

xeCJK 宏包

ctex.org*

2012/05/06 v3.0.4

1 简介

xeCJK 是一个 \LaTeX 宏包, 用于排版 CJK 文字, 包括字体选择和标点控制等。主要特点:

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 的原始作者是孙文昌, 目前由 ctex-kit¹ 项目组维护, 主要维护者是刘海洋² 和李清³。

2 使用方法

xeCJK 使用了 \LaTeX 的一些最新特性, 需要 \LaTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 还依赖 fontspec⁴ 宏包来调用系统字体。

xeCJK 提供了将在 2.1 节介绍的一些 $\langle key \rangle = \langle var \rangle$ 形式的宏包选项, 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。

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。

\backslash xeCJKsetup	\backslash xeCJKsetup { $\langle key1 \rangle = \langle var1 \rangle$, $\langle key2 \rangle = \langle var2 \rangle$, ...}
-------------------------	--

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}
```

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

¹<http://code.google.com/p/ctex-kit/>

²leoliu.pku@gmail.com

³sobenlee@gmail.com

⁴<http://mirrors.ctan.org/macros/latex/contrib/fontspec>

2.1 宏包选项

下列选项使用加粗的形式来表示 `xeCJK` 的默认设置。

<hr/> <code>xeCJKactive</code> <hr/>	<code>xeCJKactive = <i><true false></i></code> 打开/关闭对中文的特殊处理。
<hr/> <code>CJKnumber</code> ★ <hr/>	<code>CJKnumber = <i><true false></i></code> 是否使用 <code>CJKnumb</code> 宏包。请注意,为了与 <code>CJKnumb</code> 兼容, <code>xeCJK</code> 修改了它的一些内部命令,所以这个选项需要在调用 <code>xeCJK</code> 时设置,而不能通过 <code>\xeCJKsetup</code> 来设置它。
<hr/> <code>indentfirst</code> ★ <hr/>	<code>indentfirst = <i><true false></i></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 <code>xeCJK</code> 时设置,不能通过 <code>\xeCJKsetup</code> 来设置它。 <code>xeCJK</code> 宏包中只有 <code>indentfirst</code> 和 <code>CJKnumber</code> 这两个选项,不能通过 <code>\xeCJKsetup</code> 来设置。
<hr/> <code>CJKspace</code> <hr/>	<code>CJKspace = <i><true false></i></code> 缺省状态下, <code>xeCJK</code> 会忽略 <code>CJK</code> 文字之间的空格,使用这一选项来保留它们之间的空格。
<hr/> <code>CJKmath</code> ★ <hr/>	<code>CJKmath = <i><true false></i></code> 是否支持在数学环境中直接输入 <code>CJK</code> 字符。使用这个选项后,可以直接在数学环境中输出 <code>CJK</code> 字符。
<hr/> <code>CJKglue</code> <hr/>	<code>CJKglue = {\hspace{0pt plus .08\baselineskip}}</code> 设置 <code>CJK</code> 文字之间插入的 <code>glue</code> ,上边是 <code>xeCJK</code> 的默认值。一般来说,除非有特殊需要(例如,改变文字间距等),否则不需要设置这个选项,使用默认值即可。如果要设置这个选项,为了行末的对齐,设置的 <code>glue</code> 最好有一定的弹性。
<hr/> <code>CJKecglue</code> <hr/>	<code>CJKecglue = { }</code> 设置 <code>CJK</code> 文字与西文、 <code>CJK</code> 文字与行内数学公式之间的间距,默认值是一个空格。使用这个选项设置的 <code>glue</code> 最好也要用一定的弹性。请注意,这里设置的 <code>glue</code> 只影响 <code>xeCJK</code> 根据需要自动添加的空白,源文件中直接输入的 <code>CJK</code> 文字与西文之间的空格不受影响(直接输出)。有时候 <code>xeCJK</code> 可能不能正确地调整间距,需要手动加空格。
<hr/> <code>xCJKecglue</code> <hr/>	<code>xCJKecglue = <i><true false glue></i></code> 缺省状态下, <code>xeCJK</code> 不对源文件中直接输入的 <code>CJK</code> 文字与西文之间的空格进行调整,如果需要调整,请使用这个选项。如果使用这个选项,将使用 <code>CJKecglue</code> 替换源文件中直接输入的 <code>CJK</code> 文字与西文之间的空格。
<hr/> <code>CheckSingle</code> <hr/>	<code>CheckSingle = <i><true false></i></code> 是否避免单个 <code>CJK</code> 文字单独占一个段落的最后一行。
<hr/> <code>InlineEnv</code> <code>InlineEnv+</code> <code>InlineEnv-</code> <hr/>	<code>CheckSingle = {env1, env2, env3, ...}</code> 在使用 <code>CheckSingle</code> 选项的时候, <code>xeCJK</code> 会将 <code>CJK</code> 文字后接着的 <code>L^AT_EX</code> 环境的开始 <code>\begin{...}</code> 和结束 <code>\end{...}</code> 视为断行的地方,如果有某些特殊的 <code>L^AT_EX</code> 环境没有造成断行,可以使用这个选项来声明它,以便 <code>CheckSingle</code> 能正确识别。

<hr/> AutoFallBack <hr/>	AutoFallBack = $\langle true false \rangle$ 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 2.2 节中介绍。
<hr/> AutoFakeBold ☆ <hr/>	AutoFakeBold = $\langle true false 数字 \rangle$ 全局设定当没有声明对应的粗体时,是否使用 伪粗体 ;当输入的是数字时,将使用伪粗体,并将使用输入的数字作为伪粗体的默认粗细程度。
<hr/> AutoFakeSlant ☆ <hr/>	AutoFakeSlant = $\langle true false 数字 \rangle$ 全局设定当没有声明对应的斜体时,是否使用 伪斜体 ;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。
<hr/> EmboldenFactor ☆ <hr/>	EmboldenFactor = $\langle 数字 4 \rangle$ 设置伪粗体的默认粗细程度。
<hr/> SlantFactor ☆ <hr/>	SlantFactor = $\langle 数字 0.167 \rangle$ 设置伪斜体的粗细程度,范围是 $-0.999 \sim 0.999$ 。
<hr/> PunctStyle <hr/>	PunctStyle = $\langle quanjiao banjiao kaiming hangmobanjiangiao CCT plain \rangle$ 设置标点处理格式,有效值分别为 <div> <div>quanjiao</div> <div>banjiao</div> <div>kaiming</div> <div>hangmobanjiangiao</div> <div>CCT</div> <div>plain</div> </div> <div> <div>全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;</div> <div>半角式: 所有标点占半个汉字宽度;</div> <div>开明式: 句末点号用全角,其他半角;</div> <div>行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</div> <div>CCT 格式;</div> <div>原样(不调整标点间距)。</div> </div>
<hr/> KaiMingPunct ☆ <hr/>	KaiMingPunct = $\langle . . ? ! \rangle$
KaiMingPunct+ ☆	设置开明(kaiming)标点处理格式时的句末点号,KaiMingPunct 后带的 + 与 - 分别表示从已有的开明句末点号中增加或减少标点。
KaiMingPunct- ☆	
<hr/> LongPunct ☆ <hr/>	LongPunct = $\langle \text{—} \cdots \text{—} \rangle$
LongPunct+ ☆	设置长标点,例如破折号“—”与省略号“……”,允许在长标点前后断行,但是禁止在它们之间断行。
LongPunct- ☆	
<hr/> MiddlePunct ☆ <hr/>	MiddlePunct = $\langle \cdot \cdot \cdot \rangle$
MiddlePunct+ ☆	设置居中显示的标点,例如间隔号“·”。对于在 CJK 文字之间的居中标点,xeCJK 会根据不同的标点处理格式,调整居中标点与前后文字之间的空白,保证其确实居中。对于行末出现的居中标点,允许在其后面断行,但禁止在它前面断行。
MiddlePunct- ☆	
<hr/> PunctWidth ☆ <hr/>	PunctWidth = $\{ \langle \text{dimen} \rangle \}$ 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 2.5 节的 \xeCJKsetwidth。
<hr/> AllowBreakBetweenPuncts <hr/>	AllowBreakBetweenPuncts = $\langle true false \rangle$ 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。

2.2 CJK 字体族声明方式

<hr/> \setCJKmainfont ★ <hr/>	\setCJKmainfont [<i>{font features}</i>] <i>{font name}</i>
	设置正文罗马族的 CJK 字体,最后两个参数的意义请参考 <code>fontspec</code> 宏包。影响 <code>\rmfamily</code> 和 <code>\textrm</code> 的字体。需要说明的是 <code>xeCJK</code> 修改了 <code>AutoFakeBold</code> 和 <code>AutoFakeSlant</code> 选项,以便配合全局伪粗体和伪斜体的设定。
<hr/> AutoFakeBold AutoFakeSlant <hr/>	AutoFakeBold = <i>{true false 数字}</i> AutoFakeSlant = <i>{true false 数字}</i>
	局部设置当前字体族的伪粗和伪斜属性。如果没有在局部给出这些选项,将使用全局设定。
<hr/> \setCJKsansfont ★ <hr/>	\setCJKsansfont [<i>{font features}</i>] <i>{font name}</i>
	设置正文无衬线族的 CJK 字体,影响 <code>\sffamily</code> 和 <code>\textsf</code> 的字体。
<hr/> \setCJKmonofont ★ <hr/>	\setCJKmonofont [<i>{font features}</i>] <i>{font name}</i>
	设置正文等宽族的 CJK 字体,影响 <code>\ttfamily</code> 和 <code>\texttt</code> 的字体。为了有利于等宽字体的代码对齐等情形, <code>xeCJK</code> 在 <i>{font features}</i> 里增加了 <code>Mono</code> 这个选项。
<hr/> Mono <hr/>	Mono = <i>{Scale Exspace}</i>
	有效值仅为 <code>Scale</code> 或 <code>Exspace</code> 。分别表示使用放缩的方式或改变间距的方式,让 CJK 字符的宽度等于西文字体中两个空格的宽度。
<hr/> \CJKfixedspacing \CJKflexiblespacing <hr/>	\CJKfixedspacing \CJKflexiblespacing
	\CJKfixedspacing 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。 \CJKflexiblespacing 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 \CJKfixedspacing 的效果。
<hr/> \setCJKmonoscale \setCJKmonoexspace <hr/>	\setCJKmonoscale \setCJKmonoexspace
	\setCJKmonoscale 和 \setCJKmonoexspace 命令分别计算并设置当前字号下西文等宽字体与 CJK 字体的放缩比例和字距,以符合上面所说的 <code>Mono=Scale</code> 与 <code>Mono=Exspace</code> 选项的意义。在字号改变时,应该使用两个命令之一,重新计算相关参数;但注意两个命令不能同时使用。
<hr/> \setCJKfamilyfont ★ <hr/>	\setCJKfamilyfont <i>{family}</i> [<i>{font features}</i>] <i>{font name}</i>
	声明新的 CJK 字体族 <i>{family}</i> 并指定字体。
<hr/> \CJKfamily <hr/>	\CJKfamily <i>{family}</i>
	用于在文档中切换 CJK 字体族, <i>{family}</i> 必须预先声明。

<hr/> <hr/> <code>\newCJKfontfamily</code> ★	<code>\newCJKfontfamily [<i></i>] \<font-switch> [<i></i>] {}</code> <p>声明新的 CJK 字体族 <i></i> 并指定字体,并定义 <i>\<font-switch></i>,在文档中可以使用它来切换 CJK 字体族。可以不必指定 <i></i>,这时候 <i></i> 将等于 <i><font-switch></i>。事实上,<code>\newCJKfontfamily</code> 是 <code>\setCJKfamilyfont</code> 和 <code>\CJKfamily</code> 的合并。例如</p> <pre> \newCJKfontfamily[song]\songti{SimSun} </pre> <p>等价于</p> <pre> \setCJKfamilyfont{song}{SimSun} \newcommand\songti{\CJKfamily{song}} </pre>
<hr/> <hr/> <code>\CJKfontspec</code>	<code>\CJKfontspec [<i></i>] {}</code> <p>在文档中随机定义新的 CJK 字体族,并马上使用它。</p>
<hr/> <hr/> <code>\defaultCJKfontfeatures</code> ★	<code>\defaultCJKfontfeatures {}</code> <p>全局设置 CJK 字体族的默认选项。例如,使用</p> <pre> \defaultCJKfontfeatures{Scale=0.962216} </pre> <p>可以将全部 CJK 字体缩小为 0.962216。</p>
<hr/> <hr/> <code>\addCJKfontfeatures</code>	<code>\addCJKfontfeatures {}</code> <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/> <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> 。
<hr/> <hr/> <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> 。
<hr/> <hr/> <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> 。
<hr/> <hr/> <code>\CJKfamilydefault</code>	<p>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/> <code>\setCJKmathfont</code> ★	<code>\setCJKmathfont [<i></i>] {}</code> <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>

`\setCJKfallbackfamilyfont` ★ `\setCJKfallbackfamilyfont {<family>} [] {}`

设置 CJK 字体族 *<family>* 的备用字体。例如,使用

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

可以将 SimSun-ExtB 作为 SimSun 的备用字体。

FallBack `FallBack = {[]}{}}`

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

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

如果 FallBack 的值为空,将设置的是备用字体。例如,

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

等价于

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

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

`\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。

2.3 CJK 分区字体设置

众所周知,CJK 文字数量极其庞大,单一的字體不可能涵盖所有的 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。

<code>\xeCJKDeclareSubCJKBlock</code> ★	<code>\xeCJKDeclareSubCJKBlock {<block>} {<block range>}</code>
---	---

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。使用

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

就声明了 SPUA 和 Ext-B 这两个个子分区。同时在 2.2 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 2.2 介绍的 FallBack 类似。可以通过它们来设置字体。例如, 使用

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

就设置了文档的主字体是 SimSun, SPUA 分区的字体是 SunmanPUA, 而 Ext-B 分区的字体是 SimSun-ExtB。

`\xeCJKDeclareSubCJKBlock` 应该在声明所有的 CJK 字体族之前使用。如果有某个 CJK 字体族没有设置 *<block>* 选项, 将使用 `\CJKfamilydefault` 的 *<block>* 选项作为该 CJK 字体族的 *<block>* 选项。如果希望在使用某 CJK 字体族时, 不在 CJK 主分区与 *<block>* 之间切换字体, 可以使用 *<block>=** 选项。

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

在文档中取消对 CJK 分区的声明。

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

在文档中恢复对 CJK 分区的声明。

2.4 设置 CJK 字符范围

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

<class range> 的格式和 2.3 节的 *<block range>* 相同。*<class>* 的有效值见源代码(第 4.1 节)。`xeCJK` 已经支持 Unicode 中所有 CJK 文字和标点。一般来说, 不要轻易改变字符类别。

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

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

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

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

2.5 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。

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

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

