

xeCJK 宏包

ctex.org

2012/10/16 v3.0.9

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	43
5.10	CJK 字体族声明方式	45
5.11	字体切换	54
5.12	数学字体设置	57
5.13	处理等宽字体和抄录环境	59
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	68
6.4	xeCJK-example-CJKecglue.tex	70
6.5	xeCJK-example-checksingle.tex	70

1 简介

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

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

xeCJK 使用了 X_YTeX 的一些最新特性,需要 X_YTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 依赖 LaTeX 3 项目的 expl3, xparse 和 l3keys2e 宏包。xeCJK 还通过 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 X_YLaTeX 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 = $\langle\text{glue}\rangle$</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 = $\langle\text{true} \text{false} \text{glue}\rangle$</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 = $\langle\text{true} \text{false}\rangle$</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 $\backslash\text{begin}\{\dots\}$ 和结束 $\backslash\text{end}\{\dots\}$ 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 = $\langle\text{true} \text{false}\rangle$</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 = $\langle\text{true} \text{false} \text{数字}\rangle$</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 = $\langle\text{true} \text{false} \text{数字}\rangle$</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 = $\langle\text{数字} \text{4}\rangle$</p> <p>设置伪粗体的默认粗细程度。</p>
<hr/> SlantFactor ☆ <hr/>	<p>SlantFactor = $\langle\text{数字} \text{0.167}\rangle$</p> <p>设置伪斜体的粗细程度,范围是 $-0.999 \sim 0.999$。</p>
<hr/> PunctStyle <hr/>	<p>PunctStyle = $\langle\text{quanjiao} \text{banjiao} \text{kaiming} \text{hangmoban} \text{CCT} \text{plain}\rangle$</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>hangmoban 行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</p> <p>CCT CCT 格式;</p> <p>plain 原样(不调整标点间距)。</p>

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 = $\langle Scale \rangle Exspace$</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 $\langle family \rangle$ [$\langle font features \rangle$] $\langle font name \rangle$</code></p> <p>声明新的 <code>CJK</code> 字体族 $\langle family \rangle$ 并指定字体。</p>
<hr/> <code>\CJKfamily</code> <hr/>	<p><code>\CJKfamily $\langle family \rangle$</code></p> <p>用于在文档中切换 <code>CJK</code> 字体族,$\langle family \rangle$ 必须预先声明。</p>
<hr/> <code>\newCJKfontfamily</code> ★ <hr/>	<p><code>\newCJKfontfamily [$\langle family \rangle$] $\langle font-switch \rangle$ [$\langle font features \rangle$] $\langle font name \rangle$</code></p> <p>声明新的 <code>CJK</code> 字体族 $\langle family \rangle$ 并指定字体,并定义 $\langle font-switch \rangle$,在文档中可以使用它来切换 <code>CJK</code> 字体族。可以不必指定 $\langle family \rangle$,这时候 $\langle family \rangle$ 将等于 $\langle font-switch \rangle$。事实上,<code>\newCJKfontfamily</code> 是 <code>\setCJKfamilyfont</code> 和 <code>\CJKfamily</code> 的合并。例如</p> <p style="padding-left: 40px;"><code>\newCJKfontfamily[song]\songti{SimSun}</code></p> <p>等价于</p> <p style="padding-left: 40px;"> <code>\setCJKfamilyfont{song}{SimSun}</code> <code>\newcommand\songti{\CJKfamily{song}}</code> </p>
<hr/> <code>\CJKfontspec</code> <hr/>	<p><code>\CJKfontspec [$\langle font features \rangle$] $\langle font name \rangle$</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 $\langle font features \rangle$</code></p> <p>全局设置 <code>CJK</code> 字体族的默认选项。例如,使用</p> <p style="padding-left: 40px;"><code>\defaultCJKfontfeatures{Scale=0.962216}</code></p> <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> <p style="padding-left: 40px;"><code>\defaultCJKfontfeatures{Script=CJK}</code></p>

<hr/> <hr/>	<p><code>\addCJKfontfeatures</code> <code>\addCJKfontfeatures {}</code></p> <p>临时增加当前使用的 CJK 字体的选项。例如, 使用</p> <pre>\addCJKfontfeatures{Scale=1.1}</pre> <p>可以将文档中当前使用的 CJK 字体放大为 1.1。</p>
<hr/> <hr/>	<p><code>\CJKrmdefault</code> 保存 <code>\rmfamily</code> 所使用的 CJK 字体族, 默认值是 <code>rm</code>。类似西文字体的 <code>\rmdefault</code>。</p>
<hr/> <hr/>	<p><code>\CJKsfdefault</code> 保存 <code>\sffamily</code> 所使用的 CJK 字体族, 默认值是 <code>sf</code>。类似西文字体的 <code>\sfdefault</code>。</p>
<hr/> <hr/>	<p><code>\CJKttdefault</code> 保存 <code>\ttfamily</code> 所使用的 CJK 字体族, 默认值是 <code>tt</code>。类似西文字体的 <code>\ttdefault</code>。</p>
<hr/> <hr/>	<p><code>\CJKfamilydefault</code> 保存 <code>\normalfont</code> 所使用的 CJK 字体族, 默认值是 <code>\CJKrmdefault</code>。类似西文字体的 <code>\familydefault</code>。可以通过更改它来改变文档的默认 CJK 字体族。例如, 使用</p> <pre>\renewcommand\familydefault{\sfdefault} \renewcommand\CJKfamilydefault{\CJKsfdefault}</pre> <p>可以将全文的 CJK 和西文默认字体改为无衬线字体族。</p>
<hr/> <hr/>	<p><code>\setCJKmathfont</code> ★ <code>\setCJKmathfont [] { }</code></p> <p>设置数学公式中的 CJK 字体族。如果使用了 <code>CJKmath</code> 选项, 但是没有使用 <code>\setCJKmathfont</code> 设置数学公式中的 CJK 字体, 那么将使用 <code>\CJKfamilydefault</code> 作为数学公式中的 CJK 字体。</p>
<hr/> <hr/>	<p><code>\setCJKfallbackfamilyfont</code> ★ <code>\setCJKfallbackfamilyfont { <family> } [] { }</code></p> <p>设置 CJK 字体族 <code><family></code> 的备用字体。例如, 使用</p> <pre>\setCJKmainfont{SimSun} \setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}</pre> <p>可以将 <code>SimSun-ExtB</code> 作为 <code>SimSun</code> 的备用字体。</p>
<hr/> <hr/>	<p><code>FallBack</code> <code>FallBack = { [] { } }</code></p> <p><code>xeCJK</code> 在 <code></code> 里增加了 <code>FallBack</code> 这个选项。用来在声明主字体时, 同时设置备用字体。例如, 上面的例子等价于:</p> <pre>\setCJKmainfont[FallBack=SimSun-ExtB]{SimSun}</pre> <p>如果 <code>FallBack</code> 的值为空, 将设置的是备用字体。例如,</p> <pre>\setCJKmainfont[FallBack,AutoFakeBold,Scale=.97]{SimSun-ExtB}</pre> <p>等价于</p> <pre>\setCJKfallbackfamilyfont{\CJKrmdefault}[AutoFakeBold,Scale=.97]{SimSun-ExtB}</pre>

```
\setCJKfallbackfamilyfont ★ \setCJKfallbackfamilyfont {<family>} [<common font features>]
{
  {[<font features 1>]} {<font name 1>}},
  {[<font features 2>]} {<font name 2>}},
  .....
}
```

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

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

之后,就设置了 SimSun 是 KaiTi_GB2312 的备用字体,而 SimSun-ExtB 是 SimSun 的备用字体。

3.2.1 X_YTeX 的字体名查找

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

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

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

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

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

```
Times New Roman:style=cursiva,kurzíva,kursiv,Πλάγια,Italic,
Kursivoitu,Italique,Dőlt,Corsivo,Cursief,kursywa,Itálico,Курсив,
Ítalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta cursiva,tučné kurzíva,fed kursiv,
Fett Kursiv,Έντρονα Πλάγια,Bold Italic,Negrita Cursiva,
Lihavoitu Kursivoi,Gras Italique,Félkövéř dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrito Itálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın Ítalik,
Krepko poševno,nghiêng đâđ,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντρονα,Bold,Negrita,
Lihavoitu,Gras,Félkövéř,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
Полужирный,Fet,Kalın,Krepko,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>
2 <@@=xeCJK>
```

5.1 运行环境检查

xeCJK 必须使用 XeTeX 引擎的支持。

```
3 \msg_new:nnn { xejk } { Require-XeTeX }
4 {
5   The~xeCJK~package~requires~XeTeX~to~function.\\
6   You~must~change~your~typesetting~engine~to~"xelatex"~\\
7   instead~of~plain~"latex"~or~"pdflatex"~or~"lualatex".\\
8   Loading~xeCJK~will~abort!
9 }
10 \xetex_if_engine:F { \msg_critical:nn { xejk } { Require-XeTeX } }
    应该使用较新版本的 expl3 宏包。
```

```
11 \msg_new:nnn { xejk } { l3-too-old }
12 {
13   Support~package~'expl3'~too~old. \\
14   Please~update~an~up~to~date~version~of~the~bundles\\
15   'l3kernel'~and~'l3packages'\\
16   using~your~TeX~package~manager~or~from~CTAN.\\
17   Loading~xeCJK~will~abort!
18 }
19 \@ifpackagelater { expl3 } { 2012/09/10 } { }
20 { \msg_critical:nn { xejk } { l3-too-old } }
21 \RequirePackage{xparse}
22 \RequirePackage{l3keys2e}
```

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

```
23 \msg_new:nnn { xejk } { XeTeX-too-old }
24 {
25   \token_to_str:N \XeTeXglyphbouds \ is~not~defined.\\
26   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
27 }
28 \cs_if_free:NT \XeTeXglyphbouds
29 {
30   \msg_error:nn { xejk } { XeTeX-too-old }
31   \AtEndOfPackage { \__xeCJK_AfterPreamble:n { \punctstyle { plain } } }
32 }
```

5.2 内部工具

分配临时变量。

```
\l__xeCJK_tmpa_int
\l__xeCJK_tmppb_int 33 \int_new:N \l__xeCJK_tmpa_int
\l__xeCJK_tmppc_int 34 \int_new:N \l__xeCJK_tmppb_int
35 \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 36 \dim_new:N \l__xeCJK_tmpa_dim
\l__xeCJK_tmppc_dim 37 \dim_new:N \l__xeCJK_tmppb_dim
38 \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 39 \tl_new:N \l__xeCJK_tmpa_tl
40 \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 41 \clist_new:N \l__xeCJK_tmpa_clist
\l__xeCJK_tmppc_clist 42 \clist_new:N \l__xeCJK_tmppb_clist
43 \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:
44 \cs_new_protected_nopar:Npn \__xeCJK_no_break: { \tex_penalty:D \c_ten_thousand }
(End definition for \__xeCJK_no_break:)

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

\__xeCJK_if_outer_macro_p:c 判断是否是 \outer 宏,输入的是控制序列的名字。
\__xeCJK_if_outer_macro:cTF 57 \group_begin:
58 \char_set_lccode:nn { '\Z } { '\t }
59 \char_set_lccode:nn { '\Y } { '\c }
60 \char_set_lccode:nn { '\; } { '\: }
61 \char_set_lccode:nn { '\ / } { '\\ }
```

```

62 \tl_map_inline:nn { \T \F \C \J \K } { \char_set_lccode:nn { '#1 } { '#1 } }
63 \tl_map_function:nN { \O \U \Z \E \R \M \A \Y \R \; \ / } \char_set_catcode_other:N
64 \tl_to_lowercase:n
65 {
66   \group_end:
67   \prg_new_conditional:Npnn \__xeCJK_if_outer_macro:c #1 { p , T , F , TF }
68   {
69     \exp_last_unbraced:Nf \__xeCJK_if_outer_macro_aux:w
70     { \cs_meaning:c {#1} } OUZER ~ MAYRO ; \q_stop
71   }
72   \cs_new_nopar:Npn \__xeCJK_if_outer_macro_aux:w #1 OUZER ~ MAYRO ; #2 \q_stop
73   {
74     \if_cs_exist:w c__xeCJK_prefix_ #1 _aux_tl \cs_end:
75     \prg_return_true: \else: \prg_return_false: \fi:
76   }
77   \tl_new:c { c__xeCJK_prefix_/_aux_tl }
78   \tl_new:c { c__xeCJK_prefix_/long/_aux_tl }
79   \tl_new:c { c__xeCJK_prefix_/protected/_aux_tl }
80   \tl_new:c { c__xeCJK_prefix_/protected/long/_aux_tl }
81 }

```

(End definition for __xeCJK_if_outer_macro:c)

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

```

82 \token_new:Nn \l__xeCJK_peek_search_token { ? }
83 \cs_new_protected:Npn \__xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
84 {
85   \cs_set_eq:NN \l__xeCJK_peek_search_token #1 \prg_do_nothing:
86   \tl_set:Nn \__xeCJK_peek_true:w { \group_align_safe_end: #2 }
87   \tl_set:Nn \__xeCJK_peek_false:w { \group_align_safe_end: #3 }
88   \group_align_safe_begin:
89   \peek_after:Nw \__xeCJK_peek_ignore_spaces_execute_branches:
90 }
91 \cs_new_protected:Npn \__xeCJK_peek_catcode_ignore_spaces:NT #1#2
92 { \__xeCJK_peek_catcode_ignore_spaces:NTF #1 {#2} { } }
93 \cs_new_protected:Npn \__xeCJK_peek_catcode_ignore_spaces:NF #1#2
94 { \__xeCJK_peek_catcode_ignore_spaces:NTF #1 { } {#2} }
95 \cs_new_nopar:Npn \__xeCJK_peek_ignore_spaces_execute_branches:
96 {
97   \if_meaning:w \l_peek_token \c_space_token
98   \tex_afterassignment:D \__xeCJK_peek_ignore_spaces_execute_branches_aux:
99   \exp_after:wN \cs_set_eq:NN \exp_after:wN \__xeCJK_peek_tmp:w
100   \else:
101     \exp_after:wN \__xeCJK_peek_execute_branches_catcode:
102     \fi:
103 }
104 \cs_new_protected_nopar:Npn \__xeCJK_peek_ignore_spaces_execute_branches_aux:
105 { \peek_after:Nw \__xeCJK_peek_ignore_spaces_execute_branches: }
106 \cs_new_nopar:Npn \__xeCJK_peek_execute_branches_catcode:
107 {
108   \if_catcode:w \exp_not:N \l_peek_token \exp_not:N \l__xeCJK_peek_search_token
109   \exp_after:wN \__xeCJK_peek_true:w
110   \else:

```

```

111 \exp_after:wN \__xeCJK_peek_false:w
112 \fi:
113 }
(End definition for \__xeCJK_peek_catcode_ignore_spaces:NTF)

```

__xeCJK_if_blank_x:p:n 判断是否为空或者仅含一个空格。

```

\__xeCJK_if_blank_x:nTF 114 \prg_new_conditional:Npnn \__xeCJK_if_blank_x:n #1 { p , T , F , TF }
115 {
116 \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
117 \prg_return_true:
118 \else:
119 \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
120 \prg_return_true: \else: \prg_return_false: \fi:
121 \fi:
122 }
(End definition for \__xeCJK_if_blank_x:n)

```

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

```

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

```

__xeCJK_msg_new:nn 各种信息函数的缩略形式。

```

\__xeCJK_error:n 129 \cs_new_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:nx 130 \cs_new_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_warning:nx 131 \cs_new_nopar:Npn \__xeCJK_error:nx { \msg_error:nnx { xeCJK } }
\__xeCJK_info:nx 132 \cs_new_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\__xeCJK_info:nxx 133 \cs_new_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
134 \cs_new_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
135 \cs_new_nopar:Npn \__xeCJK_info:nx { \msg_info:nnx { xeCJK } }
136 \cs_new_nopar:Npn \__xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }
(End definition for \__xeCJK_msg_new:nn and others.)

```

5.3 功能开关

xeCJKactive

```

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

```

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

\makexeCJKactive

\makexeCJKinactive

```

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

```

(End definition for \makexeCJKactive and \makexeCJKinactive)
抑制 BOM。

146 \char_set_catcode_ignore:n { "FEFF }

5.4 字符类别设定

xeCJK 需要以下字符类别用于字符输出。其中 Default、CJK、FullLeft、FullRight、Boundary 为 Xe_{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

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

\c__xeCJK_CJK_class_int 147 \int_const:Nn \c__xeCJK_Default_class_int { 0 }
\c__xeCJK_FullLeft_class_int 148 \int_const:Nn \c__xeCJK_CJK_class_int { 1 }
\c__xeCJK_FullRight_class_int 149 \int_const:Nn \c__xeCJK_FullLeft_class_int { 2 }
\c__xeCJK_Boundary_class_int 150 \int_const:Nn \c__xeCJK_FullRight_class_int { 3 }
151 \int_const:Nn \c__xeCJK_Boundary_class_int { 255 }

(End definition for \c__xeCJK_Default_class_int and others.)

__xeCJK_new_class:n 新建一个字符类别。

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

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

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

\g__xeCJK_base_class_clist

\g__xeCJK_CJK_class_clist 162 \clist_set:Nn \g__xeCJK_base_class_clist
\g__xeCJK_CJK_class_prop 163 {
164 Default, CJK, FullLeft, FullRight, Boundary,
165 HalfLeft, HalfRight, NormalSpace
166 }
167 \clist_map_inline:Nn \g__xeCJK_base_class_clist

```

168 { \clist_gclear_new:c { g__xeCJK_#1_range_clist } }
169 \clist_new:N \g__xeCJK_CJK_class_clist
170 \prop_new:N \g__xeCJK_CJK_class_prop
171 \cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
172 {
173   \clist_gput_right:Nx \g__xeCJK_CJK_class_clist {#1}
174   \prop_gput:Nvx \g__xeCJK_CJK_class_prop { c__xeCJK_#1_class_int } {#1}
175 }
176 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
177 \clist_map_function:nN { CJK, FullLeft, FullRight } \__xeCJK_save_CJK_class:n
(End definition for \g__xeCJK_base_class_clist, \g__xeCJK_CJK_class_clist, and \g__xeCJK_CJK_class_prop)

```

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

```

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

```

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

```

\__xeCJK_set_char_class_aux:Nn
179 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
180 {
181   \__xeCJK_declare_char_class:nn {#2} {#3}
182   \IfBooleanT {#1} { \xeCJKResetPunctClass }
183 }
184 \cs_new_protected_nopar:Npn \__xeCJK_declare_char_class:nn #1#2
185 {
186   \clist_gput_right:cx { g__xeCJK_#1_range_clist } {#2}
187   \clist_map_inline:xn {#2}
188   {
189     \str_if_eq_x:nnF {##1} { -> }
190     {
191       \__xeCJK_set_char_class_aux:Nn \__xeCJK_set_char_class:nnn {##1}
192       { \__xeCJK_class_num:n {#1} }
193     }
194   }
195 }
196 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nn
197 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
198 \cs_generate_variant:Nn \clist_map_inline:nn { x }
(End definition for \xeCJKDeclareCharClass This function is documented on page 10.)

```

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

```

199 \NewDocumentCommand \xeCJKsetcharclass { s m m m }
200 {
201   \__xeCJK_set_char_class:nnn {#2} {#3} {#4}
202   \IfBooleanF {#1} { \xeCJKResetPunctClass }
203 }
204 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class:nnn #1#2#3
205 {
206   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
207   \int_set:Nn \l__xeCJK_tmppc_int {#3}
208   \prop_if_in:NVTF \g__xeCJK_CJK_class_prop \l__xeCJK_tmppc_int
209   { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \char_set_catcode_other:n }

```



```

210     { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \use_none:n }
211 \__xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
212 {
213     \__xeCJK_set_char_catcode:n { \l__xeCJK_tmpa_int }
214     \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
215     \int_incr:N \l__xeCJK_tmpa_int
216 }
217 }

```

(End definition for \xeCJKsetcharclass)

```

\__xeCJK_int_until_do:nn
\__xeCJK_int_until_do_aux:wn 218 \cs_new_protected:Npn \__xeCJK_int_until_do:nn #1#2
219 {
220     \__xeCJK_int_until_do_aux:wn \use_none:n
221     { \reverse_if:N \if_int_compare:w #1#2 }
222 }
223 \cs_new_protected:Npn \__xeCJK_int_until_do_aux:wn \use_none:n #1
224 {
225     #1 \exp_after:wN \__xeCJK_int_until_do_aux:wn \fi:
226     \use_none:n {#1}
227 }

```

(End definition for __xeCJK_int_until_do:nn This function is documented on page ??.)

```

\_xeCJK_check_num_range:nnNN
228 \cs_new_protected_nopar:Npn \_xeCJK_check_num_range:nnNN #1#2#3#4
229 {
230     \bool_if:nTF { \__xeCJK_if_blank_x_p:n {#1} || \__xeCJK_if_blank_x_p:n {#2} }
231     {
232         \int_set:Nn #3 { \__xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
233         \int_set_eq:NN #3 #4
234     }
235     {
236         \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
237         \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
238     }
239 }

```

(End definition for _xeCJK_check_num_range:nnNN)

\xeCJKResetPunctClass 用于重置标点符号所属的字符类。

```

240 \NewDocumentCommand \xeCJKResetPunctClass { }
241 {
242     \__xeCJK_declare_char_class:nn { HalfLeft }
243     { "28 , "2D , "5B , "60 , "7B }
244     \__xeCJK_declare_char_class:nn { HalfRight }
245     { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , }
246     \__xeCJK_declare_char_class:nn { FullLeft }
247     {
248         "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
249         "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
250         "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
251         "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6 ,
252     }

```

```

253     \__xeCJK_declare_char_class:nn { FullRight }
254     {
255         "00B7 , "2019 , "201D , "2014 , "2025 , "2026 , "2030 , "2500 , "3001 ,
256         "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 , "3015 ,
257         "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 , "3047 ,
258         "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C , "309D ,
259         "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
260         "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE , "FE50 ,
261         "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C , "FE5E ,
262         "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B , "FF1F ,
263         "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 , "FF69 ,
264         "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E , "FF9F ,
265     }
266 }

```

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

设置 CJK 文字的范围。

```

267 \__xeCJK_declare_char_class:nn { CJK }
268 {
    • Hangul Jamo (谚文字母)
269     "1100 -> "11FF ,

    • CJK Radicals Supplement (中日韩部首补充)
270     "2E80 -> "2EFF ,

    • Kangxi Radicals (康熙字典部首)
271     "2F00 -> "2FDF ,

    • Ideographic Description Characters (表意文字描述符)
272     "2FF0 -> "2FFF ,

    • CJK Symbols and Punctuation (中日韩符号和标点)
273     "3000 -> "303F ,

    • Hiragana (日文平假名)
274     "3040 -> "309F ,

    • Katakana (日文片假名)
275     "30A0 -> "30FF ,

    • Bopomofo (注音字母)
276     "3100 -> "312F ,

    • Hangul Compatibility Jamo (谚文兼容字母)
277     "3130 -> "318F ,

    • Kanbun (象形字注释标志)
278     "3190 -> "319F ,

```

- Bopomofo Extended (注音字母扩展)
279 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)
280 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)
281 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
282 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
283 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
284 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
285 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
286 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
287 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
288 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
289 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
290 "AC00 -> "D7AF ,
- Kana Supplement (日文假名增补)
291 "B000 -> "B0FF ,
- Hangul Jamo Extended-B (谚文扩展 B)
292 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
293 "F900 -> "FAFF ,
- CJK Compatibility Forms (中日韩兼容形式)
294 "FE30 -> "FE4F ,

- Halfwidth and Fullwidth Forms (半角及全角形式)

