, 1984, 2184		
H			
CheckSingle	4		
I			
MiddlePunct	5		
MiddlePunct+	5		
MiddlePunct-	5		
\if_catcode:w	107		
\if_cs_exist:w	73		
\if_dim:w	565		
\if_int_compare:w	115, 118, 211, 1917		
\if_meaning:w	96		
\IfBooleanF	201		
\IfBooleanT	181, 819, 829, 838		
\IfBooleanTF	2023, 2026		
\IfNoValueTF	223, 224, 1812		
\IfValueT	2207		
\indentfirst	2031		
\InlineEnv	785		
\InlineEnv+	785		
\InlineEnv-	785		
\int_compare:nNnF	502, 2123		
\int_compare:nNnTF	2117		
\int_compare_p:nNn	347, 393, 401, 410		
\int_const:Nn	146, 147, 148, 149, 150		
\int_eval:n	1259, 2271		
\int_gdecr:N	1819		
\int_gincr:N	1490, 1825		
\int_incr:N	212, 1918		
\int_max:nn	224		
\int_min:nn	223		
\int_new:N	32, 33, 34, 1818, 2156		
\int_set:Nn	206, 219, 223, 224, 1913, 2135		
\int_set_eq:NN	220		
\int_step_inline:nnnn	2126		
\int_to_hexadecimal:n	1311		
\int_use:N	1827		
\ItalicFeatures	1439		
\ItalicFont	1439		
J			
CJKecglue	4		
CJKglue	3		
CJKmath	3		
CJKnumber	3		
K			
\KaiMingPunct	963		
\keys_define:nn	136, 584,		
	585, 614, 675, 785, 936, 937, 948, 963,		
	1264, 1340, 1376, 1449, 1455, 1909, 2032		
\keys_define:nx	936, 946		
\keys_set:nn	2063, 2092		
\keys_set_known:nnN	1371		
\keys_set_known:nxN	1371, 1516		
L			
\l_keys_module_tl	2060		
\l_keys_choice_tl	941		
\l_keys_key_tl	2054, 2055		
\l_keys_value_tl	955, 1350, 1361, 1463, 1472		
\l_peek_token	96, 107, 754, 755		
\l_xeCJK_add_font_prop	1491, 1552, 1557, 1662		
\l_xeCJK_AutoFakeBold_bool	1459, 1462, 1504, 1615, 1628		
\l_xeCJK_AutoFakeSlant_bool	1468, 1471, 1505, 1617		
\l_xeCJK_checksingle_bool	683, 686, 688, 695, 697		
\l_xeCJK_CJKfontspec_id_tl	1822, 1823, 1829, 1831		
\l_xeCJK_current_coor_tl	1741, 1745		
\l_xeCJK_ecglue_dim	403, 590, 607, 612		
\l_xeCJK_EmboldenFactor_fp	1463, 1506, 1570		
\l_xeCJK_fallback_bool	1272, 1275, 1277, 1284, 1286		
\l_xeCJK_family_fb_tl	1322,		
	1323, 1327, 1328, 1329, 1330, 1331, 1332		
\l_xeCJK_family_name_tl	1775		
\l_xeCJK_family_tl	1384, 1513, 1520, 1521,		
	1526, 1527, 1678, 1679, 1687, 1691,		
	1693, 1700, 1703, 1813, 1814, 1815, 1816		
\l_xeCJK_fixed_spacing_bool	1988, 1990, 2007, 2009, 2019		
\l_xeCJK_fixed_xecglue_bool	1992, 2015		
\l_xeCJK_flexible_punctstyle_tl	1991, 2012		
\l_xeCJK_font_coor_tl	1016, 1242, 1736, 1736, 1739, 1741		
\l_xeCJK_fontname_bf_tl	1595		
\l_xeCJK_fontname_tl	1323, 1324, 1420, 1421, 1515, 1524, 1678		
\l_xeCJK_fontoptions_clist	1514, 1675, 1676, 1679		
\l_xeCJK_fontspec_family_tl	1523, 1526, 1531		
\l_xeCJK_lastpunct_tl	489, 495, 496, 497, 523, 530, 531, 538,		
	539, 541, 546, 548, 549, 555, 557, 558, 560		

_xeCJK_math_family_tl ..	1931, 1934, 1936	_xeCJK_tmpc_dim	35, 37,
_xeCJK_monoscale_tl			1034, 1040, 1051, 1053, 1054, 1067,