```
\xeCJKsetkern{ : }{"}{0.3em}
```

3 兼容性

为了与 `CJKnumb` 和 `CJKulem` 宏包兼容,`xeCJK` 使用并重新定义了 `CJK` 宏包的部分宏命令,如 `\CJKfamily`、`\CJKsymbol` 和 `\CJKpunctsymbol` 等。需要指出,`xeCJK` 不需要 `CJK` 的支持,并且 `xeCJK` 自动禁止在它之后载入 `CJK` 宏包。

可以在 `xeCJK` 之后载入 `CJKfntef` 宏包,以实现汉字加点等。

`xeCJK` 依赖 `XYTeX` 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。

4 `xeCJK` 代码实现

```
1 <{*package>
2 \RequirePackage{xparse}
3 \RequirePackage{l3keys2e}
4 \ProvidesExplPackage {xeCJK} {2012/05/06} {3.0.4}
5 {package for typesetting CJK scripts with XeLaTeX}

\xeCJK_msg_new:nn
  \xeCJK_error:n
  \xeCJK_error:nx
\xeCJK_warning:nx
  \xeCJK_info:nx
  \xeCJK_info:nxx

6 \cs_new_nopar:Npn \xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
7 \cs_new_nopar:Npn \xeCJK_error:n { \msg_error:nn { xeCJK } }
8 \cs_new_nopar:Npn \xeCJK_error:nx { \msg_error:nnx { xeCJK } }
9 \cs_new_nopar:Npn \xeCJK_warning:n { \msg_warning:nn { xeCJK } }
10 \cs_new_nopar:Npn \xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
11 \cs_new_nopar:Npn \xeCJK_info:nx { \msg_info:nnx { xeCJK } }
12 \cs_new_nopar:Npn \xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }

(End definition for \xeCJK_msg_new:nn and others.)
xeCJK 必须使用 XYTeX 引擎的支持。

13 \msg_new:nnn {xeCJK} { Require-XeTeX }
14 {
15   The~xeCJK~package~requires~XeTeX~to~function.\\\
16   You~must~change~your~typesetting~engine~to~"xelatex" \
17   instead~of~"plain"~"latex"~or~"pdflatex"~or~"lualatex".
18 }
19 \xetex_if_engine:F { \msg_critical:nn { xeCJK } { Require-XeTeX } }
    \XeTeXglyphbouds 可以得到一个字符的左右边距,用于标点压缩。
20 \xeCJK_msg_new:nn { XeTeX-too-old }
21 {
22   \token_to_str:N \XeTeXglyphbouds\c_space_tl is~not~defined.\\\
23   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
24 }
25 \cs_if_free:NT \XeTeXglyphbouds
26 {
27   \xeCJK_error:n { XeTeX-too-old }
28   \AtBeginDocument { \punctstyle { plain } }
29 }
    抑制 BOM。
30 \char_set_catcode_ignore:n { "FEFF }
```

xeCJKActive xeCJK 功能开关。