295 "FF00 -> "FFE0 ,

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

296 "20000 -> "2A6DF ,

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

297 "2A700 -> "2B73F ,

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

298 "2B740 -> "2B81F ,

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

299 "2F800 -> "2FA1F ,

300 }

重置标点符号的字符类。

301 \xeCJKResetPunctClass

\normalspacedchars 声明前后不加间距的字符。

302 \NewDocumentCommand \normalspacedchars { m }

303 {

\tl_map_inline:nn {#1}

304 { \XeTeXcharclass `##1 = __xeCJK_class_num:n { NormalSpace } }

305 }

306 \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 308 \cs_new_protected_nopar:Npn __xeCJK_inter_class_toks:nnn #1#2#3

_xeCJK_inter_class_toks:nmx 309 { \XeTeXinterchartoks __xeCJK_class_num:n {#1} __xeCJK_class_num:n {#2} = {#3} }

310 \cs_generate_variant:Nn __xeCJK_inter_class_toks:nnn { nnc , nmx }

(End definition for __xeCJK_inter_class_toks:nnn, __xeCJK_inter_class_toks:nnc, and __xeCJK_inter_class_toks:nmx)

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

311 \cs_new_nopar:Npn __xeCJK_get_inter_class_toks:nn #1#2

312 { \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。

313 \cs_new_protected_nopar:Npn __xeCJK_clear_inter_class_toks:nn #1#2

314 { __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。

315 \cs_new_protected_nopar:Npn __xeCJK_pre_inter_class_toks:nnn #1#2#3

316 {

__xeCJK_inter_class_toks:nmx {#1} {#2}

317 { \exp_not:n {#3} __xeCJK_get_inter_class_toks:nn {#1} {#2} }

318 }

319 }

(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
320 \cs_new_protected_nopar:Npn \_xeCJK_app_inter_class_toks:nnn #1#2#3
321 {
322   \_xeCJK_inter_class_toks:nxx {#1} {#2}
323   { \_xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
324 }
325 \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。

```
326 \cs_new_protected_nopar:Npn \_xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
327 {
328   \_xeCJK_inter_class_toks:nxx {#1} {#2}
329   { \_xeCJK_get_inter_class_toks:nn {#3} {#4} }
330 }
(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。

```
331 \cs_new_protected_nopar:Npn \_xeCJK_clear_CJK_toks:
332 {
333   \clist_map_inline:Nn \g__xeCJK_CJK_class_clist
334   { \_xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
335 }
(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	✓	✓	✓	✓	✓		✓	

```
336 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
337 {
338   \clist_map_inline:nn { CJK, FullLeft, FullRight }
339   {
340     \_xeCJK_inter_class_toks:nnn {#1} {##1}
341     {
342       \c_group_begin_token
343       \xeCJK_select_font:
344       \_xeCJK_clear_inter_class_toks:nn {#1} {##1}
345       \_xeCJK_clear_CJK_toks:
346     }
347     \_xeCJK_inter_class_toks:nnn {##1} {#1} { \c_group_end_token }
```

```

348     }
349     \__xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
350     \clist_map_inline:nn { FullLeft, FullRight }
351     { \__xeCJK_app_inter_class_toks:nnc {#1} {##1} { __xeCJK_CJK_and_##1:N } }
352 }
353 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
354 {
355     \__xeCJK_inter_class_toks:nnn { Boundary } {#1}
356     {
357         \bool_if:nTF
358         {
359             \l__xeCJK_xecglue_bool &&
360             \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
361             \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_space_glue_tl
362         }
363         {
364             \tex_unskip:D
365             \bool_if:nTF
366             { \__xeCJK_if_last_node_p:n { CJK } || \__xeCJK_if_last_node_p:n { CJK_space } }
367             { \CJKecglue } { \c_space_token }
368         }
369         {
370             \__xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
371             { \__xeCJK_if_last_node:nT { CJK_space } { \__xeCJK_space_or_xecglue: } }
372         }
373     }
374     \str_if_eq_x:nnF {#1} { NormalSpace }
375     { \__xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
376 }
377 \clist_map_inline:nn { Default, HalfRight, NormalSpace }
378 {
379     \__xeCJK_inter_class_toks:nnn {#1} { Boundary }
380     {
381         \peek_catcode:NTF \c_space_token
382         { { \__xeCJK_make_node:n { default_space } } }
383         { { \__xeCJK_make_node:n { default } } }
384     }
385     \str_if_eq_x:nnF {#1} { NormalSpace }
386     { \__xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
387 }
388 \__xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \__xeCJK_Boundary_and_CJK: }

\__xeCJK_Boundary_and_CJK:
\__xeCJK_check_for_glue:
389 \cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_CJK:
390 {
391     { \__xeCJK_check_for_glue: }
392     \c_group_begin_token
393     \__xeCJK_clear_CJK_toks:
394     \xeCJK_select_font:
395     \CJKsymbol
396 }

```

```

397 \cs_new_protected_nopar:Npn \__xeCJK_check_for_glue:
398 {
399   \bool_if:nTF
400     { \__xeCJK_if_last_node_p:n { CJK } || \__xeCJK_if_last_node_p:n { CJK_space } }
401     { \CJKglue }
402     {
403       \bool_if:nTF
404         {
405           \__xeCJK_if_last_node_p:n { default } ||
406           \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
407           \__xeCJK_if_last_node_p:n { default_itcorr }
408         }
409         { \CJKecglue }
410         {
411           \bool_if:nT
412             {
413               \l__xeCJK_xecglue_bool &&
414               \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
415               ( \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_space_glue_tl ||
416                 \dim_compare_p:nNn \tex_lastskip:D = \l__xeCJK_ecglue_dim )
417             }
418             {
419               \tex_unskip:D
420               \bool_if:nTF
421                 {
422                   \__xeCJK_if_last_node_p:n { default_space } ||
423                   \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
424                   \__xeCJK_if_last_node_p:n { default } ||
425                   \__xeCJK_if_last_node_p:n { default_itcorr }
426                 }
427                 { \CJKecglue }
428                 {
429                   \bool_if:nTF
430                     {
431                       \__xeCJK_if_last_node_p:n { CJK } ||
432                       \__xeCJK_if_last_node_p:n { CJK_space }
433                     }
434                     { \bool_if:NTF \l__xeCJK_space_bool \c_space_token \CJKglue }
435                     { \c_space_token }
436                 }
437             }
438           }
439         }
440       }
441   \tl_set:Nn \l__xeCJK_space_glue_tl
442     {
443       \tex_fontdimen:D \c_two \tex_font:D
444       \@plus \tex_fontdimen:D \c_three \tex_font:D
445       \@minus \tex_fontdimen:D \c_four \tex_font:D
446     }
447   (End definition for \__xeCJK_Boundary_and_CJK: and \__xeCJK_check_for_glue:)
448   \__xeCJK_inter_class_toks:nnn { CJK } { Boundary }

```

```

448 {
449   \c_group_end_token
450   { \__xeCJK_make_node:n { CJK } }
451   \__xeCJK_ignorespaces:
452 }
453 \__xeCJK_inter_class_toks:nnn { CJK } { CJK } { \__xeCJK_CJK_and_CJK:N }

\__xeCJK_CJK_and_CJK:N
454 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }
(End definition for \__xeCJK_CJK_and_CJK:N)
455 \clist_map_inline:nn { FullLeft, FullRight }
456 {
457   \__xeCJK_inter_class_toks:nnn { Boundary } {#1}
458   {
459     \c_group_begin_token
460     \__xeCJK_clear_CJK_toks:
461     \xeCJK_select_font:
462     \use:c { __xeCJK_CJK_and_#1:N }
463   }
464   \__xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
465   \__xeCJK_inter_class_toks:nnn {#1} { Boundary } { \c_group_end_token \tex_ignorespaces:D }
466 }
467 \clist_map_inline:Nn \g__xeCJK_base_class_clist
468 {
469   \bool_if:nF
470   { \str_if_eq_x_p:nn {#1} { FullLeft } || \str_if_eq_x_p:nn {#1} { FullRight } }
471   {
472     \__xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \__xeCJK_no_break: }
473     \__xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \__xeCJK_after_FullRight: }
474   }
475 }
476 \clist_map_inline:nn { CJK, FullLeft, FullRight }
477 {
478   \clist_map_inline:nn { FullLeft, FullRight }
479   { \__xeCJK_inter_class_toks:nnc {#1} {##1} { __xeCJK_#1_and_##1:N } }
480 }

```

`__xeCJK_punct_rule:n` 用于抹去标点符号的左/右空白。

```

481 \cs_new_protected_nopar:Npn \__xeCJK_punct_rule:n #1
482 {
483   \tex_vrule:D
484   \@width \dim_eval:c { \l__xeCJK_punct_coord_tl/rule/#1 }
485   \@depth \c_zero_dim
486   \@height \c_zero_dim \scan_stop:
487 }
(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。

```

488 \cs_new_protected_nopar:Npn \__xeCJK_punct_glue:n #1
489 {
490   \skip_horizontal:n

```



```

491     {
492       \dim_eval:c { \l__xeCJK_punct_coortl/glue/#1 }
493       \@plus \dim_eval:n { ( \tl_use:c { \l__xeCJK_punct_coortl/glue/#1 } ) /4 }
494       \@minus \dim_eval:n { ( \tl_use:c { \l__xeCJK_punct_coortl/glue/#1 } ) /2 }
495     }
496   }

```

(End definition for __xeCJK_punct_glue:n)

__xeCJK_punct_kern:n 相邻两个标点之间的间距。

```

497 \cs_new_protected_nopar:Npn \__xeCJK_punct_kern:n #1
498 { \tex_kern:D \dim_eval:c { \l__xeCJK_punct_coortl/kern/#1 } }

```

(End definition for __xeCJK_punct_kern:n)

\dim_eval:c

```

499 \cs_generate_variant:Nn \dim_eval:n { c }

```

(End definition for \dim_eval:c)

__xeCJK_before_FullLeft:N

```

500 \cs_new_protected_nopar:Npn \__xeCJK_before_FullLeft:N #1
501 {
502   \tl_set:Nx \l__xeCJK_lastpunct_tl {#1}
503   \__xeCJK_punct_rule:n { 1/#1 }
504   \CJKpunctsymbol {#1}
505 }

```

(End definition for __xeCJK_before_FullLeft:N)

__xeCJK_after_FullRight:

```

506 \cs_new_protected_nopar:Npn \__xeCJK_after_FullRight:
507 {
508   \__xeCJK_get_punct_bounds:nV { r } \l__xeCJK_lastpunct_tl
509   \__xeCJK_punct_rule:n { r/\l__xeCJK_lastpunct_tl }
510   \__xeCJK_punct_glue:n { r/\l__xeCJK_lastpunct_tl }
511 }

```

(End definition for __xeCJK_after_FullRight:)

__xeCJK_CJK_and_FullLeft:N

```

512 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft:N #1
513 {
514   \__xeCJK_get_punct_bounds:nN { 1 } {#1}
515   \int_compare:nNf \etex_lastnodetype:D = \c_one
516     { \__xeCJK_punct_glue:n { 1/#1 } }
517   \__xeCJK_before_FullLeft:N {#1}
518   \__xeCJK_punct_if_middle:NT {#1}
519   {
520     \__xeCJK_punct_rule:n { m/#1 }
521     \__xeCJK_punct_glue:n { 1/#1 }
522   }
523 }

```

(End definition for __xeCJK_CJK_and_FullLeft:N)

_xeCJK_CJK_and_FullRight:N

```
524 \cs_new_protected_nopar:Npn \_xeCJK_CJK_and_FullRight:N #1
525 {
526   \_xeCJK_get_punct_bounds:nN { r } {#1}
527   \_xeCJK_punct_if_long:NTF {#1} \CJKglue
528   {
529     \_xeCJK_no_break:
530     \_xeCJK_punct_if_middle:NT {#1}
531     {
532       \_xeCJK_punct_glue:n { r/#1 }
533       \_xeCJK_punct_rule:n { m/#1 }
534     }
535   }
536   \tl_gset:Nx \l__xeCJK_lastpunct_tl {#1}
537   \CJKpunctsymbol {#1}
538 }
```

(End definition for _xeCJK_CJK_and_FullRight:N)

_xeCJK_FullLeft_and_FullLeft:N

```
539 \cs_new_protected_nopar:Npn \_xeCJK_FullLeft_and_FullLeft:N #1
540 {
541   \_xeCJK_no_break:
542   \_xeCJK_get_punct_bounds:nN { l } {#1}
543   \_xeCJK_get_kern:VN \l__xeCJK_lastpunct_tl {#1}
544   \_xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
545   \_xeCJK_before_FullLeft:N {#1}
546 }
```

(End definition for _xeCJK_FullLeft_and_FullLeft:N)

_xeCJK_FullLeft_and_FullRight:N

```
547 \cs_new_protected_nopar:Npn \_xeCJK_FullLeft_and_FullRight:N #1
548 {
549   \_xeCJK_no_break:
550   \_xeCJK_get_punct_bounds:nN { r } {#1}
551   \_xeCJK_get_kern:VN \l__xeCJK_lastpunct_tl {#1}
552   \_xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
553   \_xeCJK_no_break:
554   \tl_set:Nx \l__xeCJK_lastpunct_tl {#1}
555   \CJKpunctsymbol {#1}
556 }
```

(End definition for _xeCJK_FullLeft_and_FullRight:N)

_xeCJK_FullRight_and_FullLeft:N

```
557 \cs_new_protected_nopar:Npn \_xeCJK_FullRight_and_FullLeft:N #1
558 {
559   \_xeCJK_punct_rule:n { r/\l__xeCJK_lastpunct_tl }
560   \_xeCJK_get_punct_bounds:nN { l } {#1}
561   \_xeCJK_get_kern:VN \l__xeCJK_lastpunct_tl {#1}
562   \_xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
563   \_xeCJK_punct_nobreak:
564   \_xeCJK_before_FullLeft:N {#1}
565 }
```

(End definition for __xeCJK_FullRight_and_FullLeft:N)

__xeCJK_FullRight_and_FullRight:N

```
566 \cs_new_protected_nopar:Npn \__xeCJK_FullRight_and_FullRight:N #1
567 {
568   \__xeCJK_punct_rule:n { r/\l__xeCJK_lastpunct_tl }
569   \__xeCJK_get_punct_bounds:nN { r } {#1}
570   \__xeCJK_get_kern:VN \l__xeCJK_lastpunct_tl {#1}
571   \__xeCJK_punct_kern:n { \l__xeCJK_lastpunct_tl - #1 }
572   \__xeCJK_no_break:
573   \tl_set:Nx \l__xeCJK_lastpunct_tl {#1}
574   \CJKpunctsymbol {#1}
575 }
```

(End definition for __xeCJK_FullRight_and_FullRight:N)

__xeCJK_if_last_node_p:n

__xeCJK_if_last_node:nTF

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

(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

```
581 \cs_new_protected_nopar:Npn \__xeCJK_def_node:nn #1#2
582 {
583   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
584   { \dim_gset:cn } { \dim_const:cn }
585   { c__xeCJK_#1_node_dim } {#2}
586 }
587 \cs_new_protected_nopar:Npn \__xeCJK_make_node:n #1
588 {
589   \tex_kern:D - \dim_use:c { c__xeCJK_#1_node_dim }
590   \tex_kern:D \dim_use:c { c__xeCJK_#1_node_dim }
591 }
592 \__xeCJK_def_node:nn { CJK } { 11 sp }
593 \__xeCJK_def_node:nn { CJK_space } { 12 sp }
594 \__xeCJK_def_node:nn { default } { 13 sp }
595 \__xeCJK_def_node:nn { default_space } { 14 sp }
596 \__xeCJK_def_node:nn { default_itcorr } { 15 sp }
```

(End definition for __xeCJK_def_node:nn and __xeCJK_make_node:n)

CJKglue CJK 文字之间插入的 glue。

```
597 \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

```
598 \keys_define:nn { xeCJK / options }
599 {
600   CJKecglue .code:n =
601   {
602     \cs_set:Npn \CJKecglue {#1}
```

```

603     \settowidth \l__xeCJK_ecglue_dim \CJKecglue
604   },
605   xCJKecglue .choice:,
606   xCJKecglue / true .code:n =
607   {
608     \bool_set_true:N \l__xeCJK_xecglue_bool
609     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
610   },
611   xCJKecglue / false .code:n =
612   {
613     \bool_set_false:N \l__xeCJK_xecglue_bool
614     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \c_space_token
615   },
616   xCJKecglue / unknown .code:n =
617   {
618     \bool_set_true:N \l__xeCJK_xecglue_bool
619     \cs_set:Npn \CJKecglue {#1}
620     \settowidth \l__xeCJK_ecglue_dim \CJKecglue
621     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
622   },
623   xCJKecglue .default:n = { true } ,
624 }
625 \dim_new:N \l__xeCJK_ecglue_dim
626 \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。

```

627 \keys_define:nn { xeCJK / options }
628 {
629   CJKspace .choice:,
630   CJKspace / true .code:n = { \CJKspace },
631   CJKspace / false .code:n = { \CJKnospace },
632   CJKspace .default:n = { true },
633   space .meta:n = { CJKspace = true },
634   nospace .meta:n = { CJKspace = false },
635 }

```

(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
636 \NewDocumentCommand \CJKspace { }
637 {
638   \bool_set_true:N \l__xeCJK_space_bool
639   \cs_set_eq:NN \__xeCJK_ignorespaces: \__xeCJK_peek_math:
640 }
641 \NewDocumentCommand \CJKnospace { }
642 {
643   \bool_set_false:N \l__xeCJK_space_bool
644   \cs_set_eq:NN \__xeCJK_ignorespaces: \__xeCJK_ignore_spaces:
645 }

```

(End definition for \CJKspace and \CJKnospace)

```

\__xeCJK_ignore_spaces:
  \__xeCJK_peek_math:
646 \cs_new_protected_nopar:Npn \__xeCJK_ignore_spaces:
647 {
648   \peek_catcode:NTF \c_space_token
649   {
650     \__xeCJK_if_last_node:nT { CJK }
651     { \tex_unkern:D \tex_unkern:D { \__xeCJK_make_node:n { CJK_space } } }
652     \__xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
653     { \__xeCJK_space_or_xecglue: }
654     {
655       \bool_if:nT
656       {
657         \__xeCJK_if_outer_macro_p:c { l_peek_token } ||
658         \token_if_cs_p:c { l_peek_token }
659       }
660       { \__xeCJK_space_or_xecglue: }
661     }
662   }
663   {
664     \__xeCJK_if_outer_macro:cF { l_peek_token }
665     {
666       \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
667       { \cs_if_exist:cF { l_peek_token } \exp_not:N }
668     }
669   }
670 }
671 \cs_new_protected_nopar:Npn \__xeCJK_peek_math:
672 {
673   \peek_catcode:NTF \c_space_token
674   {
675     \__xeCJK_if_last_node:nT { CJK }
676     { \tex_unkern:D \tex_unkern:D { \__xeCJK_make_node:n { CJK_space } } }
677     \__xeCJK_peek_catcode_ignore_spaces:NF \c_space_token
678     { \__xeCJK_space_or_xecglue: }
679   }
680   {
681     \__xeCJK_if_outer_macro:cF { l_peek_token }
682     {
683       \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
684       { \cs_if_exist:cF { l_peek_token } \exp_not:N }
685     }
686   }
687 }
(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。

```

688 \keys_define:nn { xeCJK / options }
689 {
690   CheckSingle .choice:,
691   CheckSingle / true .code:n = { \xeCJKenablechecksingle },

```

```

692     CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
693     CheckSingle      .default:n = { true },
694     CJKchecksingle    .meta:n = { CheckSingle = true } ,
695 }

```

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

\xeCJKenablechecksingle 孤字处理功能开关。

```

\xeCJKdisablechecksingle
696 \bool_new:N \l__xeCJK_checksingl_bool
697 \NewDocumentCommand \xeCJKenablechecksingle { }
698 {
699     \bool_if:NF \l__xeCJK_checksingl_bool
700     {
701         \bool_set_true:N \l__xeCJK_checksingl_bool
702         \cs_set_eq:NN \__xeCJK_checksingl_save_CJKsymbol:N \__xeCJK_CJK_and_CJK:N
703         \cs_set_eq:NN \__xeCJK_CJK_and_CJK:N \__xeCJK_checksingl:N
704     }
705 }
706 \NewDocumentCommand \xeCJKdisablechecksingle { }
707 {
708     \bool_if:NT \l__xeCJK_checksingl_bool
709     {
710         \bool_set_false:N \l__xeCJK_checksingl_bool
711         \cs_set_eq:NN \__xeCJK_CJK_and_CJK:N \__xeCJK_checksingl_save_CJKsymbol:N
712     }
713 }

```

(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle)

```

\__xeCJK_checksingl:N
\__xeCJK_checksingl:NN
\__xeCJK_checksingl:NNN
714 \cs_new_protected_nopar:Npn \__xeCJK_checksingl:N #1
715 {
716     \peek_catcode:NTF \c_catcode_other_token
717     { \__xeCJK_checksingl:NN {#1} }
718     {
719         \bool_if:nTF
720         {
721             ! ( \__xeCJK_if_outer_macro_p:c { l_peek_token } )      &&
722             \__xeCJK_if_blank_x_p:n { \token_get_arg_spec:c { l_peek_token } } &&
723             \exp_args:NNc \exp_args:No \tl_if_single_token_p:n { l_peek_token } &&
724             \exp_args:NNc \exp_after:wN \token_if_other_p:N { l_peek_token }
725         }
726         { \__xeCJK_checksingl:NN {#1} }
727         { \__xeCJK_checksingl_save_CJKsymbol:N {#1} }
728     }
729 }
730 \cs_new_protected_nopar:Npn \__xeCJK_checksingl:NN #1#2
731 {
732     \peek_catcode:NTF \c_catcode_other_token
733     { \__xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
734     {
735         \bool_if:nTF
736         {
737             ! ( \__xeCJK_if_outer_macro_p:c { l_peek_token } ) &&

```

```

738      ( \token_if_space_p:c { l_peek_token } ||
739      \token_if_math_toggle_p:c { l_peek_token } ||
740      \token_if_cs_p:c { l_peek_token } )
741    }
742    {
743      \bool_if:nTF { \token_if_space_p:c { l_peek_token } }
744      {
745        \__xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_other_token
746        { \__xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
747        {
748          \bool_if:nTF
749          {
750            ! ( \__xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
751            ( \token_if_math_toggle_p:c { l_peek_token } ||
752            \token_if_cs_p:c { l_peek_token } )
753          }
754          { \__xeCJK_checksingl:N {#1} {#2} \c_space_token }
755          { \__xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
756        }
757      }
758      { \__xeCJK_checksingl:N {#1} {#2} { } }
759    }
760    { \__xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
761  }
762 }
763 \cs_new_protected_nopar:Npn \__xeCJK_checksingl:N {#1} {#2} {#3}
764 {
765   \bool_if:nTF
766   {
767     \token_if_eq_meaning_p:NN \l_peek_token \par ||
768     \token_if_eq_meaning_p:NN \l_peek_token \[ % \]
769   }
770   { \CJKsymbol {#1} \__xeCJK_no_break: #2 }
771   { \__xeCJK_checksingl_env:NNN {#1} {#2} {#3} }
772 }

```

(End definition for __xeCJK_checksingl:N, __xeCJK_checksingl:NN, and __xeCJK_checksingl:NNN)

__xeCJK_checksingl_env:NNN