	1479, 1484, 1489, 1664, 1665		1070, 1072, 1074, 1079, 1080, 1088,
_xeCJK_pass_features_clist			1089, 1095, 1096, 1097, 1100, 1103, 1109
	1497, 1517, 1524, 1658, 1663, 1665	_xeCJK_tmpc_int	
_xeCJK_peek_search_token ...	81, 84, 107		32, 34, 206, 207, 210, 1913, 1916
_xeCJK_punct_coor_tl	471, 479, 480, 481,	_xeCJK_xecglue_bool	
	485, 1016, 1016, 1019, 1024, 1025,		346, 400, 595, 600, 605, 613, 1992
	1028, 1037, 1039, 1065, 1075, 1105,	SlantFactor	4
	1107, 1109, 1118, 1125, 1126, 1127,	AllowBreakBetweenPuncts	5
	1128, 1143, 1152, 1160, 1162, 1174,	_LongPunct	963
	1184, 1191, 1192, 1193, 1194, 1212,	_loop	208, 209, 1914
	1214, 1216, 1217, 1219, 1225, 1228, 1230		
_xeCJK_punctstyle_tl			
	941, 942, 1016, 1021,		
	1056, 1081, 1146, 1157, 1165, 1181, 1991		
_xeCJK_SlantFactor_fp ..	1472, 1507, 1575		
_xeCJK_space_bool	421, 625, 630		
_xeCJK_space_glue_tl	348, 402, 428		
_xeCJK_sub_cancel_bool			
	822, 825, 827, 834, 836, 849		
_xeCJK_sub_family_tl	1687,		
	1690, 1692, 1696, 1698, 1706, 1707, 1720		
_xeCJK_sub_fontfeat_clist			
	1407, 1415, 1422, 1435		
_xeCJK_sub_fontname_tl	1412,		
	1414, 1418, 1419, 1420, 1421, 1423, 1436		
_xeCJK_sub_key_clist	1387, 1498, 1684, 1685		
_xeCJK_tmpa_clist			
	40, 40, 1693, 1695, 1696, 1844,		
	1848, 1850, 1857, 1858, 1867, 1870, 1872		
_xeCJK_tmpa_dim	35, 35, 1032, 1037, 1079,		
	1088, 1089, 1095, 1096, 1097, 1100,		
	1106, 1145, 1149, 1152, 1160, 1162,		
	1166, 1172, 1173, 1174, 1184, 1185,		
	1188, 1189, 1191, 1192, 1195, 1202,		
	1204, 1224, 1225, 1226, 1227, 1228, 1229		
_xeCJK_tmpa_int	32, 32, 205,		
	208, 210, 211, 212, 1912, 1916, 1917, 1918		
_xeCJK_tmpa_tl			
	38, 38, 1406, 1408, 1409, 1411, 1414,		
	1418, 1542, 1544, 1545, 1691, 1692,		
	1703, 1706, 1711, 1731, 1757, 1763,		
	1845, 1855, 1872, 1925, 1928, 1929, 1931		
_xeCJK_tmpl_clist	40, 41, 1862, 1863, 1865		
_xeCJK_tmpl_dim			
	35, 36, 1033, 1038, 1074, 1080, 1108,		
	1193, 1194, 1195, 1226, 1229, 1231, 1243		
_xeCJK_tmpl_int	32, 33, 205, 211, 1912, 1917		
_xeCJK_tmpl_tl	38, 39, 1851, 1854, 1855, 1858		
_xeCJK_tmpl_clist	40, 42, 1864, 1868, 1870		

O	
LongPunct	5
LongPunct+	5
LongPunct-	5
Mono	6
P	
\p@	1983
\par	754
\PassOptionsToPackage	2042, 2049, 2055
\pdfstringdefPreHook	2190, 2193
\pdftex_strcmp:D	115, 118
\peek_after:Nw	88, 104
\peek_catcode:NTF	368, 635, 660, 703, 719, 764
\peek_catcode_remove:NTF	762
\peek_meaning_remove:NTF	769, 772
\Pifont	2176
\prg_do_nothing: ..	809, 1083, 2195, 2196, 2197
\prg_new_conditional:cpnn	1135
\prg_new_conditional:Npnn	66, 113, 122, 563, 1116, 1121, 1132, 1602, 1607, 1647
\prg_new_protected_conditional:Npnn	1770
\prg_return_false: ...	74, 119, 125, 566, 1119, 1130, 1138, 1605, 1610, 1650, 1773