```

31 \keys_define:nn { xeCJK / options }
32 {
33   xeCJKActive .choice:,
34   xeCJKActive / true .code:n = { \makexeCJKActive },
35   xeCJKActive / false .code:n = { \makexeCJKInactive },
36   xeCJKActive .default:n = { true },
37 }

```

(End definition for xeCJKActive. This function is documented on page 2.)

\makexeCJKActive
\makexeCJKInactive

```

38 \NewDocumentCommand \makexeCJKActive { } { \XeTeXinterchartokenstate = \c_one }
39 \NewDocumentCommand \makexeCJKInactive { } { \XeTeXinterchartokenstate = \c_zero }

```

(End definition for \makexeCJKActive and \makexeCJKInactive.)

4.1 字符类别设定

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

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

\xeCJK_Default_class
\xeCJK_CJK_class
\xeCJK_FullLeft_class
\xeCJK_FullRight_class
\xeCJK_Boundary_class

这五类是 Xe_{La}TeX 预定义的类别。

```

40 \int_const:Nn \xeCJK_Default_class { 0 }
41 \int_const:Nn \xeCJK_CJK_class { 1 }
42 \int_const:Nn \xeCJK_FullLeft_class { 2 }
43 \int_const:Nn \xeCJK_FullRight_class { 3 }
44 \int_const:Nn \xeCJK_Boundary_class { 255 }

```

(End definition for \xeCJK_Default_class and others.)

\xeCJK_new_class:n
\g_xeCJK_class_seq

新建一个字符类别。

```

45 \cs_new_nopar:Nn \xeCJK_new_class:n
46 {
47   \exp_after:wN \newXeTeXintercharclass \cs:w xeCJK_#1_class\cs_end:
48   \clist_gclear_new:c { g_xeCJK_#1_range_clist }
49   \seq_gput_right:Nv \g_xeCJK_class_seq { xeCJK_#1_class }
50 }
51 \seq_new:N \g_xeCJK_class_seq

```

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

\xeCJK_HalfLeft_class
\xeCJK_HalfRight_class
\xeCJK_NormalSpace_class

新增西文半角左/右标点和前后原始间距的符号类。

```

52 \xeCJK_new_class:n { HalfLeft }
53 \xeCJK_new_class:n { HalfRight }
54 \xeCJK_new_class:n { NormalSpace }

```

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

```

\g_xeCJK_base_class_clist
\g_xeCJK_CJK_class_clist 55 \clist_set:Nn \g_xeCJK_base_class_clist
\g_xeCJK_CJK_class_prop 56 {
57   Default, CJK, FullLeft, FullRight, Boundary,
58   HalfLeft, HalfRight, NormalSpace
59 }
60 \clist_map_inline:Nn \g_xeCJK_base_class_clist
61 { \clist_gclear_new:c { g_xeCJK_#1_range_clist } }
62 \clist_new:N \g_xeCJK_CJK_class_clist
63 \prop_new:N \g_xeCJK_CJK_class_prop
64 \cs_new_nopar:Nn \xeCJK_save_CJK_class:n
65 {
66   \clist_gput_right:Nx \g_xeCJK_CJK_class_clist {#1}
67   \prop_gput:Nvx \g_xeCJK_CJK_class_prop { xeCJK_#1_class } {#1}
68 }
69 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
70 \clist_map_inline:nn { CJK, FullLeft, FullRight } { \xeCJK_save_CJK_class:n {#1} }
(End definition for \g_xeCJK_base_class_clist, \g_xeCJK_CJK_class_clist, and \g_xeCJK_CJK_class_prop.)

```

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

```

71 \cs_new_nopar:Nn \xeCJK_class_num:n { \use:c { xeCJK_#1_class } }
(End definition for \xeCJK_class_num:n.)

```

```

\tl_if_blank_p:x
\tl_if_blank:xTF 72 \prg_set_conditional:Nnn \tl_if_blank:x { p , T , F , TF }
73 {
74   \if_int_compare:w \pdfstrcmp:D { \c_empty_tl } {#1} = \c_zero
75   \prg_return_true:
76   \else:
77     \if_int_compare:w \pdfstrcmp:D { \c_space_tl } {#1} = \c_zero
78     \prg_return_true: \else: \prg_return_false: \fi:
79   \fi:
80 }
(End definition for \tl_if_blank:x.)

```

\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。

```

\XeCJKsetcharclass_aux:Nn 81 \NewDocumentCommand \xeCJKDeclareCharClass { > { \TrimSpaces } m m }
82 {
83   \clist_gput_right:cx { g_xeCJK_#1_range_clist } {#2}
84   \clist_map_inline:xn {#2}
85   {
86     \str_if_eq:xxF {##1} { -> }
87     {
88       \xeCJKsetcharclass_aux:Nn \xeCJKsetcharclass {##1}
89       { \xeCJK_class_num:n {#1} }
90     }
91   }
92 }
93 \NewDocumentCommand \xeCJKsetcharclass_aux:Nn
94 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
95 \cs_generate_variant:Nn \clist_map_inline:nn { x }
(End definition for \xeCJKDeclareCharClass. This function is documented on page 7.)

```

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

```
96 \NewDocumentCommand \xeCJKsetcharclass { m m m }
97 {
98   \xeCJK_check_num_range:nnNN {#1} {#2} \l_tmpa_int \l_tmpb_int
99   \int_set:Nn \l_tmpc_int {#3}
100   \prop_if_in:NVTF \g_xeCJK_CJK_class_prop \l_tmpc_int
101   { \cs_set_eq:NN \xeCJK_set_catcode:n \char_set_catcode_other:n }
102   { \cs_set_eq:NN \xeCJK_set_catcode:n \use_none:n }
103   \loop
104     \xeCJK_set_catcode:n \l_tmpa_int
105     \XeTeXcharclass \l_tmpa_int = \l_tmpc_int
106     \if_int_compare:w \l_tmpa_int < \l_tmpb_int
107       \int_incr:N \l_tmpa_int
108   \repeat
109 }
(End definition for \xeCJKsetcharclass.)
```

\xeCJK_check_num_range:nnNN

```
110 \cs_new_nopar:Nn \xeCJK_check_num_range:nnNN
111 {
112   \bool_if:nTF { \tl_if_blank_p:x {#1} || \tl_if_blank_p:x {#2} }
113   {
114     \int_set:Nn #3 { \tl_if_blank:xTF {#1} {#2} {#1} }
115     \int_set_eq:NN #3 #4
116   }
117   {
118     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
119     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
120   }
121 }
(End definition for \xeCJK_check_num_range:nnNN.)
```

\xeCJKResetPunctClass 用于重置标点符号所属的字符类。

```
122 \NewDocumentCommand \xeCJKResetPunctClass { }
123 {
124   \xeCJKDeclareCharClass { HalfLeft }
125   { "28 , "2D , "5B , "60 , "7B }
126   \xeCJKDeclareCharClass { HalfRight }
127   { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , }
128   \xeCJKDeclareCharClass { FullLeft }
129   {
130     "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
131     "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
132     "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
133     "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6 ,
134   }
135   \xeCJKDeclareCharClass { FullRight }
136   {
137     "00B7 , "2019 , "201D , "2014 , "2025 , "2026 , "2030 , "2500 , "3001 ,
138     "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 , "3015 ,
139     "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 , "3047 ,
140     "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C , "309D ,
141     "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
142     "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE , "FE50 ,
143     "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C , "FE5E ,
144     "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B , "FF1F ,
145     "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 , "FF69 ,
146     "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E , "FF9F ,
```

```

147     }
148 }
(End definition for \xeCJKResetPunctClass. This function is documented on page 7.)
设置 CJK 文字的范围。
149 \xeCJKDeclareCharClass { CJK }
150 {
    • Hangul Jamo (谚文字母)
151     "1100 -> "11FF ,

    • CJK Radicals Supplement (中日韩部首补充)
152     "2E80 -> "2EFF ,

    • Kangxi Radicals (康熙字典部首)
153     "2F00 -> "2FDF ,

    • Ideographic Description Characters (表意文字描述符)
154     "2FF0 -> "2FFF ,

    • CJK Symbols and Punctuation (中日韩符号和标点)
155     "3000 -> "303F ,

    • Hiragana (日文平假名)
156     "3040 -> "309F ,

    • Katakana (日文片假名)
157     "30A0 -> "30FF ,

    • Bopomofo (注音字母)
158     "3100 -> "312F ,

    • Hangul Compatibility Jamo (谚文兼容字母)
159     "3130 -> "318F ,

    • Kanbun (象形字注释标志)
160     "3190 -> "319F ,

    • Bopomofo Extended (注音字母扩展)
161     "31A0 -> "31BF ,

    • CJK Strokes (中日韩笔画)
162     "31C0 -> "31EF ,

    • Katakana Phonetic Extensions (日文片假名语音扩展)
163     "31F0 -> "31FF ,

    • Enclosed CJK Letters and Months (带圈中日韩字母和月份)
164     "3200 -> "32FF ,

    • CJK Compatibility (中日韩兼容)
165     "3300 -> "33FF ,

```

- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
166 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
167 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
168 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
169 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
170 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
171 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
172 "AC00 -> "D7AF ,
- Kana Supplement (日文假名增补)
173 "B000 -> "B0FF ,
- Hangul Jamo Extended-B (谚文扩展 B)
174 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
175 "F900 -> "FAFF ,
- CJK Compatibility Forms (中日韩兼容形式)
176 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)
177 "FF00 -> "FFEF ,
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)
178 "20000 -> "2A6DF ,
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)
179 "2A700 -> "2B73F ,
- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)
180 "2B740 -> "2B81F ,
- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)
181 "2F800 -> "2FA1F ,
- 182 }
- 重置标点符号的字符类。
- 183 \xeCJKResetPunctClass

`\normalspacedchars` 声明前后不加间距的字符。