```

\__xeCJK_checksingl_env:NNNN
773 \cs_new_protected_nopar:Npn \__xeCJK_checksingl_env:NNN {#1} {#2} {#3}
774 {
775   \peek_catcode_remove:NTF \c_math_toggle_token
776   {
777     \peek_catcode:NTF \c_math_toggle_token
778     { \CJKsymbol {#1} \__xeCJK_no_break: #2 \c_math_toggle_token }
779     { \__xeCJK_checksingl_save_CJKsymbol:N {#1} #2 #3 \c_math_toggle_token }
780   }
781   {
782     \peek_meaning_remove:NTF \begin
783     { \__xeCJK_checksingl_env:NNNN {#1} {#2} {#3} \begin }
784     {
785       \peek_meaning_remove:NTF \end
786       { \__xeCJK_checksingl_env:NNNN {#1} {#2} {#3} \end }

```

```

787         { \__xeCJK_checksingle_save_CJKsymbol:N {#1} #2#3 }
788     }
789 }
790 }
791 \cs_new_protected_nopar:Npn \__xeCJK_checksingle_env:NNNNn #1#2#3#4#5
792 {
793     \clist_if_in:NnTF \g__xeCJK_inline_env_clist {#5}
794     { \__xeCJK_checksingle_save_CJKsymbol:N {#1} #2#3 }
795     { \CJKsymbol {#1} \__xeCJK_no_break: #2 }
796     \scan_stop: #4 {#5}
797 }
(End definition for \__xeCJK_checksingle_env:NNN and \__xeCJK_checksingle_env:NNNNn)

```

InlineEnv

InlineEnv+ 798 \keys_define:nn { xeCJK / options }

InlineEnv- 799 {

```

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

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

```

812 \cs_generate_variant:Nn \token_if_cs_p:N { c }
813 \cs_generate_variant:Nn \token_if_space_p:N { c }
814 \cs_generate_variant:Nn \token_get_arg_spec:N { c }
815 \cs_generate_variant:Nn \token_if_math_toggle_p:N { c }
816 \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 817 \tl_new:N \__xeCJK_UL_subclass_patch_tl
818 \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: 819 \cs_new_protected_nopar:Npn \__xeCJK_erase_CJKsymbol:
820 {
821     \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
822     \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
823 }
824 \cs_new_protected_nopar:Npn \__xeCJK_restore_CJKsymbol:
825 {
826     \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block

```



```

827     \CJKsymbol
828 }
(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。

```

829 \NewDocumentCommand \xeCJKDeclareSubCJKBlock { s m m }
830 {
831     \xeCJKDeclareSubCharClass { CJK } {#2} {#3}
832     \IfBooleanT {#1} { \xeCJKResetPunctClass }
833 }
834 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock This function is documented on page 9.)

```

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

```

835 \bool_new:N \l__xeCJK_sub_cancel_bool
836 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
837 {
838     \bool_if:NF \l__xeCJK_sub_cancel_bool
839     {
840         \bool_set_true:N \l__xeCJK_sub_cancel_bool
841         \__xeCJK_sub_restore_or_cancel:n {#2}
842         \IfBooleanT {#1} { \xeCJKResetPunctClass }
843     }
844 }
845 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
846 {
847     \bool_if:NT \l__xeCJK_sub_cancel_bool
848     {
849         \bool_set_false:N \l__xeCJK_sub_cancel_bool
850         \__xeCJK_sub_restore_or_cancel:n {#2}
851         \IfBooleanT {#1} { \xeCJKResetPunctClass }
852     }
853 }
854 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
855 {
856     \clist_map_inline:xn {#1}
857     {
858         \int_if_exist:CTF { c__xeCJK_ CJK/##1 _class_int }
859         {
860             \__xeCJK_declare_char_class:nn
861             { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
862             { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
863         }
864         { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
865     }
866 }
867 \__xeCJK_msg_new:nn { SubBlock-undefined }
868 {
869     The~CJK~sub~block~'#1'~is~undefined.\\
870     Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
871     to~declare~it.
872 }

```

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

\xeCJKDeclareSubCharClass

```

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

```

__xeCJK_set_sub_class_toks:nn

```

886 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
887 {
888   \clist_map_inline:Nn \g__xeCJK_base_class_clist
889   {
890     \__xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
891     \__xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
892     \str_if_eq_x:nnTF {##1} { CJK }
893     {
894       \__xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
895       { \__xeCJK_switch_font:nn {#1} {#2} }
896     }
897     {
898       \__xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
899       { \__xeCJK_erase_CJKsymbol: }
900       \__xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
901       { \__xeCJK_switch_font:nn {#1} {#2} \__xeCJK_restore_CJKsymbol: }
902     }
903   }
904   \__xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
905   \clist_if_empty:NF \g__xeCJK_CJK_subclass_clist
906   {
907     \clist_map_inline:Nn \g__xeCJK_CJK_subclass_clist
908     {
909       \__xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
910       \__xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
911       \__xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
912       { \__xeCJK_switch_font:nn {#2} {##1} }
913       \__xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
914       { \__xeCJK_switch_font:nn {##1} {#2} }
915     }
916   }
917   \clist_gput_right:Nn \g__xeCJK_CJK_subclass_clist {#2}
918   \__xeCJK_save_CJK_class:n { #1/#2 }

```

```

919 \clist_map_inline:nn { CJK, FullLeft, FullRight }
920 {
921   \__xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
922   { \__xeCJK_switch_font:nn {#2} {#1} }
923 }
924 }

```

(End definition for __xeCJK_set_sub_class_toks:nn)

_xeCJK_UL_subclass_patch:nn

```

925 \cs_new_protected_nopar:Npn \__xeCJK_UL_subclass_patch:nn #1#2
926 {
927   \tl_put_right:Nn \__xeCJK_UL_subclass_patch_tl
928   {
929     \clist_map_inline:nn { Default, HalfLeft, HalfRight }
930     {
931       \str_if_eq_x:nnTF {##1} { HalfLeft }
932       { \__xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKe glue \CJKsymbol } }
933       { \__xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
934       \__xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
935       {
936         \__xeCJK_erase_CJKsymbol:
937         \__xeCJK_Boundary_and_CJK:
938         \__xeCJK_switch_font:nn {#1} {#2}
939         \__xeCJK_restore_CJKsymbol:
940       }
941     }
942   }
943 }

```

(End definition for __xeCJK_UL_subclass_patch:nn)

5.8 标点处理

\g__xeCJK_punctstyle_clist 标点处理方式。

```

944 \clist_new:N \g__xeCJK_punctstyle_clist
945 \clist_set:Nn \g__xeCJK_punctstyle_clist
946 { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
947 \clist_map_inline:Nn \g__xeCJK_punctstyle_clist
948 { \tl_const:cn { c__xeCJK_ps_#1_tl } {#1} }

```

(End definition for \g__xeCJK_punctstyle_clist)

\keys_define:nx

```

949 \cs_generate_variant:Nn \keys_define:nn { nx }

```

(End definition for \keys_define:nx)

PunctStyle

```

950 \keys_define:nn { xeCJK / options }
951 {
952   PunctStyle .choice_code:n =
953   {
954     \tl_set:Nx \l__xeCJK_punctstyle_tl { \l_keys_choice_tl }
955     \tl_if_eq:NNT \l__xeCJK_punctstyle_tl \c__xeCJK_ps_plain_tl
956     { \xeCJKallowbreakbetweenpuncts }

```

```

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

```

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

AllowBreakBetweenPuncts 相关选项声明。

```

KaiMingPunct 976 \keys_define:nn { xeCJK / options }
LongPunct    977 {
MiddlePunct  978   AllowBreakBetweenPuncts .choice:,
PunctWidth   979   AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
              980   AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
              981   AllowBreakBetweenPuncts .default:n = { true },
              982   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixedwidth } {#1} },
              983   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixedwidth } {#1} },
              984   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
              985   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} },
              986   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} },
              987   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} },
              988   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} },
              989   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} },
              990   MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} },
              991   PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
              992 }

```

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

```

\xeCJKallowbreakbetweenpuncts
\xeCJKnobreakbetweenpuncts 993 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
                              994 { \cs_set_eq:NN \__xeCJK_punct_nobreak: \__xeCJK_zero_glue: }
                              995 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
                              996 { \cs_set_eq:NN \__xeCJK_punct_nobreak: \__xeCJK_no_break: }
                              997 \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 998 \clist_set:Nn \g__xeCJK_special_ps_clist { mixedwidth , long , middle }
  \_xeCJK_add_special_punct:nn 999 \clist_map_inline:Nn \g__xeCJK_special_ps_clist
  \_xeCJK_sub_special_punct:nn

```

```

1000 {
1001   \tl_new:c { l__xeCJK_#1_punct_tl }
1002   \prop_new:c { l__xeCJK_#1_punct_prop }
1003 }
1004 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1005 {
1006   \tl_set:cx { l__xeCJK_#1_punct_tl } {#2}
1007   \prop_clear:c { l__xeCJK_#1_punct_prop }
1008   \tl_map_inline:cn { l__xeCJK_#1_punct_tl }
1009     { \prop_put:cnn { l__xeCJK_#1_punct_prop } {##1} { } }
1010 }
1011 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1012 {
1013   \tl_put_right:cx { l__xeCJK_#1_punct_tl } {#2}
1014   \tl_map_inline:cn { l__xeCJK_#1_punct_tl }
1015     { \prop_put:cnn { l__xeCJK_#1_punct_prop } {##1} { } }
1016 }
1017 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1018 {
1019   \tl_map_inline:xn {#2}
1020   {
1021     \tl_remove_all:cn { l__xeCJK_#1_punct_tl } {##1}
1022     \prop_remove:cn { l__xeCJK_#1_punct_prop } {##1}
1023   }
1024 }
1025 \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 1026 \cs_generate_variant:Nn \dim_set:Nn { Nc }

\dim_min:cc 1027 \cs_generate_variant:Nn \dim_add:Nn { Nc }

1028 \cs_generate_variant:Nn \dim_min:nn { cc }

(End definition for \dim_set:Nc, \dim_add:Nc, and \dim_min:cc)

__xeCJK_dim_set_max:Nn

__xeCJK_dim_set_min:Nn 1029 \cs_new_protected_nopar:Npn __xeCJK_dim_set_max:Nn #1#2

1030 { \dim_set:Nn #1 { \dim_max:nn {#1} {#2} } }

1031 \cs_new_protected_nopar:Npn __xeCJK_dim_set_min:Nn #1#2

1032 { \dim_set:Nn #1 { \dim_min:nn {#1} {#2} } }

(End definition for __xeCJK_dim_set_max:Nn and __xeCJK_dim_set_min:Nn)

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

1033 \tl_set:Nn \l__xeCJK_punct_coor_tl { \l__xeCJK_font_coor_tl/\l__xeCJK_punctstyle_tl }

(End definition for \l__xeCJK_punct_coor_tl)

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

1034 \cs_new_protected_nopar:Npn __xeCJK_get_punct_bounds:nN #1#2

1035 {

1036 \tl_if_exist:cF { \l__xeCJK_punct_coor_tl/rule/#1/#2 }

1037 {

```

1038 \tl_if_eq:NNTF \l__xeCJK_punctstyle_tl \c__xeCJK_ps_plain_tl
1039 {
1040   \clist_map_inline:nn { glue , rule , bound }
1041     { \tl_gset:cV { \l__xeCJK_punct_coors_tl/##1/##1/##2 } \c_zero_dim }
1042   \tl_gset:cV { \l__xeCJK_punct_coors_tl/rule/m/##2 } \c_zero_dim
1043 }
1044 {
1045   \tl_if_exist:cF { \l__xeCJK_punct_coors_tl/bound/##1/##2 }
1046     { { \xeCJK_select_font: \__xeCJK_get_punct_dimen:N {##2} } }
1047   \__xeCJK_punct_if_long:NNTF {##2}
1048   {
1049     \dim_zero:N \l__xeCJK_tmpa_dim
1050     \dim_zero:N \l__xeCJK_tmpb_dim
1051     \dim_zero:N \l__xeCJK_tmppc_dim
1052   }
1053   {
1054     \dim_set:Nc \l__xeCJK_tmpa_dim { \l__xeCJK_punct_coors_tl/bound/##1/##2 }
1055     \dim_set:Nc \l__xeCJK_tmpb_dim
1056       { \l__xeCJK_punct_coors_tl/bound/\str_if_eq_x:nNTF {##1} lrl/##2 }
1057     \dim_set:Nn \l__xeCJK_tmppc_dim
1058       {
1059         \cs_if_exist_use:cF { g__xeCJK_punct_width/##2 }
1060         {
1061           \__xeCJK_if_blank_x:nTF \g__xeCJK_punct_width_tl
1062             { \c_zero_dim }
1063             { \g__xeCJK_punct_width_tl }
1064         }
1065       }
1066     \__xeCJK_punct_if_middle:NNTF {##2}
1067     {
1068       \dim_set:Nn \l__xeCJK_tmppc_dim
1069       {
1070         \dim_compare:nNNTF \l__xeCJK_tmppc_dim > \c_zero_dim
1071         { \l__xeCJK_tmppc_dim }
1072         {
1073           \tl_case:Nnn \l__xeCJK_punctstyle_tl
1074             {
1075               \c__xeCJK_ps_halfwidth_tl { .5 em }
1076               \c__xeCJK_ps_mixedwidth_tl { .5 em }
1077               \c__xeCJK_ps_CCT_tl { .7 em }
1078               \c__xeCJK_ps_fullwidth_tl { 1 em }
1079             }
1080             { 1 em }
1081         }
1082         - \tl_use:c { \l__xeCJK_punct_coors_tl/dimen/##2 }
1083       }
1084     \dim_set:Nn \l__xeCJK_tmppc_dim { .5 \l__xeCJK_tmppc_dim }
1085   }
1086   {
1087     \dim_compare:nNNTF \l__xeCJK_tmppc_dim > \c_zero_dim
1088     {
1089       \dim_set:Nn \l__xeCJK_tmppc_dim

```

```

1090         {
1091             \l__xeCJK_tmpc_dim - \l__xeCJK_tmpb_dim
1092             - \tl_use:c { \l__xeCJK_punct_coors_tl/dimen/#2 }
1093         }
1094     }
1095     {
1096         \dim_set:Nn \l__xeCJK_tmpc_dim
1097         { \dim_min:nn \l__xeCJK_tmpa_dim \l__xeCJK_tmpb_dim }
1098         \tl_case:Nnn \l__xeCJK_punctstyle_tl
1099         {
1100             \c__xeCJK_ps_halfwidth_tl { \prg_do_nothing: }
1101             \c__xeCJK_ps_mixedwidth_tl
1102             {
1103                 \__xeCJK_punct_if_mixedwidth:NT {#2}
1104                 {
1105                     \dim_add:Nn \l__xeCJK_tmpc_dim { .5 \l__xeCJK_tmpa_dim }
1106                     \__xeCJK_dim_set_min:Nn \l__xeCJK_tmpc_dim \l__xeCJK_tmpa_dim
1107                 }
1108             }
1109             \c__xeCJK_ps_CCT_tl
1110             {
1111                 \__xeCJK_punct_if_mixedwidth:NTF {#2}
1112                 { \dim_add:Nn \l__xeCJK_tmpc_dim { .5 \l__xeCJK_tmpa_dim } }
1113                 { \dim_add:Nn \l__xeCJK_tmpc_dim { .3 \l__xeCJK_tmpa_dim } }
1114                 \__xeCJK_dim_set_min:Nn \l__xeCJK_tmpc_dim \l__xeCJK_tmpa_dim
1115             }
1116         }
1117         { \dim_set_eq:NN \l__xeCJK_tmpc_dim \l__xeCJK_tmpa_dim }
1118     }
1119 }
1120 \__xeCJK_dim_set_max:Nn \l__xeCJK_tmpc_dim \c_zero_dim
1121 }
1122 \tl_gset:cx { \l__xeCJK_punct_coors_tl/rule/#1/#2 }
1123 { - \dim_use:N \l__xeCJK_tmpa_dim }
1124 \tl_gset:cx { \l__xeCJK_punct_coors_tl/rule/m/#2 }
1125 { - \dim_use:N \l__xeCJK_tmpb_dim }
1126 \tl_gset:cV { \l__xeCJK_punct_coors_tl/glue/#1/#2 } \l__xeCJK_tmpc_dim
1127 }
1128 }
1129 }
1130 \cs_generate_variant:Nn \__xeCJK_get_punct_bounds:nN { nV }
(End definition for \__xeCJK_get_punct_bounds:nN)

```

\xeCJKsetwidth

```

1131 \NewDocumentCommand \xeCJKsetwidth { m m }
1132 { \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 1133 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
\__xeCJK_punct_if_long_p:N 1134 {
\__xeCJK_punct_if_middle_p:N 1135 \tl_if_exist:ctF { \l__xeCJK_punct_coors_tl/glue/r/#1 }
\__xeCJK_punct_if_mixedwidth p:N
\__xeCJK_punct_if_right:NTF
\__xeCJK_punct_if_both_left_right:NNTF
\__xeCJK_punct_if_long:NTF
\__xeCJK_punct_if_middle:NTF
\__xeCJK_punct_if_mixedwidth:NTF

```

```

1136     \prg_return_true: \prg_return_false:
1137 }
1138 \prg_new_conditional:Npnn \__xeCJK_punct_if_both_left_right:NN #1#2 { p , T , F , TF }
1139 {
1140     \bool_if:nTF
1141     {
1142         ( \tl_if_exist_p:c { \l__xeCJK_punct_coord_tl/glue/r/#1 } &&
1143           \tl_if_exist_p:c { \l__xeCJK_punct_coord_tl/glue/r/#2 } ) ||
1144         ( \tl_if_exist_p:c { \l__xeCJK_punct_coord_tl/glue/l/#1 } &&
1145           \tl_if_exist_p:c { \l__xeCJK_punct_coord_tl/glue/l/#2 } )
1146     }
1147     \prg_return_true: \prg_return_false:
1148 }
1149 \cs_generate_variant:Nn \prg_new_conditional:Npnn { c }
1150 \clist_map_inline:Nn \g__xeCJK_special_ps_clist
1151 {
1152     \prg_new_conditional:cpnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1153     {
1154         \prop_if_in:cnTF { l__xeCJK_#1_punct_prop } {##1}
1155         \prg_return_true: \prg_return_false:
1156     }
1157 }

```

(End definition for __xeCJK_punct_if_right:N and others.)

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

__xeCJK_calc_kern:NN

```

1158 \cs_new_protected_nopar:Npn \__xeCJK_get_kern:NN #1#2
1159 {
1160     \tl_if_exist:cF { \l__xeCJK_punct_coord_tl/kern/#1-#2 }
1161     {
1162         \dim_zero:N \l__xeCJK_tmpa_dim
1163         \tl_if_eq:NNTF \l__xeCJK_punctstyle_tl \c__xeCJK_ps_plain_tl
1164         {
1165             \tl_if_exist:cTF { g__xeCJK_punct/kern/#1-#2 }
1166             { \dim_set:Nc \l__xeCJK_tmpa_dim { g__xeCJK_punct/kern/#1-#2 } }
1167             { \__xeCJK_calc_kern:NN {#1} {#2} }
1168         }
1169         \tl_gset:cV { \l__xeCJK_punct_coord_tl/kern/#1-#2 } \l__xeCJK_tmpa_dim
1170     }
1171 }
1172 \cs_new_protected_nopar:Npn \__xeCJK_calc_kern:NN #1#2
1173 {
1174     \tl_if_eq:NNTF \l__xeCJK_punctstyle_tl \c__xeCJK_ps_marginkerning_tl
1175     {
1176         \__xeCJK_punct_if_right:NT {#1}
1177         { \dim_add:Nc \l__xeCJK_tmpa_dim { \l__xeCJK_punct_coord_tl/bound/r/#1 } }
1178         \__xeCJK_punct_if_right:NF {#2}
1179         { \dim_add:Nc \l__xeCJK_tmpa_dim { \l__xeCJK_punct_coord_tl/bound/l/#2 } }
1180     }
1181     {
1182         \tl_if_eq:NNTF \l__xeCJK_punctstyle_tl \c__xeCJK_ps_fullwidth_tl
1183         { \__xeCJK_calc_kern:NnNN \l__xeCJK_tmpa_dim { 1.5 em } {#1} {#2} }

```



```

1184 {
1185     \__xeCJK_punct_if_both_left_right:NNTF {#1} {#2}
1186     {
1187         \__xeCJK_punct_if_mixedwidth:NT {#1}
1188         {
1189             \__xeCJK_calc_kern:NnNN \l__xeCJK_tmpa_dim { 1 em } {#1} {#2}
1190             \dim_compare:nNnT \l__xeCJK_tmpa_dim < { .1 em }
1191             { \dim_set:Nc \l__xeCJK_tmpa_dim { \l__xeCJK_punct_coor_tl/bound/l/#1 } }
1192         }
1193     }
1194     {
1195         \bool_if:nTF
1196         {
1197             \__xeCJK_punct_if_mixedwidth_p:N {#1} &&
1198             ! ( \tl_if_eq_p:NN \l__xeCJK_punctstyle_tl \c__xeCJK_ps_halfwidth_tl )
1199         }
1200         {
1201             \dim_set:Nc \l__xeCJK_tmpa_dim { \l__xeCJK_punct_coor_tl/glue/r/#1 }
1202             \dim_set:Nn \l__xeCJK_tmpa_dim { .7 \l__xeCJK_tmpa_dim }
1203         }
1204         {
1205             \__xeCJK_calc_kern:NnNN \l__xeCJK_tmpa_dim { 1 em } {#1} {#2}
1206             \dim_compare:nNnT \l__xeCJK_tmpa_dim < { .1 em }
1207             {
1208                 \dim_set:Nn \l__xeCJK_tmpa_dim
1209                 {
1210                     \dim_max:nn
1211                     {
1212                         \dim_min:cc
1213                         { \l__xeCJK_punct_coor_tl/bound/l/#1 }
1214                         { \l__xeCJK_punct_coor_tl/bound/r/#1 }
1215                     }
1216                     {
1217                         \dim_min:cc
1218                         { \l__xeCJK_punct_coor_tl/bound/l/#2 }
1219                         { \l__xeCJK_punct_coor_tl/bound/r/#2 }
1220                     }
1221                 }
1222             }
1223         }
1224     }
1225 }
1226 \bool_if:nT
1227 { \__xeCJK_punct_if_long_p:N {#1} || \__xeCJK_punct_if_long_p:N {#2} }
1228 { \__xeCJK_dim_set_max:Nn \l__xeCJK_tmpa_dim { .1 em } }
1229 }
1230 \__xeCJK_dim_set_max:Nn \l__xeCJK_tmpa_dim \c_zero_dim
1231 }
1232 \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。