\prg_return_true: ..	74, 116, 119, 125, 566, 1119, 1130, 1138, 1605, 1610, 1650, 1777
\ProcessKeysOptions	2083
\prop_clear:c	994
\prop_clear:N	1491
\prop_get:Nn	1440, 1563
\prop_get:NnNF	1564
\prop_get:NnNT	1542, 1702, 1731
\prop_get:NnNTF	1780
\prop_get:Nv	1310
\prop_get:NvN	1845
\prop_get:NvNF	1323, 1862
\prop_get:NvNT	1691, 1693, 1931
\prop_get:NxNF	1775, 1851, 1853
\prop_gpop:NnNT	1544
\prop_gput:Nnn	175
\prop_gput:NnV	1525, 1705, 1731
\prop_gput:NvV	1678, 1679, 1692, 1696
\prop_gput:Nvx	173
\prop_if_empty:NTF	1950
\prop_if_in:cnTF	1137
\prop_if_in:NnF	1565
\prop_if_in:NvTF	207, 1841
\prop_map_inline:Nn	1447, 1492, 1578, 1654, 1662
\prop_new:c	989
\prop_new:N	169, 1441, 1667, 1668, 1669
\prop_put:cn	996, 1002
\prop_put:Nnn	1442, 1443, 1444, 1445, 1446, 1562
\prop_put:Nxx	1552
\prop_put_if_new:Nnn	1566
\prop_put_if_new:Nxx	1557
\prop_remove:cn	1009
\PunctStyle	937
\punctstyle	30, 2099, 2099, 2100
\PunctWidth	963
Q	
\q_no_value	1929
\q_stop	69, 71
R	
\r	2172, 2173
\RenewDocumentCommand	2176
\repeat	213, 1919
\RequirePackage	20, 21, 2084, 2272
\reset@font	1899
\rmfamily	1896
S	
\scan_stop:	473, 783, 2105
\selectfont	1532, 1712, 1972, 1981, 2179
\seq_count:N	2124
\seq_gput_right:Nv	155
\seq_gput_right:Nx	1788
\seq_if_in:NnF	2128
\seq_if_in:NxF	1786
\seq_new:N	157, 1794
\setCJKfallbackfamilyfont ..	7, 8, 1313, 1313
\setCJKfamilyfont	6, 1805, 1808, 1808
\setCJKmainfont ..	5, 1801, 1883, 1883, 1885, 1901
\setCJKmathfont	7, 1902, 1906, 1906
\setCJKmonoexspace	6, 1977, 1977
\setCJKmonofont ...	5, 1803, 1904, 2021, 2021
\setCJKmonoscale	6, 1963, 1963
\setCJKromanfont	1885, 1905
\setCJKsansfont ...	5, 1802, 1883, 1886, 1903
\SetSymbolFont	1935
\settowidth	590, 607
\sffamily	1897
\shapedefault	1934, 1936
\skip_horizontal:N	984, 1999
\skip_horizontal:n	477, 2000, 2065
\skip_if_eq_p:nn	348, 402
\SlantedFeatures	1439
\SlantedFont	1439
\SlantFactor	1340
\SplitArgument	196
\str_case:x:nnn	1253, 1799

<code>\str_if_eq_x:nnF</code>	<code>\tl_if_in:nnTF</code>	1315
..... 188, 361, 372, 1240, 1751, 1755	<code>\tl_if_single_token_p:n</code>	710
<code>\str_if_eq_x:nnT</code>	<code>\tl_map_function:nN</code>	62
..... 1590	<code>\tl_map_inline:cn</code>	995, 1001
<code>\str_if_eq_x:nnTF</code>	<code>\tl_map_inline:nn</code> ..	61, 291, 1012, 1729, 2161
..... 879, 918, 1039, 1388, 1614, 1646	<code>\tl_map_inline:xn</code>	1006, 1115
<code>\str_if_eq_x_p:nn</code> 457, 1235, 1609, 1621, 1622	<code>\tl_new:c</code>	76, 77, 78, 79, 988, 1892
	<code>\tl_new:N</code>	38, 39, 44, 45, 46, 804
T	<code>\tl_put_left:Nn</code>	54, 2158
<code>\tex_afterassignment:D</code>	<code>\tl_put_right:cx</code>	1000
<code>\tex_char:D</code>	<code>\tl_put_right:Nn</code> ...	55, 914, 1297, 1327,