```

184 \NewDocumentCommand \normalspacedchars { m }
185 {
186   \tl_map_inline:nn {#1}
187   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
188 }
189 \normalspacedchars{/}

```

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

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

`\xeCJK_inter_class_toks:nnc`

```

190 \cs_new_nopar:Nn \xeCJK_inter_class_toks:nnn
191 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = {#3} }
192 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc }

```

(End definition for \xeCJK_inter_class_toks:nnn and \xeCJK_inter_class_toks:nnc.)

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

```

193 \cs_new_nopar:Nn \xeCJK_clear_inter_class_toks:nn
194 {
195   \XeTeXinterchartoks
196   \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = { \c_empty_tl }
197 }

```

(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。

```

198 \cs_new_nopar:Nn \xeCJK_pre_inter_class_toks:nnn
199 {
200   \use:x
201   {
202     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
203     {
204       \exp_not:n {#3} \tex_the:D \XeTeXinterchartoks
205       \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2}
206     }
207   }
208 }

```

(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`

```

209 \cs_new_nopar:Nn \xeCJK_app_inter_class_toks:nnn
210 {
211   \use:x
212   {
213     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
214     {
215       \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1}
216       \xeCJK_class_num:n {#2} \exp_not:n {#3}
217     }
218   }
219 }
220 \cs_generate_variant:Nn \xeCJK_app_inter_class_toks:nnn { nnc }

```

(End definition for \xeCJK_app_inter_class_toks:nnn and \xeCJK_app_inter_class_toks:nnc.)

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

```

221 \cs_new_nopar:Nn \xeCJK_get_inter_class_toks:nn
222 {
223   \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2}
224 }

```

(End definition for \xeCJK_get_inter_class_toks:nn.)

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

```

225 \cs_new_nopar:Nn \xeCJK_copy_inter_class_toks:nnnn
226 {
227   \use:x
228   {
229     \xeCJK_inter_class_toks:nnn {#1} {#2}
230     { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
231   }
232 }

```

(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。

```

233 \cs_new_nopar:Nn \xeCJK_clear_CJK_toks:
234 {
235   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
236   { \xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
237 }

```

(End definition for \xeCJK_clear_CJK_toks:.)

4.2 字符输出规则

```

238 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
239 {
240   \clist_map_inline:nn { CJK, FullLeft, FullRight }
241   {
242     \xeCJK_inter_class_toks:nnn {#1} {##1}
243     {
244       \c_group_begin_token
245       \xeCJK_select_font:
246       \xeCJK_clear_inter_class_toks:nn {#1} {##1}
247       \xeCJK_clear_CJK_toks:
248     }
249   }
250   \xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
251   \clist_map_inline:nn { FullLeft, FullRight }
252   { \xeCJK_app_inter_class_toks:nnc {#1} {##1} { xeCJK_CJK_and_##1:N } }
253   \xeCJK_inter_class_toks:nnn { CJK } {#1} { \c_group_end_token }
254   \xeCJK_inter_class_toks:nnn { FullLeft } {#1} { \nobreak \c_group_end_token }
255   \xeCJK_inter_class_toks:nnn { FullRight } {#1}
256   { \xeCJK_after_FullRight: \c_group_end_token }
257 }

258 \xeCJK_pre_inter_class_toks:nnn { Default } { CJK } { \CJKecglue }

259 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
260 {
261   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
262   {
263     \int_compare:nNnTF \tex_lastkern:D = \c_one { \CJKecglue }
264     { \int_compare:nNnT \tex_lastkern:D = \c_two \xeCJK_space_or_xecglue: }
265   }
266   \str_if_eq:xxF {#1} { NormalSpace }
267   { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
268 }

269 \clist_map_inline:nn { Default, HalfRight, NormalSpace }
270 {

```

```

271 \xeCJK_inter_class_toks:nnn {#1} { Boundary }
272 {
273   \peek_catcode:NTF \c_space_token
274   { { \xeCJK_default_space_kern: } } { { \xeCJK_default_kern: } }
275 }
276 }
277 \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }

\xeCJK_Boundary_and_CJK:
278 \cs_new_nopar:Nn \xeCJK_Boundary_and_CJK:
279 {
280 {
281   \bool_if:nTF
282   {
283     \int_compare_p:nNn \tex_lastkern:D = \c_one ||
284     \int_compare_p:nNn \tex_lastkern:D = \c_two
285   }
286   { \CJKglue }
287   {
288     \bool_if:nTF
289     {
290       \int_compare_p:nNn \tex_lastkern:D = \c_three ||
291       \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
292       ( \int_compare_p:nNn \etex_lastnodetype:D = \c_twelve &&
293         \g_xeCJK_itcorr_bool )
294     }
295     { \CJKecglue }
296     {
297       \bool_if:nT
298       {
299         \l_xeCJK_xecglue_bool &&
300         \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
301         ( \int_compare_p:nNn
302           \tex_lastskip:D = { \tex_fontdimen:D \c_two \tex_font:D } ||
303           \int_compare_p:nNn \tex_lastskip:D = \l_xeCJK_ecglue_dim )
304       }
305       {
306         \skip_set_eq:NN \l_tmpa_skip \tex_lastskip:D
307         \tex_unskip:D
308         \bool_if:nTF
309         {
310           \int_compare_p:nNn \tex_lastkern:D = \c_four ||
311           \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
312           \int_compare_p:nNn \tex_lastkern:D = \c_three ||
313           \g_xeCJK_itcorr_bool
314         }
315         { \CJKecglue }
316         {
317           \bool_if:nT
318           {
319             ! ( \int_compare_p:nNn \tex_lastkern:D = \c_one ) ||
320             ! ( \int_compare_p:nNn \tex_lastkern:D = \c_two ) ||
321             \l_xeCJK_space_bool
322           }
323           { \skip_horizontal:N \l_tmpa_skip }
324         }
325       }
326     }
327   }
328 }

```

```

327     }
328     \bool_gset_false:N \g_xeCJK_itcorr_bool
329   }
330   \c_group_begin_token
331   \xeCJK_clear_CJK_toks:
332   \xeCJK_select_font:
333   \CJKsymbol
334 }
(End definition for \xeCJK_Boundary_and_CJK:.)
335 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
336 {
337   \c_group_end_token
338   { \xeCJK_CJK_kern: }
339   \xeCJK_ignorespaces:
340 }
341 \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }

```

\xeCJK_CJK_and_CJK:N

```

342 \cs_new_nopar:Nn \xeCJK_CJK_and_CJK:N { \CJKglue \CJKsymbol {#1} }
(End definition for \xeCJK_CJK_and_CJK:N.)
343 \clist_map_inline:nn { FullLeft, FullRight }
344 {
345   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
346   {
347     \c_group_begin_token
348     \xeCJK_clear_CJK_toks:
349     \xeCJK_select_font:
350     \use:c { xeCJK_CJK_and_#1:N }
351   }
352   \xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
353   \xeCJK_inter_class_toks:nnn {#1} { Boundary }
354   { \c_group_end_token \tex_ignorespaces:D }
355   \clist_map_inline:nn { CJK, Boundary }
356   {
357     \exp_args:Nnnx \xeCJK_pre_inter_class_toks:nnn {#1} {##1}
358     {
359       \str_if_eq:xxTF {#1} { FullLeft }
360       { \exp_not:N \nobreak } { \exp_not:N \xeCJK_after_FullRight: }
361     }
362   }
363 }

```

\xeCJK_punct_rule:n 用于抹去标点符号的左/右空白。

```

364 \cs_new_nopar:Nn \xeCJK_punct_rule:n
365 {
366   \tex_vrule:D
367   \@width \dim_eval:c { \l_xeCJK_punct_coortl/rule/#1 }
368   \@depth \c_zero_dim
369   \@height \c_zero_dim \scan_stop:
370 }
(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。

```

371 \cs_new_nopar:Nn \xeCJK_punct_glue:n
372 {
373   \skip_horizontal:n
374   {

```

```

375         \dim_eval:c { \l_xeCJK_punct_coortl/glue/#1 }
376         \@plus \dim_eval:n { \tl_use:c { \l_xeCJK_punct_coortl/glue/#1 } /4 }
377         \@minus \dim_eval:n { \tl_use:c { \l_xeCJK_punct_coortl/glue/#1 } /2 }
378     }
379 }

```

(End definition for \xeCJK_punct_glue:n.)