```
1233 \cs_new_protected_nopar:Npn \__xeCJK_calc_kern:NnNN #1#2#3#4
1234 {
1235   \dim_set:Nn #1
1236   {
1237     #2
1238     - \tl_use:c { \l__xeCJK_punct_coord_tl/
1239                 \__xeCJK_punct_if_right:NTF {#3} { bound } { glue } /l/#3 }
1240     - \tl_use:c { \l__xeCJK_punct_coord_tl/
1241                 \__xeCJK_punct_if_right:NTF {#4} { glue } { bound } /r/#4 }
1242     - \tl_use:c { \l__xeCJK_punct_coord_tl/dimen/#3 }
1243     - \tl_use:c { \l__xeCJK_punct_coord_tl/dimen/#4 }
1244     \__xeCJK_punct_if_both_left_right:NNT {#3} {#4}
1245     { + \tl_use:c { \l__xeCJK_punct_coord_tl/rule/m/#4 } }
1246   }
1247 }
(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。

```
1248 \cs_new_protected_nopar:Npn \__xeCJK_get_punct_dimen:N #1
1249 {
1250   \__xeCJK_get_glyph_bounds:nNN { left } {#1} \l__xeCJK_tmpa_dim
1251   \tl_gset:cV { \l__xeCJK_punct_coord_tl/bound/l/#1 } \l__xeCJK_tmpa_dim
1252   \dim_set_eq:Nn \l__xeCJK_tmpb_dim \l__xeCJK_tmpa_dim
1253   \__xeCJK_get_glyph_bounds:nNN { right } {#1} \l__xeCJK_tmpa_dim
1254   \tl_gset:cV { \l__xeCJK_punct_coord_tl/bound/r/#1 } \l__xeCJK_tmpa_dim
1255   \dim_add:Nn \l__xeCJK_tmpb_dim \l__xeCJK_tmpa_dim
1256   \tl_gset:cx { \l__xeCJK_punct_coord_tl/dimen/#1 }
1257   { \dim_eval:n { \tex_fontcharwd:D \tex_font:D `#1 - \l__xeCJK_tmpb_dim } }
1258   \bool_if:nT
1259   {
1260     \__xeCJK_punct_if_long_p:N {#1} &&
1261     ! ( \str_if_eq_x_p:nn {#1} { \cdots } || \str_if_eq_x_p:nn {#1} { \cdot } )
1262   }
1263   {
1264     \clist_map_inline:Nn \g__xeCJK_punctstyle_clist
1265     {
1266       \str_if_eq_x:nnF \c__xeCJK_ps_plain_tl {#1}
1267       {
1268         \tl_gset:cx { \l__xeCJK_font_coord_tl/##1/kern/#1-#1 }
1269         { - \dim_use:N \l__xeCJK_tmpb_dim }
1270       }
1271     }
1272   }
1273 }
(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。

```
1274 \cs_new_protected_nopar:Npn \__xeCJK_get_glyph_bounds:nNN #1#2#3
1275 {
1276   \dim_set:Nn #3
```

```

1277 {
1278   \XeTeXglyphbounds
1279   \str_case_x:nnn {#1}
1280   {
1281     { left } { \c_one } { top } { \c_two }
1282     { right } { \c_three } { bottom } { \c_four }
1283   }
1284   { \c_three }
1285   \int_eval:n { \XeTeXcharglyph `#2 }
1286 }
1287 }
(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。

```

1288 \NewDocumentCommand \xeCJKsetkern { m m m }
1289 { \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。

```

1290 \keys_define:nn { xeCJK / options }
1291 {
1292   AutoFallback .choice:,
1293   AutoFallback / true .code:n = { \xeCJKenablefallback },
1294   AutoFallback / false .code:n = { \xeCJKdisablefallback },
1295   AutoFallback .default:n = { true },
1296   fallback .meta:n = { AutoFallback = true },
1297 }
(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 1298 \bool_new:N \l__xeCJK_fallback_bool
1299 \NewDocumentCommand \xeCJKenablefallback { }
1300 {
1301   \bool_if:NF \l__xeCJK_fallback_bool
1302   {
1303     \bool_set_true:N \l__xeCJK_fallback_bool
1304     \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1305     \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_testsymbol:N
1306   }
1307 }
1308 \NewDocumentCommand \xeCJKdisablefallback { }
1309 {
1310   \bool_if:NT \l__xeCJK_fallback_bool
1311   {
1312     \bool_set_false:N \l__xeCJK_fallback_bool
1313     \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N
1314   }
1315 }
(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。

```
1316 \cs_new_protected_nopar:Npn \__xeCJK_fallback_testsymbol:N #1
1317 {
1318   \font_glyph_if_exist:NnTF \tex_font:D {'#1}
1319   { \__xeCJK_fallback_save_CJKsymbol:N {'#1} }
1320   {
1321     \__xeCJK_family_if_exist:xTF { \l_xeCJK_family_tl/FallBack }
1322     { {
1323       \tl_put_right:Nn \l_xeCJK_family_tl { /FallBack }
1324       \xeCJK_select_font:
1325       \__xeCJK_fallback_testsymbol:N {'#1}
1326     } }
1327     {
1328       \__xeCJK_warning:nx { fallback } {'#1}
1329       \__xeCJK_fallback_save_CJKsymbol:N {'#1}
1330     }
1331   }
1332 }
1333 \__xeCJK_msg_new:nn { fallback }
1334 {
1335   CJKfamily~'\l_xeCJK_family_tl'~
1336   ( \prop_get:NV \g__xeCJK_family_prop \l_xeCJK_family_tl )~
1337   does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {'#1}).\\
1338 }
(End definition for \__xeCJK_fallback_testsymbol:N)
```

\setCJKfallbackfamilyfont

```
\_xeCJK_set_family_fallback:nnn 1339 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1340 {
1341   \exp_args:Nx \tl_if_in:nnTF {#3} { , }
1342   { \__xeCJK_set_family_fallback:nnn {#1} {#2} {#3} }
1343   { \xeCJK_set_family:xnn { #1/FallBack } {#2} {#3} }
1344 }
1345 \cs_new_protected_nopar:Npn \__xeCJK_set_family_fallback:nnn #1#2#3
1346 {
1347   \group_begin:
1348   \tl_set:Nx \l__xeCJK_family_fb_tl {#1}
1349   \prop_get:NVNF \g__xeCJK_family_prop \l__xeCJK_family_fb_tl \l__xeCJK_fontname_tl
1350   { \tl_clear:N \l__xeCJK_fontname_tl }
1351   \clist_map_inline:xn {#3}
1352   {
1353     \tl_put_right:Nn \l__xeCJK_family_fb_tl { /FallBack }
1354     \__xeCJK_get_sub_features:nn \l__xeCJK_family_fb_tl {##1}
1355     \clist_put_left:cx { l__xeCJK_ \l__xeCJK_family_fb_tl _fontfeat_clist } {#2}
1356     \xeCJK_set_family:xcc \l__xeCJK_family_fb_tl
1357     { l__xeCJK_ \l__xeCJK_family_fb_tl _fontfeat_clist }
1358     { l__xeCJK_ \l__xeCJK_family_fb_tl _fontname_tl }
1359   }
1360   \group_end:
1361 }
(End definition for \setCJKfallbackfamilyfont This function is documented on page 8.)
```

5.10 CJK 字体族声明方式

```

1362 \bool_new:N \g__xeCJK_AutoFakeBold_bool
1363 \bool_new:N \g__xeCJK_AutoFakeSlant_bool
1364 \fp_new:N \g__xeCJK_EmboldenFactor_fp
1365 \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
1366 \keys_define:nn { xeCJK / options }
1367 {
1368   AutoFakeBold .choice:,
1369   AutoFakeBold / true .code:n =
1370     { \bool_set_true:N \g__xeCJK_AutoFakeBold_bool },
1371   AutoFakeBold / false .code:n =
1372     { \bool_set_false:N \g__xeCJK_AutoFakeBold_bool },
1373   AutoFakeBold / unknown .code:n =
1374     {
1375       \bool_set_true:N \g__xeCJK_AutoFakeBold_bool
1376       \fp_set:Nn \g__xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1377     },
1378   AutoFakeBold .default:n = { true },
1379   AutoFakeSlant .choice:,
1380   AutoFakeSlant / true .code:n =
1381     { \bool_set_true:N \g__xeCJK_AutoFakeSlant_bool },
1382   AutoFakeSlant / false .code:n =
1383     { \bool_set_false:N \g__xeCJK_AutoFakeSlant_bool },
1384   AutoFakeSlant / unknown .code:n =
1385     {
1386       \bool_set_true:N \g__xeCJK_AutoFakeSlant_bool
1387       \fp_set:Nn \g__xeCJK_SlantFactor_fp { \l_keys_value_tl }
1388     },
1389   AutoFakeSlant .default:n = { true },
1390   EmboldenFactor .fp_set:N = \g__xeCJK_EmboldenFactor_fp,
1391   SlantFactor .fp_set:N = \g__xeCJK_SlantFactor_fp,
1392   BoldFont .meta:n = { AutoFakeBold = true },
1393   boldfont .meta:n = { AutoFakeBold = true },
1394   SlantFont .meta:n = { AutoFakeSlant = true },
1395   slantfont .meta:n = { AutoFakeSlant = true },
1396 }

```

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

\keys_set_known:nxN

```

1397 \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。

```

\__xeCJK_sub_key_clist
1398 \clist_new:N \g__xeCJK_sub_key_clist
1399 \cs_new_protected_nopar:Npn \__xeCJK_new_sub_key:n #1
1400 {
1401   \clist_gput_right:Nx \g__xeCJK_sub_key_clist {#1}
1402   \keys_define:nn { xeCJK / features }
1403   {
1404     #1 .code:n =

```

```

1405     {
1406         \__xeCJK_if_blank_x:nTF {##1}
1407         {
1408             \bool_set_false:c { l__xeCJK_copy_#1_bool }
1409             \bool_set_false:c { l__xeCJK_add_#1_bool }
1410             \tl_put_right:Nx \l__xeCJK_family_tl { /#1 }
1411         }
1412         {
1413             \clist_put_right:Nx \l__xeCJK_sub_key_clist {#1}
1414             \str_if_eq_x:nnTF {##1} *
1415             {
1416                 \bool_set_true:c { l__xeCJK_copy_#1_bool }
1417                 \bool_set_false:c { l__xeCJK_add_#1_bool }
1418             }
1419             {
1420                 \bool_set_false:c { l__xeCJK_copy_#1_bool }
1421                 \bool_set_true:c { l__xeCJK_add_#1_bool }
1422                 \__xeCJK_get_sub_features:nn {#1} {##1}
1423             }
1424         }
1425     },
1426     #1 .default:n = \c_empty_tl,
1427 }
1428 }
1429 \cs_generate_variant:Nn \__xeCJK_new_sub_key:n { x }
1430 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
1431 {
1432     \tl_set:Nx \l__xeCJK_tmpa_tl {#2}
1433     \clist_clear:N \l__xeCJK_sub_fontfeat_clist
1434     \__xeCJK_gobble_brace:N \l__xeCJK_tmpa_tl
1435     \exp_args:No \tl_if_head_eq_charcode:nNTF \l__xeCJK_tmpa_tl [ % ]
1436     {
1437         \exp_after:wN \__xeCJK_get_sub_features_aux:n \l__xeCJK_tmpa_tl \c_empty_tl
1438         \__xeCJK_if_blank_x:nT \l__xeCJK_sub_fontname_tl
1439         {
1440             \tl_set:Nx \l__xeCJK_sub_fontname_tl \l__xeCJK_tmpa_tl
1441             \clist_clear:N \l__xeCJK_sub_fontfeat_clist
1442         }
1443     }
1444     { \tl_set:Nx \l__xeCJK_sub_fontname_tl \l__xeCJK_tmpa_tl }
1445     \__xeCJK_if_blank_x:nTF \l__xeCJK_sub_fontname_tl
1446     { \tl_set:Nx \l__xeCJK_sub_fontname_tl \l__xeCJK_fontname_tl }
1447     { \tl_replace_all:Nnx \l__xeCJK_sub_fontname_tl * \l__xeCJK_fontname_tl }
1448     \clist_set:cx { l__xeCJK_#1_fontfeat_clist } \l__xeCJK_sub_fontfeat_clist
1449     \tl_set:cx { l__xeCJK_#1_fontname_tl } \l__xeCJK_sub_fontname_tl
1450 }
(End definition for \__xeCJK_new_sub_key:n and \g__xeCJK_sub_key_clist)

\__xeCJK_gobble_brace:N
\__xeCJK_get_sub_features:nn 1451 \cs_new_protected_nopar:Npn \__xeCJK_gobble_brace:N #1
\__xeCJK_get_sub_features_aux:n 1452 {
1453     \exp_args:No \tl_if_head_is_group:nT #1

```

```

1454     {
1455         \exp_last_unbraced:NNo \tl_set:Nn #1 #1
1456         \__xeCJK_gobble_brace:N #1
1457     }
1458 }
1459 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features_aux:n [#1] #2
1460 {
1461     \clist_set:Nx \l__xeCJK_sub_fontfeat_clist {#1}
1462     \tl_set:Nx \l__xeCJK_sub_fontname_tl {#2}
1463 }
(End definition for \__xeCJK_gobble_brace:N, \__xeCJK_get_sub_features:nn, and \__xeCJK_get_sub_features_aux:n)

```

FallBack

```

1464 \__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 1465 \cs_new_nopar:Npn \__xeCJK_map_features_id:n #1
      ItalicFont 1466 { \prop_get:Nn \g__xeCJK_features_id_prop {#1} }
      BoldItalicFont 1467 \prop_new:N \g__xeCJK_features_id_prop
      SlantedFont 1468 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
      BoldSlantedFont 1469 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
      BoldFeatures 1470 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
      ItalicFeatures 1471 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
      BoldItalicFeatures 1472 \prop_put:Nnn \g__xeCJK_features_id_prop { bfsl } { BoldSlanted }
      SlantedFeatures 1473 \prop_map_inline:Nn \g__xeCJK_features_id_prop
      BoldSlantedFeatures 1474 {
1475     \keys_define:nn { xeCJK / features }
1476     {
1477         #2Font .tl_set_x:c = { l__xeCJK_fontname_#1_tl },
1478         #2Features .tl_set_x:c = { l__xeCJK_fontfeat_#1_clist } ,
1479     }
1480 }
(End definition for \g__xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

Mono

```

1481 \keys_define:nn { xeCJK / features }
1482 {
1483     AutoFakeBold .choice:,
1484     AutoFakeBold / false .code:n =
1485         { \bool_set_false:N \l__xeCJK_AutoFakeBold_bool },
1486     AutoFakeBold / unknown .code:n =
1487         {
1488             \bool_set_true:N \l__xeCJK_AutoFakeBold_bool
1489             \fp_set:Nn \l__xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1490         },
1491     AutoFakeBold .default:n = { \g__xeCJK_EmboldenFactor_fp },
1492     AutoFakeSlant .choice:,
1493     AutoFakeSlant / false .code:n =
1494         { \bool_set_false:N \l__xeCJK_AutoFakeSlant_bool },
1495     AutoFakeSlant / unknown .code:n =
1496         {

```

```

1497         \bool_set_true:N \l__xeCJK_AutoFakeSlant_bool
1498         \fp_set:Nn \l__xeCJK_SlantFactor_fp { \l_keys_value_tl }
1499     },
1500     AutoFakeSlant .default:n = { \g__xeCJK_SlantFactor_fp },
1501     Mono .choice:,
1502     Mono / Exspace .code:n =
1503     {
1504         \__xeCJK_set_monoexspace:
1505         \tl_clear:N \l__xeCJK_monoscale_tl
1506     },
1507     Mono / Scale .code:n =
1508     {
1509         \__xeCJK_set_monoscale:
1510         \tl_set:Nx \l__xeCJK_monoscale_tl { Scale = { \fp_use:N \g__xeCJK_monoscale_fp } }
1511     }
1512 }

```

(End definition for AutoFakeBold, AutoFakeSlant, and Mono)

__xeCJK_set_init:

```

1513 \cs_new_protected_nopar:Npn \__xeCJK_set_init:
1514 {
1515     \tl_clear:N \l__xeCJK_monoscale_tl
1516     \int_gincr:N \g__xeCJK_fontspec_int
1517     \prop_clear:N \l__xeCJK_add_font_prop
1518     \prop_map_inline:Nn \g__xeCJK_features_id_prop
1519     {
1520         \tl_clear:c { l__xeCJK_fontname_##1_tl }
1521         \clist_clear:c { l__xeCJK_fontfeat_##1_clist }
1522     }
1523     \clist_clear:N \l__xeCJK_pass_features_clist
1524     \clist_clear:N \l__xeCJK_sub_key_clist
1525     \clist_map_inline:Nn \g__xeCJK_sub_key_clist
1526     {
1527         \bool_set_false:c { l__xeCJK_copy_##1_bool }
1528         \bool_set_false:c { l__xeCJK_add_##1_bool }
1529     }
1530     \bool_set_eq:NN \l__xeCJK_AutoFakeBold_bool \g__xeCJK_AutoFakeBold_bool
1531     \bool_set_eq:NN \l__xeCJK_AutoFakeSlant_bool \g__xeCJK_AutoFakeSlant_bool
1532     \fp_set_eq:NN \l__xeCJK_EmboldenFactor_fp \g__xeCJK_EmboldenFactor_fp
1533     \fp_set_eq:NN \l__xeCJK_SlantFactor_fp \g__xeCJK_SlantFactor_fp
1534 }

```

(End definition for __xeCJK_set_init:)

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

```

1535 \cs_new_protected_nopar:Npn \xeCJK_set_family:xnn #1#2#3
1536 {
1537     \group_begin:
1538     \__xeCJK_set_init:
1539     \tl_set:Nx \l__xeCJK_family_tl {#1}
1540     \clist_set:Nx \l__xeCJK_fontoptions_clist {#2}
1541     \tl_set:Nx \l__xeCJK_fontname_tl {#3}
1542     \keys_set_known:nxN { xeCJK / features }

```



```

1543     { \g__xeCJK_default_features_clist, #2 } \l__xeCJK_pass_features_clist
1544 \__xeCJK_parse_features:
1545 \__xeCJK_pass_features:
1546 \__xeCJK_check_family:V \l__xeCJK_family_tl
1547 \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_family_tl }
1548 {
1549     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
1550     { \l__xeCJK_pass_features_clist } { \l__xeCJK_fontname_tl }
1551     \prop_gput:NnV \exp_not:N \g__xeCJK_family_name_prop
1552     { \l__xeCJK_family_tl } \exp_not:N \l__xeCJK_fontspec_family_tl
1553     \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_family_tl }
1554     \exp_not:n
1555     { {
1556         \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1557         \exp_not:N \fontfamily { \l__xeCJK_fontspec_family_tl }
1558         \exp_not:N \selectfont
1559     } }
1560 }
1561 \__xeCJK_save_family_info:
1562 \__xeCJK_add_sub_block:
1563 \group_end:
1564 }
1565 \cs_generate_variant:Nn \xeCJK_set_family:xnn { xcc }
1566 (End definition for \xeCJK_set_family:xnn)

```

__xeCJK_check_family:n

```

1566 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
1567 {
1568     \prop_get:NnNT \g__xeCJK_family_prop {#1} \l__xeCJK_tmpa_tl
1569     {
1570         \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl { }
1571         \__xeCJK_warning:nxx { CJKfamily-redef } {#1} \l__xeCJK_tmpa_tl
1572     }
1573 }
1574 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
1575 \__xeCJK_msg_new:nn { CJKfamily-redef } { Redefining~CJKfamily~`#1'~(#2). }
1576 (End definition for \__xeCJK_check_family:n)

```

__xeCJK_add_font:nn

__xeCJK_add_font_if_new:nn

```

1576 \cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
1577 {
1578     \prop_put:Nxx \l__xeCJK_add_font_prop
1579     { \__xeCJK_map_features_id:n {#1} Font } {#2}
1580 }
1581 \cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
1582 {
1583     \prop_put_if_new:Nxx \l__xeCJK_add_font_prop
1584     { \__xeCJK_map_features_id:n {#1} Font } {#2}
1585 }
1586 \cs_generate_variant:Nn \__xeCJK_add_font:nn { nx, nc, nV, nv }
1587 \cs_generate_variant:Nn \__xeCJK_add_font_if_new:nn { nx, nc, nV, nv }

```

(End definition for __xeCJK_add_font:nn and __xeCJK_add_font_if_new:nn)

```

1588 \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1589 \cs_generate_variant:Nn \prop_get:Nn { NV }
1590 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1591 \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1592 \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

__xeCJK_add_fake_bold:n

```

\__xeCJK_add_fake_slant:n 1593 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
1594 {
1595   \clist_put_left:cx { l__xeCJK_fontfeat_#1_clist }
1596   { FakeBold = { \fp_use:N \l__xeCJK_EmboldenFactor_fp } }
1597 }
1598 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
1599 {
1600   \clist_put_left:cx { l__xeCJK_fontfeat_#1_clist }
1601   { FakeSlant = { \fp_use:N \l__xeCJK_SlantFactor_fp } }
1602 }

```

(End definition for __xeCJK_add_fake_bold:n and __xeCJK_add_fake_slant:n)

__xeCJK_parse_features:

```

\__xeCJK_parse_features:n 1603 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:
1604 { \prop_map_inline:Nn \g__xeCJK_features_id_prop { \__xeCJK_parse_features:n {##1} } }
1605 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:n #1
1606 {
1607   \__xeCJK_if_font_select:nTF {#1}
1608   {
1609     \__xeCJK_add_font:nv {#1} { l__xeCJK_fontname_#1_tl }
1610     \__xeCJK_if_it_or_sl:nTF {#1}
1611     {
1612       \__xeCJK_if_font_select:nF {bf#1}
1613       { \__xeCJK_add_font_if_new:nv {bf#1} { l__xeCJK_fontname_#1_tl } }
1614     }
1615     {
1616       \str_if_eq_x:nnT {#1} {bf}
1617       {
1618         \clist_map_inline:nn { it , sl }
1619         {
1620           \__xeCJK_if_font_select:nF {bf##1}
1621           { \__xeCJK_add_font:nv {bf##1} \l__xeCJK_fontname_bf_tl }
1622         }
1623       }
1624     }
1625   }
1626   { \__xeCJK_set_fake:n {#1} }
1627 }

```

(End definition for __xeCJK_parse_features: and __xeCJK_parse_features:n)

__xeCJK_if_font_select_p:n

```

\__xeCJK_if_it_or_sl_p:n 1628 \prg_new_conditional:Npnn \__xeCJK_if_font_select:n #1 { p, T, F, TF }
\__xeCJK_if_font_select:nTF 1629 {
\__xeCJK_if_it_or_sl:nTF 1630   \exp_args:Nc \__xeCJK_if_blank_x:nTF { l__xeCJK_fontname_#1_tl }

```

```

1631     \prg_return_false: \prg_return_true:
1632 }
1633 \prg_new_conditional:Npnn \__xeCJK_if_it_or_sl:n #1 { p, T, F, TF }
1634 {
1635     \bool_if:nTF { \str_if_eq_x_p:nn { it } {#1} || \str_if_eq_x_p:nn { sl } {#1} }
1636     \prg_return_true: \prg_return_false:
1637 }
(End definition for \__xeCJK_if_font_select:n and \__xeCJK_if_it_or_sl:n)

```

__xeCJK_set_fake:n

```

1638 \cs_new_protected_nopar:Npn \__xeCJK_set_fake:n #1
1639 {
1640     \str_if_eq_x:nnTF {#1} { bf }
1641     { \bool_if:NT \l__xeCJK_AutoFakeBold_bool { \__xeCJK_add_fake_bold:n {#1} } }
1642     {
1643         \bool_if:NTF \l__xeCJK_AutoFakeSlant_bool
1644         {
1645             \bool_if:nT
1646             { \__xeCJK_if_it_or_sl_p:n {#1} ||
1647             ( \str_if_eq_x_p:nn {#1} {bf} ) && ! ( \__xeCJK_if_font_select_p:n {it} ) ) ||
1648             ( \str_if_eq_x_p:nn {#1} {bfsl} ) && ! ( \__xeCJK_if_font_select_p:n {sl} ) )
1649             }
1650             { \__xeCJK_add_fake_slant:n {#1} }
1651         }
1652         { \__xeCJK_if_it_or_sl:nT {#1} { \__xeCJK_map_it_sl:n {#1} } }
1653         \bool_if:nT
1654         {
1655             \l__xeCJK_AutoFakeBold_bool
1656             && ! ( \__xeCJK_if_it_or_sl_p:n {#1} )
1657             && ! ( \__xeCJK_if_font_select_p:n {bf} )
1658             }
1659             { \__xeCJK_add_fake_bold:n {#1} }
1660     }
1661     \__xeCJK_add_font_if_new:nn {#1} *
(End definition for \__xeCJK_set_fake:n)

```

__xeCJK_map_it_sl:n