<code>\tex_font:D</code>	1895, 1896, 1897, 1898, 2020, 2158, 2193	
430, 431, 432, 1231, 1292, 1745, 1974, 1983	<code>\tl_put_right:Nx</code>	1384, 1758
<code>\tex_fontdimen:D</code> ..	<code>\tl_remove_all:cn</code>	1008
430, 431, 432, 1974, 1983	<code>\tl_remove_all:Nn</code>	1754
<code>\tex_ignorespaces:D</code>	<code>\tl_replace_all:Nnx</code>	1421
.. 55, 452, 1792, 1832, 1875, 2093, 2221	<code>\tl_set:cn</code>	2202
<code>\tex_kern:D</code>	<code>\tl_set:cx</code>	993, 1423
485, 576, 577	<code>\tl_set:Nn</code> ..	85, 86, 428, 1016, 1429, 1736,
<code>\tex_lastkern:D</code>	1888, 1889, 1890, 1891, 2267, 2274, 2275	
565	<code>\tl_set:Nx</code>	489, 541, 560, 941,
<code>\tex_lastskip:D</code>	1322, 1406, 1414, 1418, 1420, 1436,	
348, 402, 403	1484, 1513, 1515, 1687, 1741, 1757,	
<code>\tex_penalty:D</code>	1784, 1813, 1814, 1822, 1855, 1925, 1928	
43	<code>\tl_set_eq:NN</code>	1929, 1991
<code>\tex_the:D</code>	<code>\tl_to_lowercase:n</code>	63
299	<code>\tl_to_str:N</code>	1880
<code>\tex_unkern:D</code>	<code>\tl_use:c</code> ..	177, 480, 481, 850, 1065, 1075,
638, 663, 2108	1212, 1214, 1216, 1217, 1219, 1646, 1659	
<code>\tex_unskip:D</code>	<code>\token_get_arg_spec:c</code>	709
351, 406	<code>\token_get_arg_spec:N</code>	801
<code>\tex_vrule:D</code>	<code>\token_if_cs_p:c</code>	645, 727, 739
470	<code>\token_if_cs_p:N</code>	799
<code>\textcentereddot</code>	<code>\token_if_eq_meaning_p:NN</code>	754, 755
2163	<code>\token_if_math_toggle:cTF</code>	653, 670
<code>\textellipsis</code>	<code>\token_if_math_toggle:NTF</code>	803
2163	<code>\token_if_math_toggle_p:c</code>	726, 738
<code>\textemdash</code>	<code>\token_if_math_toggle_p:N</code>	802
2163	<code>\token_if_other_p:N</code>	711
<code>\textperiodcentered</code>	<code>\token_if_space_p:c</code>	725, 730
2163	<code>\token_if_space_p:N</code>	800
<code>\textquotedblleft</code>	<code>\token_new:Nn</code>	81
2164	<code>\token_to_str:N</code>	
<code>\textquotedblright</code> 24, 857, 1801, 1802, 1803, 1805, 1958	
2164	<code>\TrimSpaces</code>	178, 861
<code>\textquoteright</code>	<code>\ttdefault</code>	1972, 1981
2164	<code>\ttfamily</code>	1898
<code>\tipaencoding</code>		
2171	U	
<code>\tl_case:Nnn</code>	<code>\UL@hook</code>	2243, 2245
1056, 1081	<code>PunctStyle</code>	4
<code>\tl_clear:c</code>	<code>PunctWidth</code>	5
1494	<code>\Unicode</code>	2270
<code>\tl_clear:N</code>		
1324, 1479, 1489		
<code>\tl_const:cn</code>		
935		
<code>\tl_const:Nn</code>		
1908		
<code>\tl_const:Nx</code>		
127		
<code>\tl_gput_right:Nn</code>		
48, 50, 52		
<code>\tl_gset:cn</code>		
1115, 1263		
<code>\tl_gset:cV</code> 1024, 1025, 1109, 1152, 1225, 1228		
<code>\tl_gset:cx</code>		