\xeCJK_punct_kern:n 相邻两个标点之间的间距。

```

380 \cs_new_nopar:Nn \xeCJK_punct_kern:n
381 { \tex_kern:D \dim_eval:c { \l_xeCJK_punct_coortl/kern/#1 } }

```

(End definition for \xeCJK_punct_kern:n.)

\dim_eval:c

```

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

```

(End definition for \dim_eval:c.)

\xeCJK_before_FullLeft:N

```

383 \cs_new_nopar:Nn \xeCJK_before_FullLeft:N
384 {
385     \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
386     \xeCJK_punct_rule:n { 1/#1 }
387     \CJKpunctsymbol {#1}
388 }

```

(End definition for \xeCJK_before_FullLeft:N.)

\xeCJK_after_FullRight:

```

389 \cs_new_nopar:Nn \xeCJK_after_FullRight:
390 {
391     \xeCJK_get_punct_bounds:nV { r } \l_xeCJK_lastpunct_tl
392     \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
393     \xeCJK_punct_glue:n { r/\l_xeCJK_lastpunct_tl }
394 }

```

(End definition for \xeCJK_after_FullRight:.)

```

395 \clist_map_inline:nn { CJK, FullLeft, FullRight }
396 {
397     \clist_map_inline:nn { FullLeft, FullRight }
398     { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
399 }

```

\xeCJK_CJK_and_FullLeft:N

```

400 \cs_new_nopar:Nn \xeCJK_CJK_and_FullLeft:N
401 {
402     \xeCJK_get_punct_bounds:nN { 1 } {#1}
403     \int_compare:nNf \etex_lastnodetype:D = \c_one
404     { \xeCJK_punct_glue:n { 1/#1 } }
405     \xeCJK_before_FullLeft:N {#1}
406     \xeCJK_punct_if_middle:NT {#1}
407     {
408         \xeCJK_punct_rule:n { m/#1 }
409         \xeCJK_punct_glue:n { 1/#1 }
410     }
411 }

```

(End definition for \xeCJK_CJK_and_FullLeft:N.)

\xeCJK_CJK_and_FullRight:N

```
412 \cs_new_nopar:Nn \xeCJK_CJK_and_FullRight:N
413 {
414   \xeCJK_get_punct_bounds:nN { r } {#1}
415   \xeCJK_punct_if_long:NTF {#1} \CJKglue
416   {
417     \nobreak
418     \xeCJK_punct_if_middle:NT {#1}
419     {
420       \xeCJK_punct_glue:n { r/#1 }
421       \xeCJK_punct_rule:n { m/#1 }
422     }
423   }
424   \tl_gset:Nx \l_xeCJK_lastpunct_tl {#1}
425   \CJKpunctsymbol {#1}
426 }
(End definition for \xeCJK_CJK_and_FullRight:N)
```

\xeCJK_FullLeft_and_FullLeft:N

```
427 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullLeft:N
428 {
429   \nobreak
430   \xeCJK_get_punct_bounds:nN { l } {#1}
431   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
432   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
433   \xeCJK_before_FullLeft:N {#1}
434 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N)
```

\xeCJK_FullLeft_and_FullRight:N

```
435 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullRight:N
436 {
437   \nobreak
438   \xeCJK_get_punct_bounds:nN { r } {#1}
439   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
440   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
441   \nobreak
442   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
443   \CJKpunctsymbol {#1}
444 }
(End definition for \xeCJK_FullLeft_and_FullRight:N)
```

\xeCJK_FullRight_and_FullLeft:N

```
445 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullLeft:N
446 {
447   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
448   \xeCJK_get_punct_bounds:nN { l } {#1}
449   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
450   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
451   \xeCJK_punct_nobreak:
452   \xeCJK_before_FullLeft:N {#1}
453 }
(End definition for \xeCJK_FullRight_and_FullLeft:N)
```

\xeCJK_FullRight_and_FullRight:N

```
454 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullRight:N
455 {
```

```

456 \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
457 \xeCJK_get_punct_bounds:nN { r } {#1}
458 \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
459 \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
460 \nobreak
461 \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
462 \CJKpunctsymbol {#1}
463 }

```

(End definition for \xeCJK_FullRight_and_FullRight:N.)

\xeCJK_CJK_kern: 用于判断插入的各种 kern。

```

\xeCJK_CJK_space_kern: 464 \cs_new_nopar:Nn \xeCJK_CJK_kern: { \tex_kern:D -1 sp \tex_kern:D 1 sp }
\xeCJK_default_kern: 465 \cs_new_nopar:Nn \xeCJK_CJK_space_kern: { \tex_kern:D -2 sp \tex_kern:D 2 sp }
\xeCJK_default_space_kern: 466 \cs_new_nopar:Nn \xeCJK_default_kern: { \tex_kern:D -3 sp \tex_kern:D 3 sp }
467 \cs_new_nopar:Nn \xeCJK_default_space_kern: { \tex_kern:D -4 sp \tex_kern:D 4 sp }
(End definition for \xeCJK_CJK_kern: and others.)

```

CJKglue CJK 文字之间插入的 glue。

```

468 \keys_define:nn { xeCJK / options } { CJKglue .tl_set_x:N = \CJKglue }
(End definition for CJKglue. This function is documented on page 2.)

```

CJKecglue CJK 与西文和数学行内数学公式之间自动添加的空白。

```

xCJKecglue 469 \keys_define:nn { xeCJK / options }
470 {
471   CJKecglue .code:n =
472   {
473     \cs_set:Npx \CJKecglue {#1}
474     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
475   },
476   xCJKecglue .choice:,
477   xCJKecglue / true .code:n =
478   {
479     \bool_set_true:N \l_xeCJK_xecglue_bool
480     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
481   },
482   xCJKecglue / false .code:n =
483   {
484     \bool_set_false:N \l_xeCJK_xecglue_bool
485     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_token
486   },
487   xCJKecglue / unknown .code:n =
488   {
489     \bool_set_true:N \l_xeCJK_xecglue_bool
490     \cs_set:Npx \CJKecglue {#1}
491     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
492     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
493   },
494   xCJKecglue .default:n = { true } ,
495 }
496 \dim_new:N \l_xeCJK_ecglue_dim
497 \bool_new:N \l_xeCJK_xecglue_bool
(End definition for CJKecglue and xCJKecglue. These functions are documented on page 2.)

```

CJKspace 是否保留 CJK 文字间的空白,默认不保留。

```

498 \keys_define:nn { xeCJK / options }
499 {
500   CJKspace .choice:,

```

```

501     CJKspace / true .code:n = { \CJKspace },
502     CJKspace / false .code:n = { \CJKnospace },
503     CJKspace .default:n = { true },
504     space .meta:n = { CJKspace = true },
505     nospace .meta:n = { CJKspace = false },
506 }

```

(End definition for CJKspace. This function is documented on page 2.)

`\CJKspace` 是否保留 CJK 文字间的空白,默认不保留。

```

\CJKnospace 507 \NewDocumentCommand \CJKspace { }
508 {
509     \bool_set_true:N \l_xeCJK_space_bool
510     \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_peek_math:
511 }
512 \NewDocumentCommand \CJKnospace { }
513 {
514     \bool_set_false:N \l_xeCJK_space_bool
515     \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_ignore_spaces:
516 }

```

(End definition for \CJKspace and \CJKnospace.)

`\xeCJK_ignore_spaces:`

`\xeCJK_peek_math:`

```

517 \cs_new_nopar:Nn \xeCJK_ignore_spaces:
518 {
519     \peek_catcode:NTF \c_space_token
520     {
521         \int_compare:nNtT \tex_lastkern:D = \c_one
522         { \tex_unkern:D \tex_unkern:D { \xeCJK_CJK_space_kern: } }
523         \peek_catcode_ignore_spaces:NNTF \c_math_toggle_token
524         { \xeCJK_space_or_xecglue: }
525         { \bool_if:nT { \token_if_cs_p:N \l_peek_token } \xeCJK_space_or_xecglue: }
526     }
527     {
528         \token_if_math_toggle:NNTF \l_peek_token \CJKeckglue
529         { \cs_if_exist:NF \l_peek_token \exp_not:N }
530     }
531 }
532 \cs_new_nopar:Nn \xeCJK_peek_math:
533 {
534     \peek_catcode:NNTF \c_space_token
535     {
536         \int_compare:nNtT \tex_lastkern:D = \c_one
537         { \tex_unkern:D \tex_unkern:D { \xeCJK_CJK_space_kern: } }
538         \peek_catcode_ignore_spaces:NNTF \c_space_token { \xeCJK_space_or_xecglue: }
539     }
540     {
541         \token_if_math_toggle:NNTF \l_peek_token \CJKeckglue
542         { \cs_if_exist:NF \l_peek_token \exp_not:N }
543     }
544 }

```

(End definition for \xeCJK_ignore_spaces: and \xeCJK_peek_math:.)