```

1662 \cs_new_protected_nopar:Npn \__xeCJK_map_it_sl:n #1
1663 {
1664     \__xeCJK_if_map_font_select:nT {#1}
1665     {
1666         \__xeCJK_add_font:nx {#1} { \__xeCJK_get_map_font:n {#1} }
1667         \__xeCJK_if_font_select:nF {bf#1}
1668         { \__xeCJK_add_font_if_new:nx {bf#1} { \__xeCJK_get_map_font:n {#1} } }
1669     }
1670 }
(End definition for \__xeCJK_map_it_sl:n)

```

__xeCJK_get_map_font:n

```

1671 \cs_new_nopar:Npn \__xeCJK_get_map_font:n #1
1672 { \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 1673 \prg_new_conditional:Npnn \__xeCJK_if_map_font_select:n #1 { p, T, F, TF }
1674 {
1675   \__xeCJK_if_blank_x:nTF { \__xeCJK_get_map_font:n {#1} }
1676   \prg_return_false: \prg_return_true:
1677 }
(End definition for \__xeCJK_if_map_font_select:n)

\__xeCJK_pass_features:
1678 \cs_new_protected_nopar:Npn \__xeCJK_pass_features:
1679 {
1680   \prop_map_inline:Nn \g__xeCJK_features_id_prop
1681   {
1682     \clist_if_empty:cF { l__xeCJK_fontfeat_##1_clist }
1683     {
1684       \clist_put_right:Nx \l__xeCJK_pass_features_clist
1685       { ##2Features = { \tl_use:c { l__xeCJK_fontfeat_##1_clist } } }
1686     }
1687   }
1688   \prop_map_inline:Nn \l__xeCJK_add_font_prop
1689   { \clist_put_right:Nx \l__xeCJK_pass_features_clist { ##1 = { ##2 } } }
1690   \__xeCJK_if_blank_x:nF \l__xeCJK_monoscale_tl
1691   { \clist_put_right:Nx \l__xeCJK_pass_features_clist \l__xeCJK_monoscale_tl }
1692 }
(End definition for \__xeCJK_pass_features:)

\g__xeCJK_family_prop
\g__xeCJK_family_name_prop 1693 \prop_new:N \g__xeCJK_family_prop
\g__xeCJK_family_options_prop 1694 \prop_new:N \g__xeCJK_family_name_prop
1695 \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:
1696 \cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
1697 {
1698   \group_begin:
1699   \clist_map_inline:Nn \g__xeCJK_CJK_subclass_clist
1700   {
1701     \clist_remove_all:Nn \l__xeCJK_fontoptions_clist {##1}
1702     \clist_remove_all:Nn \l__xeCJK_fontoptions_clist { ##1 = }
1703   }
1704   \prop_gput:NVV \g__xeCJK_family_prop \l__xeCJK_family_tl \l__xeCJK_fontname_tl
1705   \prop_gput:NVV \g__xeCJK_family_options_prop \l__xeCJK_family_tl \l__xeCJK_fontoptions_clist
1706   \group_end:
1707 }
(End definition for \__xeCJK_save_family_info:)

\__xeCJK_add_sub_block:
1708 \cs_new_protected_nopar:Npn \__xeCJK_add_sub_block:
1709 {
1710   \clist_remove_duplicates:N \l__xeCJK_sub_key_clist
1711   \clist_map_inline:Nn \l__xeCJK_sub_key_clist

```

```

1712 {
1713     \tl_set:Nx \l__xeCJK_sub_family_tl { \l__xeCJK_family_tl/##1 }
1714     \bool_if:cT { l__xeCJK_copy_##1_bool }
1715     {
1716         \__xeCJK_check_family:V \l__xeCJK_sub_family_tl
1717         \prop_get:NVNT \g__xeCJK_family_prop \l__xeCJK_family_tl \l__xeCJK_tmpa_tl
1718         { \prop_gput:NVV \g__xeCJK_family_prop \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl }
1719         \prop_get:NVNT \g__xeCJK_family_options_prop \l__xeCJK_family_tl \l__xeCJK_tmpa_clist
1720         {
1721             \clist_remove_all:Nn \l__xeCJK_tmpa_clist { ##1 = * }
1722             \prop_gput:NVV \g__xeCJK_family_options_prop \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
1723         }
1724         \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_sub_family_tl }
1725         {
1726             \__xeCJK_family_if_exist:xT { \l__xeCJK_family_tl }
1727             {
1728                 \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
1729                 { \l__xeCJK_family_tl } \exp_not:N \l__xeCJK_tmpa_tl
1730                 {
1731                     \prop_gput:NnV \exp_not:N \g__xeCJK_family_name_prop
1732                     { \l__xeCJK_sub_family_tl } \exp_not:N \l__xeCJK_tmpa_tl
1733                     \cs_gset_protected_nopar:cpx { __xeCJK/family/\l__xeCJK_sub_family_tl }
1734                     \exp_not:n
1735                     { {
1736                         \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1737                         \exp_not:N \fontfamily { \l__xeCJK_tmpa_tl }
1738                         \exp_not:N \selectfont
1739                     } }
1740                 }
1741             }
1742         }
1743     }
1744     \bool_if:cT { l__xeCJK_add_##1_bool }
1745     {
1746         \xeCJK_set_family:xcc \l__xeCJK_sub_family_tl
1747         { l__xeCJK_##1_fontfeat_clist } { l__xeCJK_##1_fontname_tl }
1748     }
1749 }
1750 }

```

(End definition for __xeCJK_add_sub_block:)

__xeCJK_copy_family:nn

```

1751 \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
1752 {
1753     \__xeCJK_family_if_exist:xT {#2}
1754     {
1755         \tl_map_inline:nn
1756         { \g__xeCJK_family_prop \g__xeCJK_family_options_prop \g__xeCJK_family_name_prop }
1757         { \prop_get:NnNT ##1 {#2} \l__xeCJK_tmpa_tl { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_tl } }
1758         \cs_gset_eq:cc { __xeCJK/family/#1 } { __xeCJK/family/#2 }
1759     }
1760 }
1761 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }

```

(End definition for __xeCJK_copy_family:nn)

5.11 字体切换

\l__xeCJK_font_coor_tl 缓存当前字体的原始格式,以加速编译。

```
\xeCJK_select_font: 1762 \tl_set:Nn \l__xeCJK_font_coor_tl { __xeCJK/\l_xeCJK_family_tl/\f@series/\f@shape/\f@size }
__xeCJK_switch_font:nn 1763 \cs_new_protected_nopar:Npn \xeCJK_select_font:
  1764 {
  1765   \cs_if_exist_use:cF { \l__xeCJK_font_coor_tl }
  1766   {
  1767     \tl_set:Nx \l__xeCJK_current_coor_tl \l__xeCJK_font_coor_tl
  1768     \__xeCJK_family_if_exist:xF { \l_xeCJK_family_tl }
  1769     { \use:c { __xeCJK/family/\l_xeCJK_family_tl } }
  1770     { \use_none:nnnn }
  1771     \exp_last_unbraced:NNV \cs_gset_eq:cN \l__xeCJK_current_coor_tl \tex_font:D
  1772   }
  1773 }
  1774 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
  1775 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
  1776 {
  1777   \str_if_eq_x:nnF {#1} {#2}
  1778   {
  1779     \__xeCJK_info:nxx { CJK-block } {#1} {#2}
  1780     \tl_remove_all:Nn \l_xeCJK_family_tl { /#1 }
  1781     \str_if_eq_x:nnF {#2} { CJK }
  1782     {
  1783       \tl_set:Nx \l__xeCJK_tmpa_tl \l_xeCJK_family_tl
  1784       \tl_put_right:Nx \l_xeCJK_family_tl { /#2 }
  1785       \__xeCJK_family_if_exist:xF \l_xeCJK_family_tl
  1786       {
  1787         \__xeCJK_family_if_exist:xF { \CJKfamilydefault/#2 }
  1788         { \__xeCJK_copy_family:xx \l_xeCJK_family_tl { \CJKfamilydefault/#2 } }
  1789         { \__xeCJK_copy_family:xx \l_xeCJK_family_tl \l__xeCJK_tmpa_tl }
  1790       }
  1791     }
  1792   }
  1793   \xeCJK_select_font:
  1794 }
  1795 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~'#1'~to~'#2'. }
(End definition for \l__xeCJK_font_coor_tl, \xeCJK_select_font:, and \__xeCJK_switch_font:nn)
```

__xeCJK_family_if_exist:xF

```
1796 \prg_new_protected_conditional:Npnn \__xeCJK_family_if_exist:x #1 { T, F, TF }
1797 {
1798   \cs_if_free:cTF { __xeCJK/family/#1 }
1799   { \prg_return_false: }
1800   {
1801     \prop_get:NxNF \g__xeCJK_family_name_prop {#1} \l__xeCJK_family_name_tl
1802     { \use:c { __xeCJK/family/#1 } }
1803     \prg_return_true:
1804   }
1805 }
```

```

1806 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
      (End definition for \__xeCJK_family_if_exist:xF)

```

\CJKfamily 用于切换 CJK 字体族。

```

1807 \NewDocumentCommand \CJKfamily { m }
1808 {
1809   \cs_if_exist:cTF { __xeCJK/family/#1 }
1810   { \tl_set:Nx \l__xeCJK_family_tl {#1} }
1811   {
1812     \seq_if_in:NxF \g__xeCJK_unknown_family_seq {#1}
1813     {
1814       \seq_gput_right:Nx \g__xeCJK_unknown_family_seq {#1}
1815       \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
1816     }
1817   }
1818   \tex_ignorespaces:D
1819 }
1820 \seq_new:N \g__xeCJK_unknown_family_seq
1821 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
1822 {
1823   Unknown~CJK~family~'#1'~is~ignored.\\\
1824   Try~to~use~
1825   \str_case:x:nnn {#1}
1826   {
1827     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1828     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1829     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1830   }
1831   { \token_to_str:N \setCJKfamilyfont {#1} }
1832   [...]{}~to~define~it.
1833 }
      (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。

```

1834 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
1835 { \xeCJK_set_family:xnn {#1} {#2} {#3} }
1836 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1837 {
1838   \tl_set:Nx \l__xeCJK_family_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1839   \xeCJK_set_family:xnn \l__xeCJK_family_tl {#3} {#4}
1840   \cs_new_protected_nopar:Npx #2 { \exp_not:N \CJKfamily { \l__xeCJK_family_tl } }
1841 }
1842 \int_new:N \g__xeCJK_fontspec_int
1843 \int_gdecr:N \g__xeCJK_fontspec_int
1844 \NewDocumentCommand \CJKfontspec { O{} m }
1845 {
1846   \tl_set:Nx \l__xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1847   \cs_if_free:cT \l__xeCJK_CJKfontspec_id_tl
1848   {
1849     \int_gincr:N \g__xeCJK_fontspec_int
1850     \tl_gset:Nx \g__xeCJK_family_spec_tl
1851       { CJKfontspec ( \int_use:N \g__xeCJK_fontspec_int ) }

```

```

1852     \xeCJK_set_family:xnn \g__xeCJK_family_spec_tl {#1} {#2}
1853     \tl_gset:cx \l__xeCJK_CJKfontspec_id_tl \g__xeCJK_family_spec_tl
1854   }
1855   \exp_args:Nv \CJKfamily \l__xeCJK_CJKfontspec_id_tl
1856   \tex_ignorespaces:D
1857 }

```

(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
1858 \clist_new:N \g__xeCJK_default_features_clist
1859 \NewDocumentCommand \defaultCJKfontfeatures { m }
1860 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
1861 \defaultCJKfontfeatures { Script = CJK }
1862 \@onlypreamble \defaultCJKfontfeatures
1863 \NewDocumentCommand \addCJKfontfeatures { m }
1864 {
1865   \prop_if_in:NVTF \g__xeCJK_family_prop \l__xeCJK_family_tl
1866   {
1867     \group_begin:
1868     \clist_set:Nx \l__xeCJK_tmpa_clist {#1}
1869     \prop_get:NVN \g__xeCJK_family_prop \l__xeCJK_family_tl \l__xeCJK_tmpa_tl
1870     \clist_map_inline:Nn \g__xeCJK_CJK_subclass_clist
1871     {
1872       \clist_if_in:NnT \l__xeCJK_tmpa_clist {##1}
1873       {
1874         \clist_remove_all:Nn \l__xeCJK_tmpa_clist {##1}
1875         \prop_get:NxNF \g__xeCJK_family_prop { \l__xeCJK_family_tl/##1 } \l__xeCJK_tmpb_tl
1876         {
1877           \prop_get:NxNF \g__xeCJK_family_prop
1878           { \CJKfamilydefault/##1 } \l__xeCJK_tmpb_tl
1879           { \tl_set:Nx \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
1880         }
1881         \clist_set:Nx \l__xeCJK_tmpa_clist
1882         { ##1 = { [ \l__xeCJK_tmpa_clist ] { \l__xeCJK_tmpb_tl } } }
1883         \clist_map_break:
1884       }
1885     }
1886     \prop_get:NVNF \g__xeCJK_family_options_prop \l__xeCJK_family_tl \l__xeCJK_tmpb_clist
1887     { \clist_clear:N \l__xeCJK_tmpb_clist }
1888     \clist_clear:N \l__xeCJK_tmpc_clist
1889     \clist_map_inline:Nn \l__xeCJK_tmpb_clist
1890     {
1891       \clist_if_in:NnF \l__xeCJK_tmpa_clist {##1}
1892       { \clist_put_right:Nn \l__xeCJK_tmpc_clist {##1} }
1893     }
1894     \clist_put_left:Nv \l__xeCJK_tmpa_clist \l__xeCJK_tmpc_clist
1895     \use:x
1896     { \group_end: \exp_not:N \CJKfontspec [ \l__xeCJK_tmpa_clist ] { \l__xeCJK_tmpa_tl } }
1897   }
1898   { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
1899   \tex_ignorespaces:D
1900 }

```



```

1901 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
1902 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1903 {
1904   \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
1905   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1906 }
(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 1907 \NewDocumentCommand \setCJKmainfont { O{} m }
1908 { \xeCJK_set_family:xnn \CJKrmdefault {#1} {#2} }
1909 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1910 \NewDocumentCommand \setCJKsansfont { O{} m }
1911 { \xeCJK_set_family:xnn \CJKsfdefault {#1} {#2} }
(End definition for \setCJKmainfont and \setCJKsansfont These functions are documented on page 5.)

```

```

\CJKrmdefault
\CJKsfdefault 1912 \tl_if_exist:NF \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
\CJKttdefault 1913 \tl_if_exist:NF \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 1914 \tl_if_exist:NF \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1915 \tl_if_exist:NF \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1916 \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。

```

1917 \AtEndOfPackage
1918 {
1919   \etex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1920   \etex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }
1921   \etex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1922   \etex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1923   \cs_set_eq:NN \reset@font \normalfont
1924 }

```

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。

```

1925 \@onlypreamble \setCJKmainfont
1926 \@onlypreamble \setCJKmathfont
1927 \@onlypreamble \setCJKsansfont
1928 \@onlypreamble \setCJKmonofont
1929 \@onlypreamble \setCJKromanfont

```

5.12 数学字体设置

\setCJKmathfont 设置 CJK 数学字体。

```

1930 \NewDocumentCommand \setCJKmathfont { O{} m }
1931 { \xeCJK_set_family:xnn \c__xeCJK_math_tl {#1} {#2} }
(End definition for \setCJKmathfont This function is documented on page 7.)

```

CJKmath 数学公式内的 CJK 字体。

```

1932 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }
1933 \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

```

1934 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1935 {
1936   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
1937   \int_set:Nn \l__xeCJK_tmpc_int { \__xeCJK_math_type:n {#3} }
1938   \int_set:Nn \l_tmpa_int { \use:c { sym #4 } }
1939   \__xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
1940   {
1941     \XeTeXmathcode \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int \l_tmpa_int \l__xeCJK_tmpa_int
1942     \int_incr:N \l__xeCJK_tmpa_int
1943   }
1944 }
1945 \cs_new_eq:NN \__xeCJK_math_type:n \mathchar@type
(End definition for \xeCJKsetmathcode)

```

__xeCJK_set_mathfont:

```

1946 \cs_new_protected_nopar:Npn \__xeCJK_set_mathfont:
1947 {
1948   \__xeCJK_family_if_exist:xTF \c__xeCJK_math_tl
1949   { \tl_set:Nx \l__xeCJK_tmpa_tl \c__xeCJK_math_tl }
1950   {
1951     \__xeCJK_family_if_exist:xTF \CJKfamilydefault
1952     { \tl_set:Nx \l__xeCJK_tmpa_tl \CJKfamilydefault }
1953     { \tl_set_eq:NN \l__xeCJK_tmpa_tl \q_no_value }
1954   }
1955   \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_math_family_tl
1956   {
1957     \DeclareSymbolFont \c__xeCJK_math_tl \g_fontspec_encoding_tl
1958     \l__xeCJK_math_family_tl \mddefault \shapedefault
1959     \SetSymbolFont \c__xeCJK_math_tl { bold } \g_fontspec_encoding_tl
1960     \l__xeCJK_math_family_tl \bfdefault \shapedefault
1961     \clist_map_inline:nn { CJK, FullLeft, FullRight }
1962     {
1963       \clist_map_inline:cn { g__xeCJK_##1_range_clist }
1964       {
1965         \__xeCJK_set_char_class_aux:Nn \xeCJKsetmathcode {####1}
1966         \mathalpha \c__xeCJK_math_tl
1967       }
1968     }
1969   }
1970 }
(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。

```

1971 \__xeCJK_AtEndPreamble:n
1972 {
1973   \normalfont
1974   \prop_if_empty:NTF \g__xeCJK_family_prop
1975   { \__xeCJK_warning:n { no-CJKfamily } }
1976   { \bool_if:NT \g__xeCJK_math_bool { \__xeCJK_set_mathfont: } }
1977 }

```

```

1978 \__xeCJK_msg_new:nn { no-CJKfamily }
1979 {
1980   It~seems~that~you~have~not~declare~a~CJKfamily.\\
1981   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
1982   \token_to_str:N \xeCJKmainfont[...]{...}\\
1983   in~the~preamble~to~declare~the~main~CJKfamily.\\
1984 }

```

5.13 处理等宽字体和抄录环境

\g__xeCJK_monoscale_fp

\g__xeCJK_exspace_dim

```

1985 \fp_new:N \g__xeCJK_monoscale_fp
1986 \dim_new:N \g__xeCJK_exspace_dim
(End definition for \g__xeCJK_monoscale_fp and \g__xeCJK_exspace_dim)

```

\setCJKmonoscale

```

1987 \NewDocumentCommand \setCJKmonoscale { }
1988 {
1989   \CJKflexiblespacing
1990   \__xeCJK_set_monoscale:
1991   \addCJKfontfeatures { Scale = \fp_use:N \g__xeCJK_monoscale_fp }
1992 }
1993 \cs_new_protected_nopar:Npn \__xeCJK_set_monoscale:
1994 {
1995   \group_begin:
1996   \fontfamily \ttdefault \selectfont
1997   \fp_gset:Nn \g__xeCJK_monoscale_fp
1998   { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } / ( \f@size ) }
1999   \group_end:
2000 }
(End definition for \setCJKmonoscale This function is documented on page 6.)

```

\setCJKmonoexspace

```

2001 \NewDocumentCommand \setCJKmonoexspace { } { \__xeCJK_set_monoexspace: }
2002 \cs_new_protected_nopar:Npn \__xeCJK_set_monoexspace:
2003 {
2004   \group_begin:
2005   \fontfamily \ttdefault \selectfont
2006   \dim_gset:Nn \g__xeCJK_exspace_dim
2007   { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size \p@ }
2008   \group_end:
2009 }
(End definition for \setCJKmonoexspace This function is documented on page 6.)

```

\CJKfixedspacing

\CJKflexiblespacing

```

2010 \NewDocumentCommand \CJKfixedspacing { }
2011 {
2012   \bool_if:NF \l__xeCJK_fixed_spacing_bool
2013   {
2014     \bool_set_true:N \l__xeCJK_fixed_spacing_bool
2015     \tl_set_eq:NN \l__xeCJK_flexible_punctstyle_tl \l__xeCJK_punctstyle_tl
2016     \bool_set_eq:NN \l__xeCJK_fixed_xecglue_bool \l__xeCJK_xecglue_bool

```

```

2017     \cs_set_eq:NN \__xeCJK_flexible_ecglue: \CJKecglue
2018     \cs_set_eq:NN \__xeCJK_flexible_cjkg glue: \CJKglue
2019     \__xeCJK_set_monoexspace:
2020     \xeCJKsetup
2021     {
2022         PunctStyle = plain ,
2023         CJKglue     = { \skip_horizontal:N \g__xeCJK_exspace_dim } ,
2024         CJKecglue   = { \skip_horizontal:n { .5\g__xeCJK_exspace_dim } } ,
2025         xCJKecglue  = false ,
2026     }
2027 }
2028 }
2029 \NewDocumentCommand \CJKflexiblespacing { }
2030 {
2031     \bool_if:NT \l__xeCJK_fixed_spacing_bool
2032     {
2033         \bool_set_false:N \l__xeCJK_fixed_spacing_bool
2034         \exp_args:Nx \xeCJKsetup
2035         {
2036             PunctStyle = { \l__xeCJK_flexible_punctstyle_tl } ,
2037             CJKglue     = { \__xeCJK_flexible_cjkg glue: } ,
2038             CJKecglue   = { \__xeCJK_flexible_ecglue: } ,
2039             xCJKecglue  = { \bool_if:NTF \l__xeCJK_fixed_xecglue_bool { true } { false } } ,
2040         }
2041     }
2042 }
2043 \bool_new:N \l__xeCJK_fixed_spacing_bool
2044 \__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。

```

2045 \NewDocumentCommand \setCJKmonofont { s t+ O{ } m }
2046 {
2047     \IfBooleanTF {#1}
2048     { \xeCJK_set_family:xnn \CJKttdefault { Mono = Scale, #3 } {#4} }
2049     {
2050         \IfBooleanTF {#2}
2051         { \xeCJK_set_family:xnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
2052         { \xeCJK_set_family:xnn \CJKttdefault {#3} {#4} }
2053     }
2054 }

```

(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

```

2055 \bool_new:N \g__xeCJK_indent_bool
2056 \keys_define:nn { xeCJK / options }
2057 {
2058     CJKnumber      .bool_set:N = \g__xeCJK_number_bool ,
2059     indentfirst    .bool_set:N = \g__xeCJK_indent_bool ,

```

```

2060     normalindentfirst .meta:n = { indentfirst = false } ,
2061     quiet .code:n =
2062     {
2063         \msg_redirect_module:nnn { xeCJK } { warning } { info }
2064         \msg_redirect_module:nnn { xeCJK } { info } { none }
2065         \__xeCJK_if_package_loaded:nF { fontspec }
2066         { \PassOptionsToPackage { quiet } { fontspec } }
2067     },
2068     silent .code:n =
2069     {
2070         \msg_redirect_module:nnn { xeCJK } { warning } { none }
2071         \msg_redirect_module:nnn { xeCJK } { info } { none }
2072         \__xeCJK_if_package_loaded:nF { fontspec }
2073         { \PassOptionsToPackage { silent } { fontspec } }
2074     },
2075     unknown .code:n =
2076     {
2077         \__xeCJK_if_package_loaded:nTF { fontspec }
2078         { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2079         { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2080     },
2081 }
2082 \__xeCJK_msg_new:nn { key-unknown }
2083 {
2084     Sorry,~but~\l_keys_module_tl \ does~not~have~a~key~called~'#1'.\\
2085     The~key~'#1'~is~being~ignored.
2086 }

```

(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。

```

2087 \keys_set:nn { xeCJK / options }
2088 {
2089     CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
2090     CJKecglue = \c_space_token ,
2091     xCJKecglue = false ,
2092     CheckSingle = false ,
2093     CJKspace = false ,
2094     CJKmath = false ,
2095     xeCJKactive = true ,
2096     indentfirst = true ,
2097     EmboldenFactor = 4 ,
2098     SlantFactor = .167 ,
2099     PunctStyle = quanjiao ,
2100     KaiMingPunct = { . 。 ? ! } ,
2101     LongPunct = { — … … — } ,
2102     MiddlePunct = { • • • } ,
2103     AllowBreakBetweenPuncts = false ,
2104 }

```

\CJKsymbol
\CJKpunctsymbol

```

2105 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2106 \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。
2107 \ProcessKeysOptions { xeCJK / options }
2108 \RequirePackage { fontspec } [ 2011/09/13 ]

2109 \bool_if:NT \g__xeCJK_indent_bool
2110 {
2111     \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
2112     \@afterindenttrue
2113 }

```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```

2114 \NewDocumentCommand \xeCJKsetup { m }
2115 {
2116     \keys_set:nn { xeCJK / options } {#1}
2117     \tex_ignorespaces:D
2118 }
(End definition for \xeCJKsetup This function is documented on page 3.)