1105, 1107, 1230, 1242, 1829		
<code>\tl_gset:Nx</code>		
523, 1826		
<code>\tl_if_eq:NNF</code>		
1146		
<code>\tl_if_eq:NNT</code>		
942		
<code>\tl_if_eq:NNTF</code>		
1021, 1157, 1165		
<code>\tl_if_eq_p:NN</code>		
1181		
<code>\tl_if_exist:cF</code>		
1019, 1028, 1143		
<code>\tl_if_exist:cTF</code>		
124, 1118, 1148		
<code>\tl_if_exist_p:c</code>		
1125, 1126, 1127, 1128		
<code>\tl_if_exist_p:N</code>		
2190		
<code>\tl_if_head_eq_charcode:nNTF</code>		
1409		
<code>\tl_if_head_is_group:nT</code>		
1427		

\use:c	449, 1743, 1776, 1916, 2273	\xeCJK_checksingl:N	690, 701, 701
\use:x	1871	\xeCJK_checksingl:NN ...	701, 704, 713, 717
\use_none:n	2198	\xeCJK_checksingl:NNN ..	701, 741, 745, 750
\use_none:nnnn	1744	\xeCJK_checksingl_env:NNN ..	758, 760, 760
AutoFakeBold	4, 5	\xeCJK_checksingl_env:NNNN	760, 770, 773, 778
AutoFakeSlant	4, 5	\xeCJK_checksingl_save_CJKsymbol:N ..	689,
AutoFallBack	4	698, 714, 720, 733, 742, 747, 766, 774, 781
V			
\verbatim@font	2020	\xeCJK_CJK_and_CJK:N	440, 441, 441, 689, 690, 698
X			
\xCJKecglue	585	\xeCJK_CJK_and_FullLeft:N	499, 499
\xe@alloc@intercharclass	2124, 2126	\xeCJK_CJK_and_FullRight:N	511, 511
\xeCJK@family 1295, 1297, 1309, 1310, 1736,	1742, 1743, 1754, 1757, 1758, 1759,	\xeCJK_CJK_class	146, 147
1762, 1763, 1784, 1841, 1845, 1851, 1862		\xeCJK_class_num:n 177, 177, 191, 292, 296, 299	
\xeCJK@setfont	1748	\xeCJK_clear_CJK_toks: 318, 318, 332, 380, 447	
\xeCJK_add_fake_bold:n 1567, 1567, 1615, 1632		\xeCJK_clear_inter_class_toks:nn ...	300, 300, 321, 331
\xeCJK_add_fake_slant:n .. 1567, 1572, 1624		\xeCJK_copy_family:nn 1725, 1725, 1735	
\xeCJK_add_font:nn	1550, 1550, 1560	\xeCJK_copy_family:xx	1762, 1763
\xeCJK_add_font:nV	1595	\xeCJK_copy_inter_class_toks:nnnn ...	313, 313, 877,
\xeCJK_add_font:nv	1583	878, 891, 896, 897, 2138, 2139, 2146, 2151
\xeCJK_add_font:nx	1640	\xeCJK_declare_char_class:nn	180, 183, 229, 231, 233, 240, 254, 848, 870
\xeCJK_add_font_if_new:nn	1550, 1555, 1561, 1634	\xeCJK_def_node:nn	568, 568, 579, 580, 581, 582, 583
\xeCJK_add_font_if_new:nv	1587	\xeCJK_Default_class	146, 146
\xeCJK_add_font_if_new:nx	1642	\xeCJK_erase_CJKsymbol: . 806, 806, 886, 923	
\xeCJK_add_special_punct:nn	970, 973, 975, 985, 998	\xeCJK_error:n	128, 129
\xeCJK_add_sub_block: 1536, 1682, 1682		\xeCJK_error:nx 128, 130, 846, 955, 2054	
\xeCJK_after_FullRight: 460, 493, 493		\xeCJK_fallback_save_CJKsymbol:N ...	1278, 1287, 1293, 1303
\xeCJK_AfterEndPreamble:n 44, 51, 2121, 2185		\xeCJK_fallback_testsymbol:N	1279, 1290, 1290, 1299
\xeCJK_AfterPreamble:n	30, 44, 49, 2020, 2159, 2241	\xeCJK_family_if_exist:x	1770
\xeCJK_app_inter_class_toks:nnc . 307, 338		\xeCJK_family_if_exist:xF	1759