4.3 段末孤字处理

`CheckSingle` 孤字处理功能选项。

`CheckSingleglue`

```

545 \keys_define:nn { xeCJK / options }
546 {

```

```

547     CheckSingle .choice:,
548     CheckSingle / true .code:n = { \xeCJKenablechecksingle },
549     CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
550     CheckSingle .default:n = { true },
551     CJKchecksingle .meta:n = { CheckSingle = true } ,
552     CheckSingleglue .tl_set_x:N = { \CJKchecksingleglue },
553 }

```

(End definition for CheckSingle. This function is documented on page 2.)

\xeCJKenablechecksingle 孤字处理功能开关。

```

\xeCJKdisablechecksingle
554 \bool_new:N \l_xeCJK_checksingl_bool
555 \NewDocumentCommand \xeCJKenablechecksingle { }
556 {
557     \bool_if:NF \l_xeCJK_checksingl_bool
558     {
559         \bool_set_true:N \l_xeCJK_checksingl_bool
560         \cs_set_eq:NN \xeCJK_checksingl_save_CJKsymbol:N \xeCJK_CJK_and_CJK:N
561         \cs_set_eq:NN \xeCJK_checksingl_save_CJKglue \CJKglue
562         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingl:N
563         \cs_set_nopar:Npn \CJKglue { \CJKchecksingleglue }
564     }
565 }
566 \NewDocumentCommand \xeCJKdisablechecksingle { }
567 {
568     \bool_if:NT \l_xeCJK_checksingl_bool
569     {
570         \bool_set_false:N \l_xeCJK_checksingl_bool
571         \cs_set_eq:NN \CJKglue \xeCJK_checksingl_save_CJKglue
572         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingl_save_CJKsymbol:N
573     }
574 }

```

(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle.)

```

\xeCJK_checksingl:N
\xeCJK_checksingl:NN
\xeCJK_checksingl:NNN
575 \cs_new_nopar:Nn \xeCJK_checksingl:N
576 {
577     \peek_catcode:NTF \c_catcode_other_token
578     { \xeCJK_checksingl:NN {#1} }
579     {
580         \bool_if:nTF
581         {
582             \tl_if_empty_p:x { \token_get_arg_spec:N \l_peek_token } &&
583             \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
584             \exp_after:wN \token_if_other_p:N \l_peek_token
585         }
586         { \xeCJK_checksingl:NN {#1} }
587         { \xeCJK_checksingl_save_CJKsymbol:N {#1} }
588     }
589 }
590 \cs_new_nopar:Nn \xeCJK_checksingl:NN
591 {
592     \peek_catcode:NTF \c_catcode_other_token
593     { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
594     {
595         \bool_if:nTF
596         {
597             \token_if_space_p:N \l_peek_token ||
598             \token_if_math_toggle_p:N \l_peek_token ||

```

```

599         \token_if_cs_p:N          \l_peek_token
600     }
601     {
602         \bool_if:nTF { \token_if_space_p:N \l_peek_token }
603         {
604             \peek_catcode_ignore_spaces:NTF \c_catcode_other_token
605             { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
606             {
607                 \bool_if:nTF
608                 {
609                     \token_if_math_toggle_p:N \l_peek_token ||
610                     \token_if_cs_p:N          \l_peek_token
611                 }
612                 { \xeCJK_checksingl:N {#1} {#2} \c_space_token }
613                 { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
614             }
615         }
616         { \xeCJK_checksingl:N {#1} {#2} { } }
617     }
618     { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
619 }
620 }
621 \cs_new_nopar:Nn \xeCJK_checksingl:N
622 {
623     \bool_if:nTF
624     {
625         \token_if_eq_meaning_p:NN \l_peek_token \par ||
626         \token_if_eq_meaning_p:NN \l_peek_token \[ \% \]
627     }
628     { \CJKsymbol {#1} \nobreak #2 }
629     { \xeCJK_checksingl_env:NNN {#1} {#2} {#3} }
630 }

```

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

\xeCJK_checksingl_env:NNN
\xeCJK_checksingl_env:NNNNn

```

631 \cs_new_nopar:Nn \xeCJK_checksingl_env:NNN
632 {
633     \peek_catcode_remove:NTF \c_math_toggle_token
634     {
635         \peek_catcode:NTF \c_math_toggle_token
636         { \CJKsymbol {#1} \nobreak #2 \c_math_toggle_token }
637         { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 \c_math_toggle_token }
638     }
639     {
640         \peek_meaning_remove:NTF \begin
641         { \xeCJK_checksingl_env:NNNNn {#1} {#2} {#3} \begin }
642         {
643             \peek_meaning_remove:NTF \end
644             { \xeCJK_checksingl_env:NNNNn {#1} {#2} {#3} \end }
645             { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 }
646         }
647     }
648 }
649 \cs_new_nopar:Nn \xeCJK_checksingl_env:NNNNn
650 {
651     \clist_if_in:NnTF \g_xeCJK_inline_env_clist {#5}
652     { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 }
653     { \CJKsymbol {#1} \nobreak #2 }

```

```

654     \scan_stop: #4 {#5}
655   }
(End definition for \xeCJK_checksingl_env:NNN and \xeCJK_checksingl_env:NNNNn.)

\g_xeCJK_inline_env_clist

656 \clist_new:N \g_xeCJK_inline_env_clist
(End definition for \g_xeCJK_inline_env_clist.)

InlineEnv
InlineEnv+
InlineEnv-
657 \keys_define:nn { xeCJK / options }
658 {
659   InlineEnv .clist_gset:N = \g_xeCJK_inline_env_clist ,
660   InlineEnv+ .code:n =
661   {
662     \clist_gput_right:Nx \g_xeCJK_inline_env_clist {#1}
663     \clist_gremove_duplicates:N \g_xeCJK_inline_env_clist
664   },
665   InlineEnv- .code:n =
666   {
667     \clist_map_inline:xn {#1}
668     { \clist_gremove_all:Nn \g_xeCJK_inline_env_clist {##1} }
669   },
670 }
(End definition for InlineEnv, InlineEnv+, and InlineEnv-. These functions are documented on page 2.)

```

4.4 增加 CJK 子分区

```

\xeCJK_UL_subclass_patch_tl
\g_xeCJK_CJK_subclass_clist
671 \tl_new:N \xeCJK_UL_subclass_patch_tl
672 \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:
673 \cs_new_nopar:Nn \xeCJK_erase_CJKsymbol:
674 {
675   \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
676   \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
677 }
678 \cs_new_nopar:Nn \xeCJK_restore_CJKsymbol:
679 {
680   \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block
681   \CJKsymbol
682 }
(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。
683 \NewDocumentCommand \xeCJKDeclareSubCJKBlock { m m }
684 { \xeCJKDeclareSubCharClass { CJK } {#1} {#2} }
685 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock. This function is documented on page 7.)

```

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

\xeCJKRestoreSubCJKBlock

```
686 \bool_new:N \l_xeCJK_sub_cancel_bool
687 \NewDocumentCommand \xeCJKCancelSubCJKBlock { m }
688 {
689   \bool_if:NF \l_xeCJK_sub_cancel_bool
690   {
691     \bool_set_true:N \l_xeCJK_sub_cancel_bool
692     \xeCJK_sub_restore_or_cancel:n {#1}
693   }
694 }
695 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { m }
696 {
697   \bool_if:NT \l_xeCJK_sub_cancel_bool
698   {
699     \bool_set_false:N \l_xeCJK_sub_cancel_bool
700     \xeCJK_sub_restore_or_cancel:n {#1}
701   }
702 }
703 \cs_new_nopar:Nn \xeCJK_sub_restore_or_cancel:n
704 {
705   \clist_map_inline:xn {#1}
706   {
707     \cs_if_free:cTF { xeCJK_ CJK/##1 _class }
708     { \xeCJK_error:nx { SubBlock-undefined } {##1} }
709     {
710       \xeCJKDeclareCharClass
711       { CJK \bool_if:NF \l_xeCJK_sub_cancel_bool { /##1 } }
712       { \clist_use:c { g_xeCJK_CJK/##1_range_clist } }
713     }
714   }
715 }
716 \xeCJK_msg_new:nn { SubBlock-undefined }
717 {
718   The~CJK~sub~block~'##1'~is~undefined.\\\
719   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \c_space_tl
720   to~declare~it.
721 }
```

(End definition for \xeCJKCancelSubCJKBlock and \xeCJKRestoreSubCJKBlock. These functions are documented on page 7.)