```

\xeCJKsetemboldenfactor

\xeCJKsetslantfactor

```

2119 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
2120 { \xeCJKsetup { EmboldenFactor = {#1} } }
2121 \NewDocumentCommand \xeCJKsetslantfactor { m }
2122 { \xeCJKsetup { SlantFactor = {#1} } }
(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor)

```

\punctstyle

\xeCJKplainchr

```

2123 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2124 \NewDocumentCommand \xeCJKplainchr { } { \punctstyle { plain } }
(End definition for \punctstyle and \xeCJKplainchr)

```

\CJKsetecglue

```

2125 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2126 \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。

```

2127 \cs_new_protected_nopar:Npn \__xeCJK_fix_itcorr:
2128 {
2129     \scan_stop:
2130     \__xeCJK_if_last_node:nTF { default }
2131     {
2132         \tex_unkern:D \tex_unkern:D
2133         \__xeCJK_itcorr_aux
2134         { \__xeCJK_make_node:n { default_itcorr } }
2135     }
2136     { \__xeCJK_itcorr_aux }
2137 }

```

```

2138 \cs_new_eq:NN \__xeCJK_itcorr_aux \/  

2139 \cs_set_protected_nopar:Npn \/  

2140 {  

2141   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero  

2142     { \__xeCJK_fix_itcorr: } { \__xeCJK_itcorr_aux }  

2143 }  

2144 <@@=  

2145 \cs_set_eq:NN \@@italiccorr \/  

2146 <@@=xeCJK>  

(End definition for \__xeCJK_fix_itcorr:)

```

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

```

\c__xeCJK_Others_class_int
2147 \__xeCJK_AfterEndPreamble:n
2148 {
2149   \int_compare:nNnF
2150     { \c_three + ( \seq_count:N \g__xeCJK_class_seq ) } = \xe@alloc@intercharclass
2151     {
2152       \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
2153       {
2154         \seq_if_in:NnF \g__xeCJK_class_seq {#1}
2155         { \__xeCJK_set_others_toks:n {#1} }
2156       }
2157     }
2158 }
2159 \cs_new_protected_nopar:Npn \__xeCJK_set_others_toks:n #1
2160 {
2161   \int_set:Nn \c__xeCJK_Others_class_int {#1}
2162   \clist_map_inline:Nn \g__xeCJK_CJK_class_clist
2163   {
2164     \__xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2165     \__xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2166     \exp_args:Nnnx \__xeCJK_app_inter_class_toks:nnn {##1} { Others }
2167     { \__xeCJK_get_inter_class_toks:nn { Default } { Others } }
2168     \exp_args:Nnnx \__xeCJK_pre_inter_class_toks:nnn { Others } {##1}
2169     { \__xeCJK_get_inter_class_toks:nn { Others } { Default } }
2170     \__xeCJK_if_blank_x:nT
2171     { \__xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2172     {
2173       \__xeCJK_copy_inter_class_toks:nnnn
2174       { Others } { Boundary } { Default } { Boundary }
2175     }
2176     \__xeCJK_if_blank_x:nT
2177     { \__xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2178     {
2179       \__xeCJK_copy_inter_class_toks:nnnn
2180       { Boundary } { Others } { Boundary } { Default }
2181     }
2182   }
2183 }
2184 \int_new:N \c__xeCJK_Others_class_int
(End definition for \__xeCJK_set_others_toks:n and \c__xeCJK_Others_class_int)

```

__xeCJK_patch:Nnn 给已有宏内容前后附加补丁。

```
2185 \cs_new_protected_nopar:Npn \__xeCJK_patch:Nnn #1#2#3
2186 { \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。

```
2187 \__xeCJK_AfterPreamble:n
2188 {
2189   \tl_map_inline:nn
2190   {
2191     \textellipsis \textendash \textperiodcentered \textcentereddot
2192     \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2193   }
2194   {
2195     \__xeCJK_patch:Nnn #1
2196     { \__xeCJK_group_begin: \makexeCJKinactive }
2197     { \__xeCJK_group_end: }
2198   }
2199   \__xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2200   \cs_set_eq:NN \__xeCJK_aux_r \r
2201   \cs_set_protected_nopar:Npn \r #1 { { \makexeCJKinactive \__xeCJK_aux_r {#1} } }
2202   \__xeCJK_if_package_loaded:nT { pifont }
2203   {
2204     \RenewDocumentCommand \Pifont { m }
2205     { \makexeCJKinactive \usefont {U} {#1} {m} {n} }
2206   }
2207 }
2208 \cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
2209 \cs_new_eq:NN \__xeCJK_group_end: \group_end:
```

简单处理与 hyperref 宏包的兼容问题。

```
2210 \__xeCJK_AfterEndPreamble:n
2211 {
2212   \bool_if:nT
2213   {
2214     \__xeCJK_if_package_loaded_p:n { hyperref } &&
2215     \tl_if_exist_p:N \pdfstringdefPreHook
2216   }
2217   {
2218     \tl_gput_right:Nn \pdfstringdefPreHook
2219     {
2220       \cs_set_eq:NN \makexeCJKinactive \prg_do_nothing:
2221       \cs_set_eq:NN \__xeCJK_group_begin: \prg_do_nothing:
2222       \cs_set_eq:NN \__xeCJK_group_end: \prg_do_nothing:
2223       \cs_set_eq:NN \CJKfamily \use_none:n
2224     }
2225   }
2226 }
```

禁止在 xeCJK 宏包后再载入 CJK 宏包。

```
2227 \tl_set:cn { ver@CJK.sty } { 2050/01/01 }
```


\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```
2228 \cs_if_free:NT \CJK@ifundefined
2229 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2230 \NewDocumentCommand \xeCJKcaption { o m }
2231 {
2232   \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
2233   \cs_set_nopar:Npx \__xeCJK_reset_at_catcode:
2234   {
2235     \exp_not:n { \char_set_catcode:nn { \@ } }
2236     { \char_value_catcode:n { \@ } }
2237   }
2238   \char_set_catcode_letter:N \@
2239   \file_input:n { #2.cpx }
2240   \__xeCJK_reset_at_catcode:
2241   \XeTeXdefaultencoding "UTF-8"
2242 }
```

(End definition for \xeCJKcaption)

__xeCJK_ULprepunctchar:n

__xeCJK_ULpostpunctchar:n

__xeCJK_ULroutines:

```
2243 \cs_new_protected_nopar:Npn \__xeCJK_ULprepunctchar:n #1
2244 {
2245   { \makexeCJKinactive \CJKpunctsymbol {#1} \__xeCJK_no_break: }
2246   \tex_ignorespaces:D
2247 }
2248 \cs_new_protected_nopar:Npn \__xeCJK_ULpostpunctchar:n #1
2249 {
2250   { \makexeCJKinactive \CJKpunctsymbol {#1} }
2251   \__xeCJK_ignorespaces:
2252 }
2253 \cs_new_protected_nopar:Npn \__xeCJK_ULroutines:
2254 {
2255   \__xeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKecglue \CJKsymbol }
2256   \__xeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2257   \__xeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKecglue \CJKsymbol }
2258   \__xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \__xeCJK_Boundary_and_CJK: }
2259   \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2260   {
2261     \__xeCJK_inter_class_toks:nnn {##1} { FullLeft } { \__xeCJK_ULprepunctchar:n }
2262     \__xeCJK_inter_class_toks:nnn {##1} { FullRight } { \__xeCJK_ULpostpunctchar:n }
2263   }
2264   \__xeCJK_UL_subclass_patch_tl
2265 }
```

(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。

```
2266 \__xeCJK_AfterPreamble:n
2267 {
2268   \cs_if_exist:NT \UL@hook
2269   {
2270     \addto@hook \UL@hook
2271     {
2272       \cs_set_eq:NN \__xeCJK_UL_CJKsymbol \CJKsymbol
2273       \cs_set_eq:NN \__xeCJK_UL_CJKpunctsymbol \CJKpunctsymbol

```

```

2274         \cs_set_protected_nopar:Npn \CJKsymbol #1
2275         {
2276             { \xeCJK_select_font: \__xeCJK_UL_CJKsymbol {#1} }
2277             { \__xeCJK_make_node:n { CJK } } \__xeCJK_ignorespaces:
2278         }
2279         \cs_set_protected_nopar:Npn \CJKpunctsymbol #1
2280         { { \xeCJK_select_font: \__xeCJK_UL_CJKpunctsymbol {#1} } }
2281         \__xeCJK_ULroutines:
2282     }
2283 }
2284 \cs_if_exist:NT \XeTeX@CJKfntef@hook
2285 {
2286     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2287     { \xeCJK_select_font: \makeCJKinactive }
2288 }
2289 }
    使用 CJKnumb 宏包。
2290 \bool_if:NT \g__xeCJK_number_bool
2291 {
2292     \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2293     \cs_set_protected_nopar:Npn \CJKaddEncHook #1#2
2294     { \cs_set_nopar:cpn { \__xeCJK_enc_#1 } {#2} }
2295     \cs_set_protected_nopar:Npn \Unicode #1#2
2296     { \tex_char:D \int_eval:n { (#1) * \c_two_hundred_fifty_six + (#2) } }
2297     \RequirePackage { CJKnumb }
2298     \use:c { \__xeCJK_enc_\CJK@UnicodeEnc }
2299     \tl_set:Nn \CJK@tenthousand { 万 }
2300     \tl_set:Nn \CJK@hundredmillion { 亿 }
2301 }
2302 </package>

```

6 例子

6.1 xeCJK-example-autofake.tex

```

1 <*ex-autofake>
2 \documentclass{article}
3 \usepackage[AutoFakeBold,AutoFakeSlant]{xeCJK}
4 \setCJKmainfont[BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsun.ttc}
5 \setCJKsansfont[AutoFakeSlant=false,
6   BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsun.ttc}
7 \setCJKmonofont[ItalicFont=simkai.ttf]{simsun.ttc}
8 \begin{document}
9 \centering
10 \begin{tabular}{l l l l l}
11 \hline
12 {\bf rm} & & md & up & \verb|\rmfamily\mdseries\upshape| & &
13           & & & & {\rmfamily\mdseries\upshape English 中文字体} & \\\
14           & & md & it & \verb|\rmfamily\mdseries\itshape| & &
15           & & & & {\rmfamily\mdseries\itshape English 中文字体} & \\\

```

```

16      & md & sl & \verb|\rmfamily\mdseries\slshape| &
17      {\rmfamily\mdseries\slshape English 中文字体} \\ \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{|\c>\ttfamily U+}|}}
72 北 & 3400 & 西 & 3401 & 尗 & 3402 & 个 & 3403 & \\
73 丰 & 3404 & 乂 & 3405 & 冂 & 3406 & 𠂔 & 3407 & \\
74 仝 & 3408 & 𠂔 & 3409 & 𠂔 & 340A & 𠂔 & 340B & \\
75 𠂔 & 340C & 𠂔 & 340D & 𠂔 & 340E & 𠂔 & 341F & \\
76 𠂔 & 3410 & 𠂔 & 3411 & 𠂔 & 3412 & 𠂔 & 3413 & \\
77 𠂔 & 3414 & 𠂔 & 3415 & 𠂔 & 3416 & 𠂔 & 3417 & \\
78 𠂔 & 3418 & 𠂔 & 3419 & 𠂔 & 341A & 𠂔 & 341B & \\
79 𠂔 & 341C & 𠂔 & 341D & 𠂔 & 341E & 𠂔 & 341F & \\[1ex]
80 𠂔 & 20000 & 𠂔 & 20001 & 𠂔 & 20002 & 𠂔 & 20003 & \\
81 𠂔 & 20004 & 𠂔 & 20005 & 𠂔 & 20006 & 𠂔 & 20007 & \\
82 𠂔 & 20008 & 𠂔 & 20009 & 𠂔 & 2000A & 𠂔 & 2000B & \\
83 𠂔 & 2000C & 𠂔 & 2000D & 𠂔 & 2000E & 𠂔 & 2000F & \\
84 𠂔 & 20010 & 𠂔 & 20011 & 𠂔 & 20012 & 𠂔 & 20013 & \\
85 𠂔 & 20014 & 𠂔 & 20015 & 𠂔 & 20016 & 𠂔 & 20017 & \\
86 𠂔 & 20018 & 𠂔 & 20019 & 𠂔 & 2001A & 𠂔 & 2001B & \\
87 𠂔 & 2001C & 𠂔 & 2001D & 𠂔 & 2001E & 𠂔 & 2001F & \\
88 \end{tabular}
89 \end{table}
90 \end{document}
91 </ex-fallback>

```

6.3 xeCJK-example-subCJKblock.tex

```

92 <*ex-block>
93 \documentclass{article}
94 \usepackage{xeCJK}
95 \usepackage{array}
96 \xeCJKDeclareSubCJKBlock{Ext-A} { "3400 -> "4DBF }
97 \xeCJKDeclareSubCJKBlock{Ext-B} { "20000 -> "2A6DF }
98 \xeCJKDeclareSubCJKBlock{Kana} { "3040 -> "309F, "30A0 -> "30FF, "31F0 -> "31FF, }
99 \xeCJKDeclareSubCJKBlock{Hangul}{ "1100 -> "11FF, "3130 -> "318F, "A960 -> "A97F, "AC00 -
> "D7AF }
100 \setCJKmainfont[Ext-A=SimHei,Ext-B=SimSun-ExtB]{SimSun}
101 \setCJKmainfont[Kana]{Meiryo}
102 \setCJKmainfont[Hangul]{Malgun Gothic}
103 \parindent=2em
104 \begin{document}
105 \long\def\showtext{%
106 中日韩越统一表意文字（英语：CJKV Unified Ideographs），旧称中日韩统一表意文字（英
语：CJK Unified Ideographs），也称统一汉字（英语：UniHan），目的是要把分别来自中
文、日文、韩文、越文、壮文中，对于相同起源、本义相同、形状一样或稍异的表意文字
（主要为汉字，但也有仿汉字如：方块壮字、日文汉字（かんじ / kanji）、韩文汉字（한
자 / hanja）、越南的喃字（Ch Nôm）与越文汉字（Ch Nho，在越南也称作儒字），应赋予
其在 ISO 10646 及统一码标准中有相同编码。此计划原本只包含中文、日文及韩文中所使用的
汉字，是以旧称中日韩统一表意文字（CJK）。后来，此计划加入了越文的喃字，所以合称中
日韩越统一表意文字（CJKV）。

```

107

108 CJK 統合漢字（シージェーケーとうごうかんじ、CJK Unified Ideographs）は、ISO/IEC 10646 (Universal Multiple-Octet Coded Character Set, 略称 UCS) および Unicode にて採用されている符号化用漢字集合およびその符号表である。CJK 統合漢字の名称は、中国語 (Chinese)、日本語 (Japanese)、韓国語 (Korean) で使われている漢字をひとまとめにしたことからきている。CJK 統合漢字の初版である Unified Repertoire and Ordering (URO) 第二版は 1992 年に制定されたが、1994 年にベトナムで使われていた漢字も含めることにしたため、CJKV と呼ばれる事もある。CJKV は、中国語・日本語・韓国語・ベトナム語 (英語: Chinese-Japanese-Korean-Vietnamese) の略。特に、その 4 言語で共通して使われる、または使われていた文字体系である漢字 (チュノムを含む) のこと。ソフトウェアの国際化、中でも文字コードに関する分野で用いられる。

109

110 \CJKspace

111 한중일월 통합 한자 (또는 한중일 통합 한자) 는 유니코드에 담겨 있는 한자들의 집합으로, 한국, 중국, 일본에서 쓰이는 한자를 묶은 것이기 때문에 머리 글자를 따서 한중일 (CJK) 통합 한자라고 불렀는데, 최근에는 베트남에서 쓰이는 한자도 추가되었기에 한중일월 (CJKV) 통합 한자로 부르게 되었다.

112

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

114 \CJKnospace}

115 \showtext

116

117 \bigskip

118 \xeCJKCancelSubCJKBlock{Kana,Hangul}

119 \showtext

120

121 \bigskip

122 \xeCJKRestoreSubCJKBlock{Hangul}

123 \showtext

124

125 \begin{table}[ht]

126 \caption{生僻字测试}

127 \medskip\centering

128 \begin{tabular}{*4{c>{\ttfamily U+}l}}}

129 𠂇 & 3400 & 𠂈 & 3401 & 𠂉 & 3402 & 𠂊 & 3403 & \\\

130 𠂋 & 3404 & 𠂌 & 3405 & 𠂍 & 3406 & 𠂎 & 3407 & \\\

131 𠂏 & 3408 & 𠂐 & 3409 & 𠂑 & 340A & 𠂒 & 340B & \\\

132 𠂓 & 340C & 𠂔 & 340D & 𠂕 & 340E & 𠂖 & 341F & \\\

133 𠂗 & 3410 & 𠂘 & 3411 & 𠂙 & 3412 & 𠂚 & 3413 & \\\

134 𠂛 & 3414 & 𠂜 & 3415 & 𠂝 & 3416 & 𠂞 & 3417 & \\\

135 𠂟 & 3418 & 𠂠 & 3419 & 𠂡 & 341A & 𠂢 & 341B & \\\

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

137 𠂧 & 20000 & 𠂨 & 20001 & 𠂩 & 20002 & 𠂪 & 20003 & \\\

138 𠂫 & 20004 & 𠂬 & 20005 & 𠂭 & 20006 & 𠂮 & 20007 & \\\

139 𠂯 & 20008 & 𠂰 & 20009 & 𠂱 & 2000A & 𠂲 & 2000B & \\\

140 𠂳 & 2000C & 𠂴 & 2000D & 𠂵 & 2000E & 𠂶 & 2000F & \\\

141 𠂷 & 20010 & 𠂸 & 20011 & 𠂹 & 20012 & 𠂺 & 20013 & \\\

```

142 亥 & 20014 & 互 & 20015 & 共 & 20016 & 井 & 20017 \\
143 兂 & 20018 & 丽 & 20019 & 丕 & 2001A & 丙 & 2001B \\
144 巫 & 2001C & 丕 & 2001D & 丽 & 2001E & 束 & 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{一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十。
176 $$x^2+y^2$$
177 一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十\foo
178 \begin{equation}
179 x^2+y^2
180 \end{equation}
181 一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十。}
182 \begin{document}
183 \hsize=30em
184 \parindent=0pt
185 \showtext
186
187 \hrulefill\bigskip
188

```

```

189 \xeCJKsetup{CheckSingle}
190 \showtext
191 \end{document}
192 </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	
/	61, 63, 2138, 2139, 2145
\@italiccorr	2145
\@afterindentfalse	2111
\@afterindenttrue	2111, 2112
\@depth	485
\@height	486
\@ifpackagelater	19
\@minus	445, 494
\@onlypreamble	834, 885, 1862, 1925, 1926, 1927, 1928, 1929
\@pkgextension	128
\@plus	444, 493, 2089
\@width	484
\[768
\]	768
__xeCJK_AfterEndPreamble:n	45, 52, 2147, 2210
__xeCJK_AfterPreamble:n	31, 45, 50, 2044, 2187, 2266
__xeCJK_AtEndPreamble:n	45, 48, 1971
__xeCJK_Boundary_and_CJK:	388, 389, 389, 937, 2258
__xeCJK_CJK_and_CJK:N	453, 454, 454, 702, 703, 711
__xeCJK_CJK_and_FullLeft:N	512, 512
__xeCJK_CJK_and_FullRight:N	524, 524
__xeCJK_FullLeft_and_FullLeft:N	539, 539
__xeCJK_FullLeft_and_FullRight:N	547, 547
__xeCJK_FullRight_and_FullLeft:N	557, 557
__xeCJK_FullRight_and_FullRight:N	566, 566
__xeCJK_UL_CJKpunctsymbol	2273, 2280
__xeCJK_UL_CJKsymbol	2272, 2276
__xeCJK_UL_subclass_patch:nn	881, 925, 925
__xeCJK_UL_subclass_patch_tl	817, 817, 927, 2264
__xeCJK_ULpostpunctchar:n	2243, 2248, 2262
__xeCJK_ULprepunctchar:n	2243, 2243, 2261
__xeCJK_ULroutines:	2243, 2253, 2281
__xeCJK_add_fake_bold:n	1593, 1593, 1641, 1658
__xeCJK_add_fake_slant:n	1593, 1598, 1650
__xeCJK_add_font:nV	1621
__xeCJK_add_font:nn	1576, 1576, 1586
__xeCJK_add_font:nv	1609
__xeCJK_add_font:nx	1666
__xeCJK_add_font_if_new:nn	1576, 1581, 1587, 1660
__xeCJK_add_font_if_new:nv	1613
__xeCJK_add_font_if_new:nx	1668
__xeCJK_add_special_punct:nn	983, 986, 988, 998, 1011
__xeCJK_add_sub_block:	1562, 1708, 1708
__xeCJK_after_FullRight:	473, 506, 506
__xeCJK_app_inter_class_toks:nnc	320, 351
__xeCJK_app_inter_class_toks:nnn	320, 320, 325, 349, 375, 900, 2166
__xeCJK_aux_r	2200, 2201
__xeCJK_before_FullLeft:N	500, 500, 517, 545, 564
__xeCJK_calc_kern:NN	1158, 1167, 1172
__xeCJK_calc_kern:NnNN	1183, 1189, 1205, 1233, 1233
__xeCJK_check_family:V	1546, 1716
__xeCJK_check_family:n	1566, 1566, 1574
__xeCJK_check_for_glue:	389, 391, 397
__xeCJK_check_num_range:nnNN	206, 228, 228, 1936
__xeCJK_checksingle:N	703, 714, 714
__xeCJK_checksingle:NN	714, 717, 726, 730
__xeCJK_checksingle:NNN	714, 754, 758, 763
__xeCJK_checksingle_env:NNN	771, 773, 773
__xeCJK_checksingle_env:NNNNn	773, 783, 786, 791
__xeCJK_checksingle_save_CJKsymbol:N	702, 711, 727, 733, 746, 755, 760, 779, 787, 794
__xeCJK_class_num:n	178, 178, 192, 305, 309, 312