\xeCJK_app_inter_class_toks:nnn	307, 307, 312, 336, 362, 887, 2140	\xeCJK_family_if_exist:xF 1700, 1727	
\xeCJK_AtEndPreamble:n	44, 47, 1947	\xeCJK_family_if_exist:xF	1295, 1742, 1761, 1770, 1924, 1927
\xeCJK_aux_r	2172, 2173	\xeCJK_fix_itcorr:	2103, 2103, 2118
\xeCJK_before_FullLeft:N	487, 487, 504, 532, 551	\xeCJK_flexible_cjkg glue: 1994, 2013	
\xeCJK_Boundary_and_CJK:	375, 376, 376, 924, 2233	\xeCJK_flexible_ecglue:	1993, 2014
\xeCJK_Boundary_class	146, 150	\xeCJK_FullLeft_and_FullLeft:N .. 526, 526	
\xeCJK_calc_kern:NN	1141, 1150, 1155	\xeCJK_FullLeft_and_FullRight:N . 534, 534	
\xeCJK_calc_kern:NnNN	1166, 1172, 1188, 1207, 1207	\xeCJK_FullLeft_class	146, 148
\xeCJK_check_family:n 1540, 1540, 1548		\xeCJK_FullRight_and_FullLeft:N . 544, 544	
\xeCJK_check_family:V	1520, 1690	\xeCJK_FullRight_and_FullRight:N 553, 553	
\xeCJK_check_for_glue:	376, 378, 384	\xeCJK_FullRight_class	146, 149
\xeCJK_check_num_range:nnNN	205, 215, 215, 1912	\xeCJK_get_glyph_bounds:nNN	1224, 1227, 1248, 1248

\xeCJK_get_inter_class_toks:nn .. 298 ,	\xeCJK_if_package_loaded:nTF ... 122 , 2053
298 , 305 , 310 , 316 , 2141 , 2143 , 2144 , 2149	\xeCJK_if_package_loaded_p:n ... 122 , 2189
\xeCJK_get_kern:NN 1141 , 1141 , 1206	\xeCJK_ignore_spaces: 631 , 633 , 633
\xeCJK_get_kern:VN 530 , 538 , 548 , 557	\xeCJK_ignorespaces: 438 , 626 , 631 , 2226 , 2252
\xeCJK_get_map_font:n 1640 , 1642 , 1645 , 1645 , 1649	\xeCJK_info:nx 128 , 134
\xeCJK_get_punct_bounds:nN 501 ,	\xeCJK_info:nxx 128 , 135 , 1753
513 , 529 , 537 , 547 , 556 , 1017 , 1017 , 1113	\xeCJK_inter_class_toks:nnc 295 , 466
\xeCJK_get_punct_bounds:nV 495	\xeCJK_inter_class_toks:nnn 295 , 295 , 297 , 301 , 327 , 334 , 342 , 366 ,
\xeCJK_get_punct_dimen:N . 1029 , 1222 , 1222	375 , 434 , 440 , 444 , 451 , 452 , 919 , 920 ,
\xeCJK_get_sub_features:nn 1328 , 1396 , 1404 , 1425	921 , 2230 , 2231 , 2232 , 2233 , 2236 , 2237
\xeCJK_get_sub_features_aux:n 1411 , 1425 , 1433	\xeCJK_inter_class_toks:nxx 295 , 304 , 309 , 315
\xeCJK_gobble_brace:N 1408 , 1425 , 1425 , 1430	\xeCJK_itcorr_aux ... 2109 , 2112 , 2114 , 2118
\xeCJK_group_begin: 2168 , 2183 , 2196	\xeCJK_make_node:n 369 ,
\xeCJK_group_end: 2169 , 2184 , 2197	370 , 437 , 568 , 574 , 638 , 663 , 2110 , 2252
\xeCJK_HalfLeft_class 158	\xeCJK_map_features_id:n . 1439 , 1553 , 1558
\xeCJK_HalfRight_class 158	\xeCJK_map_it_sl:n 1626 , 1636 , 1636
\xeCJK_if_blank_x:n 113	\xeCJK_math_type:n 1913 , 1921
\xeCJK_if_blank_x:nF 1664	\xeCJK_msg_new:nn ... 128 , 128 , 854 , 957 ,
\xeCJK_if_blank_x:nT 1412 , 2144 , 2149	1307 , 1549 , 1769 , 1795 , 1878 , 1954 , 2058
\xeCJK_if_blank_x:nTF 113 , 219 , 1044 , 1380 , 1419 , 1604 , 1649	\xeCJK_new_class:n 151 , 151 , 158 , 159 , 160 , 865