\xeCJKDeclareSubCharClass

```
722 \NewDocumentCommand \xeCJKDeclareSubCharClass
723 { > { \TrimSpaces } m > { \TrimSpaces } m m }
724 {
725   \cs_if_free:cT { xeCJK_ #1/#2 _class }
726   {
727     \xeCJK_new_class:n { #1/#2 }
728     \xeCJK_set_sub_class_toks:nn {#1} {#2}
729     \xeCJK_new_sub_key:n {#2}
730     \xeCJK_UL_subclass_patch:nn {#1} {#2}
731   }
732   \xeCJKDeclareCharClass { #1/#2 } {#3}
733 }
734 \@onlypreamble \xeCJKDeclareSubCharClass
735 (End definition for \xeCJKDeclareSubCharClass.)
```

\xeCJK_set_sub_class_toks:nn

```
735 \cs_new_nopar:Nn \xeCJK_set_sub_class_toks:nn
```

```

736 {
737   \clist_map_inline:Nn \g_xeCJK_base_class_clist
738   {
739     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
740     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
741     \str_if_eq:xxTF {##1} { CJK }
742     {
743       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
744       { \xeCJK_switch_font:nn {#1} {#2} }
745     }
746     {
747       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
748       { \xeCJK_erase_CJKsymbol: }
749       \xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
750       { \xeCJK_switch_font:nn {#1} {#2} \xeCJK_restore_CJKsymbol: }
751     }
752   }
753   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
754   \clist_if_empty:NF \g_xeCJK_CJK_subclass_clist
755   {
756     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
757     {
758       \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
759       \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
760       \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
761       { \xeCJK_switch_font:nn {#2} {##1} }
762       \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
763       { \xeCJK_switch_font:nn {##1} {#2} }
764     }
765   }
766   \clist_gput_right:Nn \g_xeCJK_CJK_subclass_clist {#2}
767   \xeCJK_save_CJK_class:n { #1/#2 }
768   \clist_map_inline:nn { CJK, FullLeft, FullRight }
769   {
770     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
771     { \xeCJK_switch_font:nn {#2} {#1} }
772   }
773 }
(End definition for \xeCJK_set_sub_class_toks:nn.)

```

\xeCJK_UL_subclass_patch:nn

```

774 \cs_new_nopar:Nn \xeCJK_UL_subclass_patch:nn
775 {
776   \tl_put_right:Nn \xeCJK_UL_subclass_patch_tl
777   {
778     \clist_map_inline:nn { Default, HalfLeft, HalfRight }
779     {
780       \str_if_eq:xxTF {##1} { HalfLeft }
781       { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKecglue \CJKsymbol } }
782       { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
783       \xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
784       {
785         \xeCJK_erase_CJKsymbol:
786         \xeCJK_Boundary_and_CJK:
787         \xeCJK_switch_font:nn {#1} {#2}
788         \xeCJK_restore_CJKsymbol:
789       }
790     }
791   }

```

```

791     }
792 }
(End definition for \xeCJK_UL_subclass_patch:nn.)

```

4.5 标点处理

\g_xeCJK_punctstyle_clist 标点处理方式。

```

793 \clist_new:N \g_xeCJK_punctstyle_clist
794 \clist_set:Nn \g_xeCJK_punctstyle_clist
795 { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
796 \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
797 { \tl_const:cn { c_xeCJK_ps_#1_tl } {#1} }
(End definition for \g_xeCJK_punctstyle_clist.)

```

\keys_choices_generate:V

.generate_choices:V

```

798 \cs_generate_variant:Nn \keys_choices_generate:n { V }
799 \cs_new_protected:cpn { \c_keys_props_root_tl .generate_choices:V } #1
800 { \keys_choices_generate:V {#1} }
(End definition for \keys_choices_generate:V and .generate_choices:V.)

```

PunctStyle

相关选项声明。

AllowBreakBetweenPuncts

KaiMingPunct

LongPunct

MiddlePunct

PunctWidth

```

801 \keys_define:nn { xeCJK / options }
802 {
803   AllowBreakBetweenPuncts .choice:,
804   AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
805   AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
806   AllowBreakBetweenPuncts .default:n = { true },
807   KaiMingPunct .code:n = { \xeCJK_set_special_punct:nn { mixedwidth } {#1} },
808   KaiMingPunct+ .code:n = { \xeCJK_add_special_punct:nn { mixedwidth } {#1} },
809   KaiMingPunct- .code:n = { \xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
810   LongPunct .code:n = { \xeCJK_set_special_punct:nn { long } {#1} },
811   LongPunct+ .code:n = { \xeCJK_add_special_punct:nn { long } {#1} },
812   LongPunct- .code:n = { \xeCJK_sub_special_punct:nn { long } {#1} },
813   MiddlePunct+ .code:n = { \xeCJK_add_special_punct:nn { middle } {#1} },
814   MiddlePunct .code:n = { \xeCJK_set_special_punct:nn { middle } {#1} },
815   MiddlePunct- .code:n = { \xeCJK_sub_special_punct:nn { middle } {#1} },
816   PunctWidth .tl_gset:N = \g_xeCJK_punct_width_tl ,
817   PunctStyle .choice_code:n =
818   {
819     \tl_set:Nx \l_xeCJK_punctstyle_tl { \l_keys_choice_tl }
820     \tl_if_eq:NNT \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
821     { \xeCJKallowbreakbetweenpuncts }
822   },
823   PunctStyle .generate_choices:V = \g_xeCJK_punctstyle_clist,
824   PunctStyle / banjiao .meta:n = { PunctStyle = halfwidth },
825   PunctStyle / quanjiao .meta:n = { PunctStyle = fullwidth },
826   PunctStyle / kaiming .meta:n = { PunctStyle = mixedwidth },
827   PunctStyle / hangmobanjiao .meta:n = { PunctStyle = marginkerning },
828   PunctStyle / unknown .code:n =
829   { \xeCJK_error:nx { PunctStyle-undefined } \l_keys_value_tl },
830 }
831 \xeCJK_msg_new:nn { PunctStyle-undefined }
832 {
833   Punctstyle~"~"~#1~"~is~not~available. \\\
834   You~can~only~use~one~of\\\
835   "\g_xeCJK_punctstyle_clist".\\
836 }

```

(End definition for PunctStyle and others. These functions are documented on page 3.)

```
\xeCJKallowbreakbetweenpuncts
\xeCJKnobreakbetweenpuncts 837 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
838 { \cs_set_eq:NN \xeCJK_punct_nobreak: \xeCJK_zero_glue: }
839 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
840 { \cs_set_eq:NN \xeCJK_punct_nobreak: \nobreak }
841 \cs_new_protected_nopar:Nn \xeCJK_zero_glue: { \skip_horizontal:N \c_zero_skip }
(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts.)
```

```
\g_xeCJK_special_ps_clist
\xeCJK_set_special_punct:nn 842 \clist_set:Nn \g_xeCJK_special_ps_clist { mixedwidth , long , middle }
\xeCJK_add_special_punct:nn 843 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
\xeCJK_sub_special_punct:nn 844 {
845   \tl_new:c { l_xeCJK_#1_punct_tl }
846   \prop_new:c { l_xeCJK_#1_punct_prop }
847 }
848 \cs_new_nopar:Nn \xeCJK_set_special_punct:nn
849 {
850   \tl_set:cx { l_xeCJK_#1_punct_tl } {#2}
851   \prop_clear:c { l_xeCJK_#1_punct_prop }
852   \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
853     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
854 }
855 \cs_new_nopar:Nn \xeCJK_add_special_punct:nn
856 {
857   \tl_put_right:cx { l_xeCJK_#1_punct_tl } {#2}
858   \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
859     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
860 }
861 \cs_new_nopar:Nn \xeCJK_sub_special_punct:nn
862 {
863   \tl_map_inline:xn {#2}
864   {
865     \tl_remove_all:cn { l_xeCJK_#1_punct_tl } {##1}
866     \prop_del:cn { l_xeCJK_#1_punct_prop } {##1}
867   }
868 }
869 \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 870 \cs_generate_variant:Nn \dim_set:Nn { Nc }
\dim_set_min:Nc 871 \cs_generate_variant:Nn \dim_add:Nn { Nc }
872 \cs_generate_variant:Nn \dim_set_min:Nn { Nc }
(End definition for \dim_set:Nc, \dim_add:Nc, and \dim_set_min:Nc.)
```

```
\l_xeCJK_punct_coors_tl 记录当前字体状态下的标点处理方式。
873 \tl_set:Nn \l_xeCJK_punct_coors_tl { \l_xeCJK_font_coors_tl/\l_xeCJK_punctstyle_tl }
(End definition for \l_xeCJK_punct_coors_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。
874 \cs_new_nopar:Nn \xeCJK_get_punct_bounds:nN
875 {
876   \tl_if_exist:cF { \l_xeCJK_punct_coors_tl/rule/#1/#2 }
```

```

877 {
878   \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
879   {
880     \clist_map_inline:nn { glue , rule , bound }
881     { \tl_gset:cV { \l_xeCJK_punct_coors_tl/##1/##1/##2 } \c_zero_dim }
882     \tl_gset:cV { \l_xeCJK_punct_coors_tl/rule/m/##2 } \c_zero_dim
883   }
884   {
885     \tl_if_exist:cF { \l_xeCJK_punct_coors_tl/bound/##1/##2 }
886     { { \xeCJK_select_font: \xeCJK_get_punct_dimen:N {##2} } }
887     \xeCJK_punct_if_long:NNTF {##2}
888     {
889       \dim_zero:N \l_tmpa_dim
890       \dim_zero:N \l_tmpb_dim
891       \dim_zero:N \l_tmpc_dim
892     }
893     {
894       \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coors_tl/bound/##1/##2 }
895       \dim_set:Nc \l_tmpb_dim
896       { \l_xeCJK_punct_coors_tl/bound/\str_if_eq:xxTF {##1} lrl/##2 }
897       \dim_set:Nn \l_tmpc_dim
898       {
899         \tl_if_exist:cTF { g_xeCJK_punct_width/##2 }
900         { \tl_use:c { g_xeCJK_punct_width/##2 } }
901         {
902           \tl_if_blank:xTF \g_xeCJK_punct_width_tl
903           { \c_zero_dim }
904           { \g_xeCJK_punct_width_tl }
905         }
906       }
907       \xeCJK_punct_if_middle:NNTF {##2}
908       {
909         \dim_set:Nn \l_tmpc_dim
910         {
911           \dim_compare:nNnTF \l_tmpc_dim > \c_zero_dim
912           { \l_tmpc_dim }
913           {
914             \prg_case_tl:Nnn \l_xeCJK_punctstyle_tl
915             {
916               \c_xeCJK_ps_halfwidth_tl { .5 em }
917               \c_xeCJK_ps_mixedwidth_tl { .5 em }
918               \c_xeCJK_ps_CCT_tl { .7 em }
919               \c_xeCJK_ps_fullwidth_tl { 1 em }
920             }
921             { 1 em }
922           }
923           - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/##2 }
924         }
925         \dim_set:Nn \l_tmpc_dim { .5 \l_tmpc_dim }
926       }
927       {
928         \dim_compare:nNnTF \l_tmpc_dim > \c_zero_dim
929         {
930           \dim_set:Nn \l_tmpc_dim
931           {
932             \l_tmpc_dim - \l_tmpb_dim
933             - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/##2 }
934           }