_xeCJK_clear_CJK_toks:	_xeCJK_if_blank_x:n	114
..... 331, 331, 345, 393, 460	_xeCJK_if_blank_x:nF	1690
_xeCJK_clear_inter_class_toks:nn ..	_xeCJK_if_blank_x:nT ...	1438, 2170, 2176
..... 313, 313, 334, 344	_xeCJK_if_blank_x:nTF	
_xeCJK_copy_family:nn 114, 232, 1061, 1406, 1445, 1630, 1675	
_xeCJK_copy_family:xx	_xeCJK_if_blank_x_p:n	114, 230, 722
_xeCJK_copy_inter_class_toks:nnnn .	_xeCJK_if_font_select:n	1628
..... 326, 326, 890,	_xeCJK_if_font_select:nF 1612, 1620, 1667	
891, 904, 909, 910, 2164, 2165, 2173, 2179	_xeCJK_if_font_select:nTF ...	1607, 1628
_xeCJK_declare_char_class:nn	_xeCJK_if_font_select_p:n	
181, 184, 242, 244, 246, 253, 267, 860, 883 1628, 1647, 1648, 1656	
_xeCJK_def_node:nn	_xeCJK_if_it_or_sl:n	1633
..... 581, 581, 592, 593, 594, 595, 596	_xeCJK_if_it_or_sl:nT	1652
_xeCJK_dim_set_max:Nn	_xeCJK_if_it_or_sl:nTF	1610, 1628
..... 1029, 1029, 1120, 1228, 1230	_xeCJK_if_it_or_sl_p:n .	1628, 1646, 1655
_xeCJK_dim_set_min:Nn	_xeCJK_if_last_node:n	576
..... 1029, 1031, 1106, 1114	_xeCJK_if_last_node:nT	371, 650, 675
_xeCJK_erase_CJKsymbol: 819, 819, 899, 936	_xeCJK_if_last_node:nTF ..	370, 576, 2130
_xeCJK_error:n	_xeCJK_if_last_node_p:n	366,
_xeCJK_error:nx ..	400, 405, 407, 422, 424, 425, 431, 432, 576	
_xeCJK_fallback_save_CJKsymbol:N ..	_xeCJK_if_map_font_select:n	1673
..... 1304, 1313, 1319, 1329	_xeCJK_if_map_font_select:nT	1664
_xeCJK_fallback_testsymbol:N	_xeCJK_if_map_font_select:nTF ...	1673
..... 1305, 1316, 1316, 1325	_xeCJK_if_map_font_select_p:n ...	1673
_xeCJK_family_if_exist:x	_xeCJK_if_outer_macro:c	67
_xeCJK_family_if_exist:xF	_xeCJK_if_outer_macro:cF	664, 681
_xeCJK_family_if_exist:xT ...	_xeCJK_if_outer_macro:cTF	57
_xeCJK_family_if_exist:xTF	_xeCJK_if_outer_macro_aux:w	69, 72
.... 1321, 1768, 1787, 1796, 1948, 1951	_xeCJK_if_outer_macro_p:c	
_xeCJK_fix_itcorr: 57, 657, 721, 737, 750	
_xeCJK_flexible_cjkg glue: ...	_xeCJK_if_package_loaded:n	123
2018, 2037	_xeCJK_if_package_loaded:nF .	2065, 2072
_xeCJK_flexible_ecglue:	_xeCJK_if_package_loaded:nT	2202
2017, 2038	_xeCJK_if_package_loaded:nTF .	123, 2077
_xeCJK_get_glyph_bounds:nnN	_xeCJK_if_package_loaded_p:n .	123, 2214
..... 1250, 1253, 1274, 1274	_xeCJK_ignore_spaces:	644, 646, 646
_xeCJK_get_inter_class_toks:nn 311,	_xeCJK_ignorespaces:	
311, 318, 323, 329, 2167, 2169, 2171, 2177 451, 639, 644, 2251, 2277	
_xeCJK_get_kern:NN	_xeCJK_info:nx	129, 135
1158, 1158, 1232	_xeCJK_info:nxx	129, 136, 1779
_xeCJK_get_kern:VN ...	_xeCJK_int_until_do:nn 211, 218, 218, 1939	
543, 551, 561, 570	_xeCJK_int_until_do_aux:wn	
_xeCJK_get_map_font:n 218, 220, 223, 225	
..... 1666, 1668, 1671, 1671, 1675	_xeCJK_inter_class_toks:nnc ...	308, 479
_xeCJK_get_punct_bounds:nN	_xeCJK_inter_class_toks:nnn	
514,	308, 308, 310, 314, 340, 347, 355, 379,	
526, 542, 550, 560, 569, 1034, 1034, 1130	388, 447, 453, 457, 464, 465, 932, 933,	
_xeCJK_get_punct_bounds:nV	934, 2255, 2256, 2257, 2258, 2261, 2262	
508	_xeCJK_inter_class_toks:nnx	
_xeCJK_get_punct_dimen:N 1046, 1248, 1248 308, 317, 322, 328	
_xeCJK_get_sub_features:nn	_xeCJK_itcorr_aux .	2133, 2136, 2138, 2142
..... 1354, 1422, 1430, 1451		
_xeCJK_get_sub_features_aux:n		
..... 1437, 1451, 1459		
_xeCJK_gobble_brace:N		
..... 1434, 1451, 1451, 1456		
_xeCJK_group_begin:		
2196, 2208, 2221		
_xeCJK_group_end:		
2197, 2209, 2222		

_xeCJK_make_node:n	382,	_xeCJK_punct_if_right:NF	1178
	383, 450, 581, 587, 651, 676, 2134, 2277	_xeCJK_punct_if_right:NT	1176
_xeCJK_map_features_id:n	1465, 1579, 1584	_xeCJK_punct_if_right:NTF	1133, 1239, 1241
_xeCJK_map_it_sl:n	1652, 1662, 1662	_xeCJK_punct_if_right_p:N	1133
_xeCJK_math_type:n	1937, 1945	_xeCJK_punct_kern:n	
_xeCJK_msg_new:nn	129, 129, 867, 970,		497, 497, 544, 552, 562, 571
	1333, 1575, 1795, 1821, 1902, 1978, 2082	_xeCJK_punct_nobreak:	563, 994, 996
_xeCJK_new_class:n		_xeCJK_punct_rule:n	
	152, 152, 159, 160, 161, 878		481, 481, 503, 509, 520, 533, 559, 568
_xeCJK_new_sub_key:n		_xeCJK_reset_at_catcode:	2233, 2240
	880, 1398, 1399, 1429, 1464	_xeCJK_restore_CJKsymbol:	
_xeCJK_no_break:	44, 44, 472, 529,		819, 824, 901, 939
	541, 549, 553, 572, 770, 778, 795, 996, 2245	_xeCJK_save_CJK_class:n	171, 177, 918
_xeCJK_parse_features:	1544, 1603, 1603	_xeCJK_save_family_info:	1561, 1696, 1696
_xeCJK_parse_features:n	1603, 1604, 1605	_xeCJK_set_char_catcode:n	209, 210, 213
_xeCJK_pass_features:	1545, 1678, 1678	_xeCJK_set_char_class:nnn	191, 201, 204
_xeCJK_patch:Nnn	2185, 2185, 2195, 2199	_xeCJK_set_char_class_aux:Nn	
_xeCJK_peek_catcode_ignore_spaces:NF			179, 191, 196, 1965
	93, 677	_xeCJK_set_fake:n	1626, 1638, 1638
_xeCJK_peek_catcode_ignore_spaces:NT	91	_xeCJK_set_family_fallback:nnn	
_xeCJK_peek_catcode_ignore_spaces:NTF			1339, 1342, 1345
	82, 83, 92, 94, 652, 745	_xeCJK_set_init:	1513, 1513, 1538
_xeCJK_peek_execute_branches_catcode:		_xeCJK_set_mathfont:	1946, 1946, 1976
	101, 106	_xeCJK_set_monoexspace:	
_xeCJK_peek_false:w	87, 111		1504, 2001, 2002, 2019
_xeCJK_peek_ignore_spaces_execute_branches:		_xeCJK_set_monoscale:	1509, 1990, 1993
	89, 95, 105	_xeCJK_set_others_toks:n	2147, 2155, 2159
_xeCJK_peek_ignore_spaces_execute_branches_aux:		_xeCJK_set_special_punct:nn	
	98, 104		982, 985, 989, 998, 1004
_xeCJK_peek_math:	639, 646, 671	_xeCJK_set_sub_class_toks:nn	879, 886, 886
_xeCJK_peek_tmp:w	99	_xeCJK_space_or_xecglue:	
_xeCJK_peek_true:w	86, 109		371, 609, 614, 621, 653, 660, 678
_xeCJK_pre_inter_class_toks:nnn		_xeCJK_sub_restore_or_cancel:n	
	315, 315,		841, 850, 854
	386, 472, 473, 894, 898, 911, 913, 921, 2168	_xeCJK_sub_special_punct:nn	
_xeCJK_punct_glue:n			984, 987, 990, 998, 1017
	488, 488, 510, 516, 521, 532	_xeCJK_switch_font:nn	
_xeCJK_punct_if_both_left_right:NN	1138		895, 901, 912, 914, 922, 938, 1762, 1775
_xeCJK_punct_if_both_left_right:NNT	1244	_xeCJK_warning:n	132, 1898, 1975
_xeCJK_punct_if_both_left_right:NNTF		_xeCJK_warning:nx	129, 133, 1328, 1815
	1133, 1185	_xeCJK_warning:nxx	134, 1571
_xeCJK_punct_if_both_left_right_p:NN		_xeCJK_zero_glue:	994, 997
	1133	_	25, 870, 2084
_xeCJK_punct_if_long:NTF	527, 1047, 1133		
_xeCJK_punct_if_long_p:N	1133, 1227, 1260		
_xeCJK_punct_if_middle:NT	518, 530		
_xeCJK_punct_if_middle:NTF	1066, 1133		
_xeCJK_punct_if_middle_p:N	1133		
_xeCJK_punct_if_mixedwidth:NT	1103, 1187		
_xeCJK_punct_if_mixedwidth:NTF	1111, 1133		
_xeCJK_punct_if_mixedwidth_p:N	1133, 1197		
_xeCJK_punct_if_right:N	1133		

A

\addCJKfontfeature	1901, 1904
\addCJKfontfeatures	7, 1858, 1863, 1901, 1991
\addto@hook	2270
KaiMingPunct	5
KaiMingPunct+	5
KaiMingPunct-	5

FallBack	7	\c__xeCJK_pkg_extension_tl	125, 128
\AllowBreakBetweenPuncts	976	\c__xeCJK_ps_CCT_tl	1077, 1109
\AtBeginDocument	54	\c__xeCJK_ps_fullwidth_tl	1078, 1182
\AtEndOfPackage	31, 1917	\c__xeCJK_ps_halfwidth_tl	1075, 1100, 1198
\AutoFakeBold	1366, 1481	\c__xeCJK_ps_marginkerning_tl	1174
\AutoFakeSlant	1366, 1481	\c__xeCJK_ps_mixedwidth_tl	1076, 1101
\AutoFallBack	1290	\c__xeCJK_ps_plain_tl	955, 1038, 1163, 1266
B		\c_catcode_other_token	716, 732, 745
\baselineskip	2089	\c_eleven	360, 414
\begin	782, 783	\c_empty_tl	116, 314, 1426, 1437
\bfdefault	1960	\c_four	445, 1282, 2152
\BoldFeatures	1465	\c_group_begin_token	342, 392, 459
\BoldFont	1465	\c_group_end_token	347, 449, 465
\BoldItalicFeatures	1465	\c_math_toggle_token	652, 775, 777, 778, 779
\BoldItalicFont	1465	\c_one	144, 515, 1281, 2152
\BoldSlantedFeatures	1465	\c_space_tl	119
\BoldSlantedFont	1465	\c_space_token	97, 367, 381, 434, 435, 614, 648, 673, 677, 746, 754, 755, 2090
\bool_if:cT	1714, 1744	\c_ten	406, 423
\bool_if:NF	699, 838, 861, 1301, 2012	\c_ten_thousand	44
\bool_if:nF	469	\c_three	444, 1282, 1284, 2150
\bool_if:NT	708, 847, 1310, 1641, 1976, 2031, 2109, 2290	\c_two	443, 1281, 1998, 2007
\bool_if:nT	411, 655, 1226, 1258, 1645, 1653, 2212	\c_two_hundred_fifty_six	2296
\bool_if:NTF	434, 1643, 2039	\c_zero	116, 119, 145, 2141
\bool_if:nTF	230, 357, 365, 399, 403, 420, 429, 719, 735, 743, 748, 765, 1140, 1195, 1635	\c_zero_dim	485, 486, 1041, 1042, 1062, 1070, 1087, 1120, 1230
\bool_new:N	626, 696, 835, 1298, 1362, 1363, 2043, 2055	\c_zero_skip	997, 2089
\bool_set_eq:NN	1530, 1531, 2016	\char_set_catcode:nn	2235
\bool_set_false:c	1408, 1409, 1417, 1420, 1527, 1528	\char_set_catcode_ignore:n	146
\bool_set_false:N	613, 643, 710, 849, 1312, 1372, 1383, 1485, 1494, 2033	\char_set_catcode_letter:N	2238
\bool_set_true:c	1416, 1421	\char_set_catcode_other:N	63
\bool_set_true:N	608, 618, 638, 701, 840, 1303, 1370, 1375, 1381, 1386, 1488, 1497, 2014	\char_set_catcode_other:n	209
C		\char_set_lccode:nn	58, 59, 60, 61, 62
\c__xeCJK_Boundary_class_int	147, 151	\char_value_catcode:n	2236
\c__xeCJK_CJK_class_int	147, 148	\CheckSingle	688
\c__xeCJK_Default_class_int	147, 147	\CJK@hundredmillion	2300
\c__xeCJK_FullLeft_class_int	147, 149	\CJK@ifundefined	2228, 2229
\c__xeCJK_FullRight_class_int	147, 150	\CJK@tenthousand	2299
\c__xeCJK_HalfLeft_class_int	159	\CJK@UnicodeEnc	2292, 2298
\c__xeCJK_HalfRight_class_int	159	\CJKaddEncHook	2293
\c__xeCJK_math_tl	1931, 1932, 1948, 1949, 1957, 1959, 1966	\CJKkecglue	367, 370, 375, 386, 409, 427, 598, 602, 603, 609, 619, 620, 621, 666, 683, 932, 2017, 2255, 2257
\c__xeCJK_NormalSpace_class_int	159	xCJKkecglue	4
\c__xeCJK_Others_class_int	2147, 2161, 2184	\CJKfamily	6, 1807, 1807, 1840, 1855, 1919, 1920, 1921, 1922, 2223
		\CJKfamilydefault	7, 1787, 1788, 1878, 1912, 1915, 1916, 1919, 1951, 1952
		\CJKfixedspacing	6, 2010, 2010, 2044
		\CJKflexiblespacing	6, 1989, 2010, 2029
		\CJKfontspec	6, 1834, 1844, 1896
		\CJKglue	401, 434, 454, 527, 597, 597, 2018

\CJKmath	1932	\clist_set:Nx	1461, 1540, 1868, 1881
\CJKnospace	631, 636, 641	\cs_end:	74
\CJKnumber	2055	\cs_generate_variant:Nn	
\CJKpunctsymbol	504, 537, 555, 574, 2105, 2106, 2245, 2250, 2273, 2279		176, 198, 310, 325, 499, 812, 813, 814, 815, 816, 949, 1025, 1026, 1027, 1028, 1130, 1149, 1232, 1397, 1429, 1565, 1574, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1761, 1806
\CJKrmdefault		\cs_gset_eq:cc	1758
..	7, 1827, 1908, 1912, 1912, 1915, 1920	\cs_gset_eq:cN	1771
\CJKsetecglue	2125, 2125, 2126	\cs_gset_eq:NN	821, 822, 826
\CJKsfdefault	7, 1828, 1911, 1912, 1913, 1921	\cs_gset_protected_nopar:cpx	
\CJKspace	627, 630, 636, 636		1547, 1553, 1724, 1733
\CJKsymbol	349, 395, 454, 464, 770, 778, 795, 821, 822, 826, 827, 932, 933, 1304, 1305, 1313, 2105, 2105, 2255, 2256, 2257, 2272, 2274	\cs_if_exist:cF	667, 684
\CJKsymbol_Block	821, 826	\cs_if_exist:cTF	1809
\CJKttdefault	7, 1829, 1912, 1914, 1922, 2048, 2051, 2052	\cs_if_exist:NT	2268, 2284
\clist_clear:c	1521	\cs_if_exist_use:cF	1059, 1765
\clist_clear:N		\cs_if_free:cT	1847
....	1433, 1441, 1523, 1524, 1887, 1888	\cs_if_free:cTF	1798
\clist_gclear_new:c	155, 168	\cs_if_free:NT	28, 2228
\clist_gput_right:cx	186	\cs_if_free:NTF	2229
\clist_gput_right:Nn	917	\cs_meaning:c	70
\clist_gput_right:Nx	173, 803, 1401	\cs_new_eq:NN	1774, 1901, 1909, 1945, 2126, 2138, 2208, 2209
\clist_gremove_all:Nn	809	\cs_new_nopar:Npn	72, 95, 106, 129, 130, 131, 132, 133, 134, 135, 136, 178, 311, 1465, 1671, 2105, 2106
\clist_gremove_duplicates:N	804	\cs_new_protected:Npn	
\clist_gset:Nn	1860		48, 50, 52, 83, 91, 93, 218, 223
\clist_if_empty:cF	1682	\cs_new_protected_nopar:Npn	44, 104, 152, 171, 184, 204, 228, 308, 313, 315, 320, 326, 331, 389, 397, 454, 481, 488, 497, 500, 506, 512, 524, 539, 547, 557, 566, 581, 587, 646, 671, 714, 730, 763, 773, 791, 819, 824, 854, 886, 925, 997, 1004, 1011, 1017, 1029, 1031, 1034, 1158, 1172, 1233, 1248, 1274, 1316, 1345, 1399, 1430, 1451, 1459, 1513, 1535, 1566, 1576, 1581, 1593, 1598, 1603, 1605, 1638, 1662, 1678, 1696, 1708, 1751, 1763, 1775, 1946, 1993, 2002, 2127, 2159, 2185, 2243, 2248, 2253
\clist_if_empty:Nf	905	\cs_new_protected_nopar:Npx	1840
\clist_if_in:NnF	1891	\cs_set:Npn	602, 619
\clist_if_in:NnT	1872	\cs_set_eq:NN	85, 99, 209, 210, 609, 614, 621, 639, 644, 702, 703, 711, 994, 996, 1304, 1305, 1313, 1923, 2017, 2018, 2111, 2145, 2200, 2220, 2221, 2222, 2223, 2229, 2272, 2273
\clist_if_in:NnTF	793	\cs_set_nopar:cpn	2294
\clist_map_break:	1883	\cs_set_nopar:Npn	2286
\clist_map_function:nN	177	\cs_set_nopar:Npx	2233
\clist_map_inline:cn	1963		
\clist_map_inline:Nn	167, 333, 467, 888, 907, 947, 999, 1150, 1264, 1525, 1699, 1711, 1870, 1889, 2162		
\clist_map_inline:nn			
...	198, 336, 338, 350, 353, 377, 455, 476, 478, 919, 929, 1040, 1618, 1961, 2259		
\clist_map_inline:xn	187, 808, 856, 1351		
\clist_new:N			
...	41, 42, 43, 169, 818, 944, 1398, 1858		
\clist_put_left:cx	1355, 1595, 1600		
\clist_put_left:Nv	1894		
\clist_put_right:Nn	1892		
\clist_put_right:Nx	1413, 1684, 1689, 1691		
\clist_remove_all:Nn	1701, 1702, 1721, 1874		
\clist_remove_duplicates:N	1710		
\clist_set:cx	1448		
\clist_set:Nn	162, 945, 998		

\cs_set_protected_nopar:Npn	108, 667, 684, 1551,
....	2139, 2201, 2274, 2279, 2293, 2295
\cs_to_str:N	1838
D	
\DeclareSymbolFont	1957
\defaultCJKfontfeatures	6, 1858, 1859, 1861, 1862
\dim_add:Nc	1026, 1177, 1179
\dim_add:Nn	1027, 1105, 1112, 1113, 1255
\dim_compare:nNnT	1190, 1206
\dim_compare:nNnTF	1070, 1087
\dim_compare_p:nNn	416
\dim_const:cn	584
\dim_eval:c	484, 492, 498, 499
\dim_eval:n	493, 494, 499, 1257
\dim_gset:cn	584
\dim_gset:Nn	2006
\dim_if_exist:cTF	583
\dim_max:nn	1030, 1210
\dim_min:cc	1026, 1212, 1217
\dim_min:nn	1028, 1032, 1097
\dim_new:N	36, 37, 38, 625, 1986
\dim_set:Nc	1026, 1054, 1055, 1166, 1191, 1201
\dim_set:Nn	1026, 1030, 1032, 1057, 1068,
	1084, 1089, 1096, 1202, 1208, 1235, 1276
\dim_set_eq:NN	1117, 1252
\dim_to_fp:n	1998
\dim_use:c	578, 589, 590
\dim_use:N	1123, 1125, 1269
\dim_zero:N	1049, 1050, 1051, 1162
\document	55, 56
E	
xeCJKactive	3
\else:	75, 100, 110, 118, 120, 579
\EmboldenFactor	1366
\end	785, 786
\etex_fontcharwd:D	1257
\etex_lastnodetype:D	360, 406, 414, 423, 515
\etex_protected:D	1919, 1920, 1921, 1922
\exp_after:wN	99, 101, 109, 111, 225, 724, 1437
\exp_args:Nc	154, 1630
\exp_args:NNc	723, 724
\exp_args:Nnnx	2166, 2168
\exp_args:No	723, 1435, 1453
\exp_args:Nv	1855
\exp_args:Nx	1341, 2034
\exp_last_unbraced:Nf	69
\exp_last_unbraced:NNo	1455
\exp_last_unbraced:NNV	1771
\exp_not:N	108, 667, 684, 1551,
	1552, 1556, 1557, 1558, 1728, 1729,
	1731, 1732, 1736, 1737, 1738, 1840, 1896
\exp_not:n	318, 323, 1549, 1554, 1734, 2235
\exp_not:V	960
F	
\f@series	1762
\f@shape	1762
\f@size	1762, 1998, 2007
\Fallback	1464
\fi:	75, 102, 112, 120, 121, 225, 579
\file_input:n	2239
\font_glyph_if_exist:NnTF	1318
\fontencoding	1556, 1736
\fontfamily	1557, 1737, 1996, 2005
\fontspec_set_family:Nnn	1549
\fp_gset:Nn	1997
\fp_new:N	1364, 1365, 1985
\fp_set:Nn	1376, 1387, 1489, 1498
\fp_set_eq:NN	1532, 1533
\fp_use:N	1510, 1596, 1601, 1991
G	
\g__xeCJK_after_end_preamble_hook_tl	47, 53, 56
\g__xeCJK_after_preamble_hook_tl	46, 51, 54
\g__xeCJK_at_end_preamble_hook_tl	45, 49, 55
\g__xeCJK_AutoFakeBold_bool	1362, 1370, 1372, 1375, 1530
\g__xeCJK_AutoFakeSlant_bool	1363, 1381, 1383, 1386, 1531
\g__xeCJK_base_class_clist	162, 162, 167, 467, 888
\g__xeCJK_CJK_class_clist	162, 169, 173, 333, 2162
\g__xeCJK_CJK_class_prop	162, 170, 174, 208
\g__xeCJK_CJK_subclass_clist	817, 818, 905, 907, 917, 1699, 1870
\g__xeCJK_class_seq	152, 156, 158, 2150, 2154
\g__xeCJK_default_features_clist	1543, 1858, 1860
\g__xeCJK_EmboldenFactor_fp	1364, 1376, 1390, 1491, 1532
\g__xeCJK_exspace_dim	1985, 1986, 2006, 2023, 2024
\g__xeCJK_family_name_prop	1551, 1570,
	1693, 1694, 1728, 1731, 1756, 1801, 1955
\g__xeCJK_family_options_prop	1693, 1695, 1705, 1719, 1722, 1756, 1886