\xeCJK_if_blank_x:p:n 113 , 217 , 709	\xeCJK_new_sub_key:n 867 , 1372 , 1373 , 1403 , 1438
\xeCJK_if_font_select:n 1602	\xeCJK_no_break: 43 , 43 , 459 , 516 ,
\xeCJK_if_font_select:nF . 1586 , 1594 , 1641	528 , 536 , 540 , 559 , 757 , 765 , 782 , 983 , 2220
\xeCJK_if_font_select:nTF 1581 , 1602	\xeCJK_NormalSpace_class 158
\xeCJK_if_font_select_p:n 1602 , 1621 , 1622 , 1630	\xeCJK_Others_class 2121 , 2135 , 2156
\xeCJK_if_it_or_sl:n 1607	\xeCJK_parse_features: ... 1518 , 1577 , 1577
\xeCJK_if_it_or_sl:nT 1626	\xeCJK_parse_features:n .. 1577 , 1578 , 1579
\xeCJK_if_it_or_sl:nTF 1584 , 1602	\xeCJK_pass_features: 1519 , 1652 , 1652
\xeCJK_if_it_or_sl_p:n ... 1602 , 1620 , 1629	\xeCJK_patch:Nnn 2157 , 2157 , 2167 , 2171
\xeCJK_if_last_node:n 563	\xeCJK_peek_catcode_ignore_spaces:Nf 92 , 664
\xeCJK_if_last_node:nT 358 , 637 , 662	\xeCJK_peek_catcode_ignore_spaces:NT . 90
\xeCJK_if_last_node:nTF 357 , 563 , 2106	\xeCJK_peek_catcode_ignore_spaces:Ntf 81 , 82 , 91 , 93 , 639 , 732
\xeCJK_if_last_node_p:n 353 ,	\xeCJK_peek_execute_branches_catcode: 100 , 105
387 , 392 , 394 , 409 , 411 , 412 , 418 , 419 , 563	\xeCJK_peek_false:w 86 , 110
\xeCJK_if_map_font_select:n 1647	\xeCJK_peek_ignore_spaces_execute_branches: 88 , 94 , 104
\xeCJK_if_map_font_select:nT 1638	\xeCJK_peek_ignore_spaces_execute_branches_aux: 97 , 103
\xeCJK_if_map_font_select:nTF 1647	\xeCJK_peek_math: 626 , 633 , 658
\xeCJK_if_map_font_select_p:n 1647	\xeCJK_peek_tmp:w 98
\xeCJK_if_outer_macro:c 66	\xeCJK_peek_true:w 85 , 108
\xeCJK_if_outer_macro:cF 651 , 668	\xeCJK_pre_inter_class_toks:nnn 302 , 302 ,
\xeCJK_if_outer_macro:cTF 56	373 , 459 , 460 , 881 , 885 , 898 , 900 , 908 , 2142
\xeCJK_if_outer_macro_aux:w 68 , 71	\xeCJK_punct_glue:n 475 , 475 , 497 , 503 , 508 , 519
\xeCJK_if_outer_macro_p:c 56 , 644 , 708 , 724 , 737	
\xeCJK_if_package_loaded:n 122	
\xeCJK_if_package_loaded:nF ... 2041 , 2048	
\xeCJK_if_package_loaded:nT 2174	

\xeCJK_punct_if_both_left_right:NN .	1121	\xeCJK_sub_special_punct:nn	971, 974, 977, 985, 1004
\xeCJK_punct_if_both_left_right:NNT	1218	\xeCJK_switch_font:nn	882, 888, 899, 901, 909, 925, 1736, 1749
\xeCJK_punct_if_both_left_right:NNTF	1116, 1168	\xeCJK_UL_CJKpunctsymbol	2248, 2255
\xeCJK_punct_if_both_left_right_p:NN	1116	\xeCJK_UL_CJKsymbol	2247, 2251
\xeCJK_punct_if_long:NTF . .	514, 1030, 1116	\xeCJK_UL_subclass_patch:nn . .	868, 912, 912
\xeCJK_punct_if_long_p:N .	1116, 1201, 1234	\xeCJK_UL_subclass_patch_tl	804, 804, 914, 2239
\xeCJK_punct_if_middle:NT	505, 517	\xeCJK_ULpostpunctchar:n .	2218, 2223, 2237
\xeCJK_punct_if_middle:NTF . . .	1049, 1116	\xeCJK_ULprepunctchar:n . .	2218, 2218, 2236
\xeCJK_punct_if_middle_p:N	1116	\xeCJK_ULroutines:	2218, 2228, 2256