\g__xeCJK_family_prop	1336, 1349, 1568, <u>1693</u> , 1693, 1704, 1717, 1718, 1756, 1865, 1869, 1875, 1877, 1974
\g__xeCJK_family_spec_tl .	1850, 1852, 1853
\g__xeCJK_features_id_prop <u>1465</u> , 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1518, 1604, 1680
\g__xeCJK_fontspec_int 1516, 1842, 1843, 1849, 1851
\g__xeCJK_indent_bool	2055, 2059, 2109
\g__xeCJK_inline_env_clist 793, 800, 803, 804, 809
\g__xeCJK_math_bool	1933, 1976
\g__xeCJK_monoscale_fp 1510, <u>1985</u> , 1985, 1991, 1997
\g__xeCJK_number_bool	2058, 2290
\g__xeCJK_punct_width_tl ..	991, 1061, 1063
\g__xeCJK_punctstyle_clist 944, 944, 945, 947, 960, 974, 1264
\g__xeCJK_SlantFactor_fp 1365, 1387, 1391, 1500, 1533
\g__xeCJK_special_ps_clist <u>998</u> , 998, 999, 1150
\g__xeCJK_sub_key_clist <u>1398</u> , 1398, 1401, 1525
\g__xeCJK_unknown_family_seq 1812, 1814, 1820
\g_fontspec_encoding_tl 1556, 1736, 1957, 1959
\group_align_safe_begin:	88
\group_align_safe_end:	86, 87
\group_begin:	55, 57, 1347, 1537, 1698, 1867, 1995, 2004, 2208
\group_end:	55, 66, 1360, 1563, 1706, 1896, 1999, 2008, 2209
H	
CheckSingle	4
I	
MiddlePunct	5
MiddlePunct+	5
MiddlePunct-	5
\if_catcode:w	108
\if_cs_exist:w	74
\if_dim:w	578
\if_int_compare:w	116, 119, 221
\if_meaning:w	97
\IfBooleanF	202
\IfBooleanT	182, 832, 842, 851
\IfBooleanTF	2047, 2050
\IfNoValueF	2232
\IfNoValueTF	236, 237, 1838
\indentfirst	2055
\InlineEnv	798
\InlineEnv+	798
\InlineEnv-	798
\int_compare:nNnF	515, 2149
\int_compare:nNnTF	2141
\int_compare_p:nNn	360, 406, 414, 423
\int_const:Nn	147, 148, 149, 150, 151
\int_eval:n	1285, 2296
\int_gdecr:N	1843
\int_gincr:N	1516, 1849
\int_if_exist:cF	876
\int_if_exist:cTF	858
\int_incr:N	215, 1942
\int_max:nn	237
\int_min:nn	236
\int_new:N	33, 34, 35, 1842, 2184
\int_set:Nn	207, 232, 236, 237, 1937, 1938, 2161
\int_set_eq:NN	233
\int_step_inline:nnnn	2152
\int_to_hexadecimal:n	1337
\int_use:N	1851
\ItalicFeatures	<u>1465</u>
\ItalicFont	<u>1465</u>
J	
CJKecglue	4
CJKglue	3
CJKmath	3
CJKnumber	3
CJKspace	3
K	
\KaiMingPunct	<u>976</u>
\keys_define:nn	137, 597, 598, 627, 688, 798, 949, 950, 961, 976, 1290, 1366, 1402, 1475, 1481, 1933, 2056
\keys_define:nx	<u>949</u> , 959
\keys_set:nn	2087, 2116
\keys_set_known:nnN	1397
\keys_set_known:nxN	<u>1397</u> , 1542
L	
\l__keys_module_tl	2084
\l__xeCJK_add_font_prop 1517, 1578, 1583, 1688
\l__xeCJK_AutoFakeBold_bool 1485, 1488, 1530, 1641, 1654
\l__xeCJK_AutoFakeSlant_bool 1494, 1497, 1531, 1643
\l__xeCJK_checksingle_bool 696, 699, 701, 708, 710

\l__xeCJK_CJKfontspec_id_tl	\l__xeCJK_sub_family_tl
. 1846, 1847, 1853, 1855	1713, 1716, 1718, 1722, 1724, 1732, 1733, 1746
\l__xeCJK_current_coar_tl	\l__xeCJK_sub_fontfeat_clist
1767, 1771 1433, 1441, 1448, 1461
\l__xeCJK_ecglue_dim	\l__xeCJK_sub_fontname_tl
416, 603, 620, 625	1438, 1440, 1444, 1445, 1446, 1447, 1449, 1462
\l__xeCJK_EmboldenFactor_fp 1489, 1532, 1596	\l__xeCJK_sub_key_clist
\l__xeCJK_fallback_bool 1413, 1524, 1710, 1711
. 1298, 1301, 1303, 1310, 1312	\l__xeCJK_tmpa_clist
\l__xeCJK_family_fb_tl 41, 41, 1719, 1721, 1722, 1868,
1348, 1349, 1353, 1354, 1355, 1356, 1357, 1358	1872, 1874, 1881, 1882, 1891, 1894, 1896
\l__xeCJK_family_name_tl	\l__xeCJK_tmpa_dim
1801	36, 36, 1049, 1054, 1097, 1105, 1106, 1112, 1113,
\l__xeCJK_family_tl	1114, 1117, 1123, 1162, 1166, 1169,
1410, 1539, 1546, 1547, 1552, 1553, 1704, 1705, 1713,	1177, 1179, 1183, 1189, 1190, 1191,
1717, 1719, 1726, 1729, 1838, 1839, 1840	1201, 1202, 1205, 1206, 1208, 1228,
\l__xeCJK_fixed_spacing_bool	1230, 1250, 1251, 1252, 1253, 1254, 1255
. 2012, 2014, 2031, 2033, 2043	\l__xeCJK_tmpa_int
\l__xeCJK_fixed_xecglue_bool	33, 33, 206, 211, 213, 214, 215, 1936, 1939, 1941, 1942
2016, 2039	\l__xeCJK_tmpa_tl
\l__xeCJK_flexible_punctstyle_tl 2015, 2036	39, 39, 1432, 1434, 1435, 1437, 1440,
\l__xeCJK_font_coar_tl	1444, 1568, 1570, 1571, 1717, 1718,
. 1033, 1268, 1762, 1762, 1765, 1767	1729, 1732, 1737, 1757, 1783, 1789,
\l__xeCJK_fontname_bf_tl	1869, 1879, 1896, 1949, 1952, 1953, 1955
1621	\l__xeCJK_tmppb_clist 41, 42, 1886, 1887, 1889
\l__xeCJK_fontname_tl	\l__xeCJK_tmppb_dim
1349, 1350, 1446, 1447, 1541, 1550, 1704	36, 37, 1050, 1055, 1091, 1097, 1125, 1252, 1255, 1257, 1269
\l__xeCJK_fontoptions_clist	\l__xeCJK_tmppb_int 33, 34, 206, 211, 1936, 1939
. 1540, 1701, 1702, 1705	\l__xeCJK_tmppb_tl
\l__xeCJK_fontspec_family_tl 39, 40, 1875, 1878, 1879, 1882
. 1549, 1552, 1557	\l__xeCJK_tmppc_clist 41, 43, 1888, 1892, 1894
\l__xeCJK_lastpunct_tl	\l__xeCJK_tmppc_dim
502, 508, 509, 510, 536, 543, 544, 551,	36, 38, 1051, 1057, 1068, 1070, 1071,
552, 554, 559, 561, 562, 568, 570, 571, 573	1084, 1087, 1089, 1091, 1096, 1105,
\l__xeCJK_math_family_tl	1106, 1112, 1113, 1114, 1117, 1120, 1126
1955, 1958, 1960	\l__xeCJK_tmppc_int
\l__xeCJK_monoscale_tl 33, 35, 207, 208, 214, 1937, 1941
. 1505, 1510, 1515, 1690, 1691	\l__xeCJK_xecglue_bool
\l__xeCJK_pass_features_clist 359, 413, 608, 613, 618, 626, 2016
. 1523, 1543, 1550, 1684, 1689, 1691	\l__keys_choice_tl
\l__xeCJK_peek_search_token	954
82, 85, 108	\l__keys_key_tl
\l__xeCJK_punct_coar_tl	2078, 2079
. 484, 492, 493, 494, 498, 1033, 1033, 1036, 1041, 1042,	\l__keys_value_tl
1045, 1054, 1056, 1082, 1092, 1122,	968, 1376, 1387, 1489, 1498
1124, 1126, 1135, 1142, 1143, 1144,	\l__peek_token
1145, 1160, 1169, 1177, 1179, 1191,	97, 108, 767, 768
1201, 1213, 1214, 1218, 1219, 1238,	\l__tmpa_int
1240, 1242, 1243, 1245, 1251, 1254, 1256	1938, 1941
\l__xeCJK_punctstyle_tl	\l__xeCJK_family_tl
. 954, 955, 1033, 1038, 1073, 1098, 1163, 1174, 1182, 1198, 2015 1321, 1323, 1335, 1336, 1762,
\l__xeCJK_SlantFactor_fp	1768, 1769, 1780, 1783, 1784, 1785,
1498, 1533, 1601	1788, 1789, 1810, 1865, 1869, 1875, 1886
\l__xeCJK_space_bool	SlantFactor
434, 638, 643	4
\l__xeCJK_space_glue_tl	AllowBreakBetweenPuncts
361, 415, 441	5
\l__xeCJK_sub_cancel_bool	\LongPunct
. 835, 838, 840, 847, 849, 861	976

M	
\makeCJKactive	140, <u>144</u> , 144
\makeCJKinactive	141, <u>144</u> , 145, 2196, 2199, 2201, 2205, 2220, 2245, 2250, 2287
\mathalpha	1966
\mathchar@type	1945
EmboldenFactor	4
\mddefault	1958
\MiddlePunct	<u>976</u>
\Mono	<u>1481</u>
\msg_critical:nn	10, 20
\msg_error:nn	30, 130
\msg_error:nnx	131
\msg_info:nnx	135
\msg_info:nnxx	136
\msg_new:nnn	3, 11, 23, 129
\msg_redirect_module:nnn	2063, 2064, 2070, 2071
\msg_warning:nn	132
\msg_warning:nnx	133
\msg_warning:nnxx	134
N	
indentfirst	3
\newCJKfontfamily	6, <u>1834</u> , 1836
\NewDocumentCommand	144, 145, 179, 196, 199, 240, 302, 636, 641, 697, 706, 829, 836, 845, 873, 993, 995, 1131, 1288, 1299, 1308, 1339, 1807, 1834, 1836, 1844, 1859, 1863, 1907, 1910, 1930, 1934, 1987, 2001, 2010, 2029, 2045, 2114, 2119, 2121, 2123, 2124, 2125, 2230
\newXeTeXintercharclass	154
InlineEnv	4
InlineEnv+	4
InlineEnv-	4
\normalfont	1919, 1923, 1973
\normalspacedchars	10, <u>302</u> , 302, 307
O	
LongPunct	5
LongPunct+	5
LongPunct-	5
Mono	6
P	
\p@	2007
\par	767
\PassOptionsToPackage	2066, 2073, 2079
\pdfstringdefPreHook	2215, 2218
\pdfutex_strcmp:D	116, 119
\peek_after:Nw	89, 105
\peek_catcode:NTF	381, 648, 673, 716, 732, 777
\peek_catcode_remove:NTF	775
\peek_meaning_remove:NTF	782, 785
\Pifont	2204
\prg_do_nothing:	85, 822, 1100, 2220, 2221, 2222
\prg_new_conditional:cpnn	1152
\prg_new_conditional:Npnn	67, 114, 123, 576, 1133, 1138, 1149, 1628, 1633, 1673
\prg_new_protected_conditional:Npnn	1796
\prg_return_false:	75, 120, 126, 579, 1136, 1147, 1155, 1631, 1636, 1676, 1799
\prg_return_true:	75, 117, 120, 126, 579, 1136, 1147, 1155, 1631, 1636, 1676, 1803
\ProcessKeysOptions	2107
\prop_clear:c	1007
\prop_clear:N	1517
\prop_get:Nn	1466, 1589
\prop_get:NnNF	1590
\prop_get:NnNT	1568, 1728, 1757
\prop_get:NnNTF	1806
\prop_get:Nv	1336
\prop_get:NVN	1869
\prop_get:NVNF	1349, 1886
\prop_get:NVNT	1717, 1719, 1955
\prop_get:NxNF	1801, 1875, 1877
\prop_gpop:NnNT	1570
\prop_gput:Nnn	176
\prop_gput:NnV	1551, 1731, 1757
\prop_gput:NvV	1704, 1705, 1718, 1722
\prop_gput:Nvx	174
\prop_if_empty:NTF	1974
\prop_if_in:cnTF	1154
\prop_if_in:NnF	1591
\prop_if_in:NvTF	208, 1865
\prop_map_inline:Nn	1473, 1518, 1604, 1680, 1688
\prop_new:c	1002
\prop_new:N	170, 1467, 1693, 1694, 1695
\prop_put:cn	1009, 1015
\prop_put:Nnn	1468, 1469, 1470, 1471, 1472, 1588
\prop_put:Nxx	1578
\prop_put_if_new:Nnn	1592
\prop_put_if_new:Nxx	1583
\prop_remove:cn	1022
\PunctStyle	<u>950</u>
\punctstyle	31, <u>2123</u> , 2123, 2124
\PunctWidth	<u>976</u>
Q	
\q_no_value	1953
\q_stop	70, 72

R	
<code>\r</code>	2200, 2201
<code>\RenewDocumentCommand</code>	2204
<code>\RequirePackage</code>	21, 22, 2108, 2297
<code>\reset@font</code>	1923
<code>\reverse_if:N</code>	221
<code>\rmfamily</code>	1920
S	
<code>\scan_stop:</code>	486, 796, 2129
<code>\selectfont</code>	1558, 1738, 1996, 2005
<code>\seq_count:N</code>	2150
<code>\seq_gput_right:Nv</code>	156
<code>\seq_gput_right:Nx</code>	1814
<code>\seq_if_in:NnF</code>	2154
<code>\seq_if_in:NxF</code>	1812
<code>\seq_new:N</code>	158, 1820
<code>\setCJKfallbackfamilyfont</code> .	7, 8, 1339, 1339
<code>\setCJKfamilyfont</code>	6, 1831, 1834, 1834
<code>\setCJKmainfont</code> 5, 1827, 1907, 1907, 1909, 1925	
<code>\setCJKmathfont</code>	7, 1926, 1930, 1930
<code>\setCJKmonoexspace</code>	6, 2001, 2001
<code>\setCJKmonofont</code> ...	5, 1829, 1928, 2045, 2045
<code>\setCJKmonoscale</code>	6, 1987, 1987
<code>\setCJKromanfont</code>	1909, 1929
<code>\setCJKsansfont</code> ...	5, 1828, 1907, 1910, 1927
<code>\SetSymbolFont</code>	1959
<code>\settowidth</code>	603, 620
<code>\sffamily</code>	1921
<code>\shapedefault</code>	1958, 1960
<code>\skip_horizontal:N</code>	997, 2023
<code>\skip_horizontal:n</code>	490, 2024, 2089
<code>\skip_if_eq_p:nn</code>	361, 415
<code>\SlantedFeatures</code>	1465
<code>\SlantedFont</code>	1465
<code>\SlantFactor</code>	1366
<code>\SplitArgument</code>	197
<code>\str_case_x:nnn</code>	1279, 1825
<code>\str_if_eq_x:nnF</code>	
.....	189, 374, 385, 1266, 1777, 1781
<code>\str_if_eq_x:nnT</code>	1616
<code>\str_if_eq_x:nnTF</code>	
.....	892, 931, 1056, 1414, 1640, 1672
<code>\str_if_eq_x_p:nn</code> 470, 1261, 1635, 1647, 1648	
T	
<code>\tex_afterassignment:D</code>	98
<code>\tex_char:D</code>	2296
<code>\tex_font:D</code>	443,
444, 445, 1257, 1318, 1771, 1998, 2007	
<code>\tex_fontdimen:D</code> ..	443, 444, 445, 1998, 2007
<code>\tex_ignorespaces:D</code>	
..	56, 465, 1818, 1856, 1899, 2117, 2246
<code>\tex_kern:D</code>	498, 589, 590
<code>\tex_lastkern:D</code>	578
<code>\tex_lastskip:D</code>	361, 415, 416
<code>\tex_penalty:D</code>	44
<code>\tex_the:D</code>	312
<code>\tex_unkern:D</code>	651, 676, 2132
<code>\tex_unskip:D</code>	364, 419
<code>\tex_vrule:D</code>	483
<code>\textcentereddot</code>	2191
<code>\textellipsis</code>	2191
<code>\textemdash</code>	2191
<code>\textperiodcentered</code>	2191
<code>\textquotedblleft</code>	2192
<code>\textquotedblright</code>	2192
<code>\textquoteleft</code>	2192
<code>\textquoteright</code>	2192
<code>\tipaencoding</code>	2199
<code>\tl_case:Nnn</code>	1073, 1098
<code>\tl_clear:c</code>	1520
<code>\tl_clear:N</code>	1350, 1505, 1515
<code>\tl_const:cn</code>	948
<code>\tl_const:Nn</code>	1932
<code>\tl_const:Nx</code>	128
<code>\tl_gput_right:Nn</code>	49, 51, 53, 2218
<code>\tl_gset:cn</code>	1132, 1289
<code>\tl_gset:cV</code> 1041, 1042, 1126, 1169, 1251, 1254	
<code>\tl_gset:cx</code>	1122, 1124, 1256, 1268, 1853
<code>\tl_gset:Nx</code>	536, 1850
<code>\tl_if_eq:NnF</code>	1163
<code>\tl_if_eq:NNT</code>	955
<code>\tl_if_eq:NNTF</code>	1038, 1174, 1182
<code>\tl_if_eq_p:NN</code>	1198
<code>\tl_if_exist:cF</code>	1036, 1045, 1160
<code>\tl_if_exist:cTF</code>	125, 1135, 1165
<code>\tl_if_exist:NF</code>	1912, 1913, 1914, 1915
<code>\tl_if_exist_p:c</code>	1142, 1143, 1144, 1145
<code>\tl_if_exist_p:N</code>	2215
<code>\tl_if_head_eq_charcode:nNTF</code>	1435
<code>\tl_if_head_is_group:nT</code>	1453
<code>\tl_if_in:nnTF</code>	1341
<code>\tl_if_single_token_p:n</code>	723
<code>\tl_map_function:nN</code>	63
<code>\tl_map_inline:cn</code>	1008, 1014
<code>\tl_map_inline:nn</code> .	62, 304, 1025, 1755, 2189
<code>\tl_map_inline:xn</code>	1019, 1132
<code>\tl_new:c</code>	77, 78, 79, 80, 1001, 1916
<code>\tl_new:N</code>	39, 40, 45, 46, 47, 817
<code>\tl_put_left:Nn</code>	55, 2186
<code>\tl_put_right:cx</code>	1013
<code>\tl_put_right:Nn</code>	56, 927, 1323,
1353, 1919, 1920, 1921, 1922, 2044, 2186	
<code>\tl_put_right:Nx</code>	1410, 1784

\tl_remove_all:cn	1021		
\tl_remove_all:Nn	1780		
\tl_replace_all:Nnx	1447		
\tl_set:cn	2227		
\tl_set:cx	1006, 1449		
\tl_set:Nn .. 86, 87, 441, 1033, 1455, 1762,			
1912, 1913, 1914, 1915, 2292, 2299, 2300			
\tl_set:Nx	502, 554, 573,		
954, 1348, 1432, 1440, 1444, 1446,			
1462, 1510, 1539, 1541, 1713, 1767,			
1783, 1810, 1838, 1846, 1879, 1949, 1952			
\tl_set_eq:NN	1953, 2015		
\tl_to_lowercase:n	64		
\tl_use:c .. 178, 493, 494, 862, 1082, 1092,			
1238, 1240, 1242, 1243, 1245, 1672, 1685			
\token_get_arg_spec:c	722		
\token_get_arg_spec:N	814		
\token_if_cs_p:c	658, 740, 752		
\token_if_cs_p:N	812		
\token_if_eq_meaning_p:NN	767, 768		
\token_if_math_toggle:cTF	666, 683		
\token_if_math_toggle:NTF	816		
\token_if_math_toggle_p:c	739, 751		
\token_if_math_toggle_p:N	815		
\token_if_other_p:N	724		
\token_if_space_p:c	738, 743		
\token_if_space_p:N	813		
\token_new:Nn	82		
\token_to_str:N	25,		
870, 1827, 1828, 1829, 1831, 1904, 1982			
\TrimSpaces	179, 874		
\ttdefault	1996, 2005		
\ttfamily	1922		
U			
\UL@hook	2268, 2270		
PunctStyle	4		
PunctWidth	5		
\Unicode	2295		
\use:c	462, 1769, 1802, 1938, 2298		
\use:x	1895		
\use_none:n	210, 220, 223, 226, 2223		
\use_none:nnnn	1770		
\usefont	2205		
AutoFakeBold	4, 5		
AutoFakeSlant	4, 5		
AutoFallBack	4		
V			
\verbatim@font	2044		
X			
\xCJKecglue	598		
\xe@alloc@intercharclass	2150, 2152		
\xCJK@setfont	1774		
\xCJK_select_font:			
..... 343, 394, 461, 1046, 1324,			
1762, 1763, 1774, 1793, 2276, 2280, 2287			
\xCJK_set_family:xcc	1356, 1746		
\xCJK_set_family:xnn			
. 1343, 1535, 1535, 1565, 1835, 1839,			
1852, 1908, 1911, 1931, 2048, 2051, 2052			
\xCJKactive	137		
\xCJKallowbreakbetweenpuncts			
..... 956, 979, 993, 993			
\xCJKCancelSubCJKBLOCK	10, 835, 836		
\xCJKcaption	2228, 2230		
\xCJKDeclareCharClass	10, 179, 179		
\xCJKDeclareSubCharClass	831, 873, 873, 885		
\xCJKDeclareSubCJKBLOCK	9, 829, 829, 834, 870		
\xCJKdisablechecksingle	692, 696, 706		
\xCJKdisablefallback	1294, 1298, 1308		
\xCJKenablechecksingle	691, 696, 697		
\xCJKenablefallback	1293, 1298, 1299		
\xCJKmainfont	1982		
\xCJKnobreakbetweenpuncts .. 980, 993, 995			
\xCJKplainchr	2123, 2124		
\xCJKResetPunctClass			
10, 182, 202, 240, 240, 301, 832, 842, 851			
\xCJKRestoreSubCJKBLOCK	10, 835, 845		
\xCJKsetcharclass	199, 199		
\xCJKsetecglue	2126		
\xCJKsetemboldenfactor	2119, 2119		
\xCJKsetkern	10, 1288, 1288		
\xCJKsetmathcode	1934, 1934, 1965		
\xCJKsetslantfactor	2119, 2121		
\xCJKsetup	3, 2020,		
2034, 2114, 2114, 2120, 2122, 2123, 2125			
\xCJKsetwidth	10, 1131, 1131		
\XeTeX@CJkfntef@hook	2284, 2286		
\xetex_if_engine:F	10		
\XeTeXcharclass	214, 305		
\XeTeXcharglyph	1285		
\XeTeXdefaultencoding	2232, 2241		
\XeTeXglyphbounds	25, 28, 1278		
\XeTeXinterchartokenstate .. 144, 145, 2141			
\XeTeXinterchartoks	309, 312		
\XeTeXmathcode	1941		