\xeCJK_punct_if_mixedwidth:NT .	1086, 1170	\xeCJK_warning:n	131, 1874, 1951
\xeCJK_punct_if_mixedwidth:NTF	1094, 1116	\xeCJK_warning:nx	128, 132, 1302, 1789
\xeCJK_punct_if_mixedwidth_p:N	1116, 1180	\xeCJK_warning:nxx	133, 1545
\xeCJK_punct_if_right:N	1116	\xeCJK_zero_glue:	981, 984
\xeCJK_punct_if_right:NF	1161	\xeCJKactive	136
\xeCJK_punct_if_right:NT	1159	\xeCJKallowbreakbetweenpuncts	943, 966, 980, 980
\xeCJK_punct_if_right:NTF	1116, 1213, 1215	\xeCJKCancelSubCJKBlock	10, 822, 823
\xeCJK_punct_if_right_p:N	1116	\xeCJKcaption	2203, 2205
\xeCJK_punct_kern:n	484, 484, 531, 539, 549, 558	\xeCJKDeclareCharClass	10, 178, 178
\xeCJK_punct_nobreak:	550, 981, 983	\xeCJKDeclareSubCharClass	818, 860, 860, 872
\xeCJK_punct_rule:n	468, 468, 490, 496, 507, 520, 546, 555	\xeCJKDeclareSubCJKBlock	9, 816, 816, 821, 857
\xeCJK_reset_at_catcode:	2208, 2215	\xeCJKdisablechecksingle	679, 683, 693
\xeCJK_restore_CJKsymbol: .	806, 811, 888, 926	\xeCJKdisablefallback	1268, 1272, 1282
\xeCJK_save_CJK_class:n	170, 176, 905	\xeCJKenablechecksingle	678, 683, 684
\xeCJK_save_family_info: .	1535, 1670, 1670	\xeCJKenablefallback	1267, 1272, 1273
\xeCJK_select_font:	330, 381, 448, 1029, 1298, 1736, 1737, 1748, 1767, 2251, 2255, 2262	\xeCJKmainfont	1958
\xeCJK_set_char_class:nnn . . .	190, 200, 203	\xeCJKnobreakbetweenpuncts . .	967, 980, 982
\xeCJK_set_char_class_aux:Nn	178, 190, 195, 1941	\xeCJKplainchr	2099, 2100
\xeCJK_set_fake:n	1600, 1612, 1612	\xeCJKResetPunctClass	10, 181, 201, 227, 227, 288, 819, 829, 838
\xeCJK_set_family:ncc	1330, 1720	\xeCJKRestoreSubCJKBlock	10, 822, 832
\xeCJK_set_family:nnn	1317, 1509, 1509, 1539, 1809, 1815, 1828, 1884, 1887, 1907, 2024, 2027, 2028	\xeCJKsetcharclass	198, 198
\xeCJK_set_family_fallback:nnn	1313, 1316, 1319	\xeCJKsetecglue	2102
\xeCJK_set_init:	1487, 1487, 1512	\xeCJKsetemboldenfactor	2095, 2095
\xeCJK_set_mathfont:	1922, 1922, 1952	\xeCJKsetkern	10, 1262, 1262
\xeCJK_set_monoexspace:	1478, 1977, 1978, 1995	\xeCJKsetmathcode	1910, 1910, 1941
\xeCJK_set_monoscale:	1483, 1966, 1969	\xeCJKsetslantfactor	2095, 2097
\xeCJK_set_others_toks:n .	2121, 2129, 2133	\xeCJKsetup	3, 1996, 2010, 2090, 2090, 2096, 2098, 2099, 2101
\xeCJK_set_special_punct:nn	969, 972, 976, 985, 991	\xeCJKsetwidth	10, 1114, 1114
\xeCJK_set_sub_class_toks:nn .	866, 873, 873	\XeTeX@CJKfntef@hook	2259, 2261
\xeCJK_space_or_xecglue:	358, 596, 601, 608, 640, 647, 665	\xetex_if_engine:F	9
\xeCJK_sub_restore_or_cancel:n	828, 837, 841	\XeTeXcharclass	210, 292
		\XeTeXcharglyph	1259
		\XeTeXdefaultencoding	2207, 2216
		\XeTeXglyphbounds	24, 27, 1252
		\XeTeXinterchartokenstate . .	143, 144, 2117
		\XeTeXinterchartoks	296, 299
		\XeTeXmathcode	1915