```

```

935     }
936     {
937         \dim_set_eq:Nn \l_tmpc_dim \l_tmpa_dim
938         \dim_set_min:Nn \l_tmpc_dim \l_tmpb_dim
939         \prg_case_tl:Nnn \l_xeCJK_punctstyle_tl
940         {
941             \c_xeCJK_ps_halfwidth_tl { \prg_do_nothing: }
942             \c_xeCJK_ps_mixedwidth_tl
943             {
944                 \xeCJK_punct_if_mixedwidth:NT {#2}
945                 {
946                     \dim_add:Nn \l_tmpc_dim { .5 \l_tmpa_dim }
947                     \dim_set_min:Nn \l_tmpc_dim \l_tmpa_dim
948                 }
949             }
950             \c_xeCJK_ps_CCT_tl
951             {
952                 \xeCJK_punct_if_mixedwidth:NTF {#2}
953                 { \dim_add:Nn \l_tmpc_dim { .5 \l_tmpa_dim } }
954                 { \dim_add:Nn \l_tmpc_dim { .3 \l_tmpa_dim } }
955                 \dim_set_min:Nn \l_tmpc_dim \l_tmpa_dim
956             }
957         }
958         { \dim_set_eq:Nn \l_tmpc_dim \l_tmpa_dim }
959     }
960 }
961 \dim_set_max:Nn \l_tmpc_dim \c_zero_dim
962 }
963 \tl_gset:cx { \l_xeCJK_punct_coors_tl/rule/#1/#2 }
964 { - \dim_use:N \l_tmpa_dim }
965 \tl_gset:cx { \l_xeCJK_punct_coors_tl/rule/m/#2 }
966 { - \dim_use:N \l_tmpb_dim }
967 \tl_gset:cV { \l_xeCJK_punct_coors_tl/glue/#1/#2 } \l_tmpc_dim
968 }
969 }
970 }
971 \cs_generate_variant:Nn \xeCJK_get_punct_bounds:nN { nV }
(End definition for \xeCJK_get_punct_bounds:nN.)

```

\xeCJKsetwidth

```

972 \NewDocumentCommand \xeCJKsetwidth { m m }
973 { \tl_map_inline:xn {#1} { \tl_gset:cn { g_xeCJK_punct_width/##1 } {#2} } }
(End definition for \xeCJKsetwidth. This function is documented on page 7.)

```

```

\c_xeCJK_punct_if_right_p:N
\c_xeCJK_punct_if_both_left_right_p:N
\c_xeCJK_punct_if_long_p:N
\c_xeCJK_punct_if_middle_p:N
\c_xeCJK_punct_if_mixedwidth_p:N
\c_xeCJK_punct_if_right:N
\c_xeCJK_punct_if_both_left_right:N
\c_xeCJK_punct_if_long:N
\c_xeCJK_punct_if_middle:N
\c_xeCJK_punct_if_mixedwidth:N
974 \prg_new_conditional:Nnn \xeCJK_punct_if_right:N { p , T , F , TF }
975 {
976     \tl_if_exist:cTF { \l_xeCJK_punct_coors_tl/glue/r/#1 }
977     \prg_return_true: \prg_return_false:
978 }
979 \prg_new_conditional:Nnn \xeCJK_punct_if_both_left_right:N { p , T , F , TF }
980 {
981     \bool_if:nTF
982     {
983         ( \tl_if_exist_p:c { \l_xeCJK_punct_coors_tl/glue/r/#1 } &&
984           \tl_if_exist_p:c { \l_xeCJK_punct_coors_tl/glue/r/#2 } ) ||
985         ( \tl_if_exist_p:c { \l_xeCJK_punct_coors_tl/glue/l/#1 } &&
986           \tl_if_exist_p:c { \l_xeCJK_punct_coors_tl/glue/l/#2 } )

```

```

987     }
988     \prg_return_true: \prg_return_false:
989   }
990   \cs_generate_variant:Nn \prg_new_conditional:Nnn { c }
991   \clist_map_inline:Nn \g_xeCJK_special_ps_clist
992   {
993     \prg_new_conditional:cnn { xeCJK_punct_if_#1:N } { p , T , F , TF }
994     {
995       \prop_if_in:cnTF { l_xeCJK_#1_punct_prop } {##1}
996       \prg_return_true: \prg_return_false:
997     }
998   }

```

(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。

```

999   \cs_new_nopar:Nn \xeCJK_get_kern:NN
1000   {
1001     \tl_if_exist:cF { \l_xeCJK_punct_coor_tl/kern/#1-#2 }
1002     {
1003       \dim_zero:N \l_tmpa_dim
1004       \tl_if_eq:NnF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1005       {
1006         \tl_if_exist:cTF { g_xeCJK_punct/kern/#1-#2 }
1007         { \dim_set:Nc \l_tmpa_dim { g_xeCJK_punct/kern/#1-#2 } }
1008         { \xeCJK_calc_kern:NN {#1} {#2} }
1009       }
1010       \tl_gset:cV { \l_xeCJK_punct_coor_tl/kern/#1-#2 } \l_tmpa_dim
1011     }
1012   }
1013   \cs_new_nopar:Nn \xeCJK_calc_kern:NN
1014   {
1015     \tl_if_eq:NnTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_marginkerning_tl
1016     {
1017       \xeCJK_punct_if_right:NT {#1}
1018       { \dim_add:Nc \l_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/r/#1 } }
1019       \xeCJK_punct_if_right:NF {#2}
1020       { \dim_add:Nc \l_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/l/#2 } }
1021     }
1022     {
1023       \tl_if_eq:NnTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_fullwidth_tl
1024       { \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1.5 em } {#1} {#2} }
1025       {
1026         \xeCJK_punct_if_both_left_right:NnTF {#1} {#2}
1027         {
1028           \xeCJK_punct_if_mixedwidth:NT {#1}
1029           {
1030             \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1 em } {#1} {#2}
1031             \dim_compare:nNnT \l_tmpa_dim < { .1 em }
1032             { \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/l/#1 } }
1033           }
1034         }
1035         {
1036           \bool_if:nTF
1037           {
1038             \xeCJK_punct_if_mixedwidth_p:N {#1} &&
1039             ! ( \tl_if_eq_p:NN \l_xeCJK_punctstyle_tl \c_xeCJK_ps_halfwidth_tl )
1040           }

```

```

1041         {
1042             \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/glue/r/#1 }
1043             \dim_set:Nn \l_tmpa_dim { .7 \l_tmpa_dim }
1044         }
1045         {
1046             \xeCJK_calc_kern:NnnN \l_tmpa_dim { 1 em } {#1} {#2}
1047             \dim_compare:nNnT \l_tmpa_dim < { .1 em }
1048             {
1049                 \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#1 }
1050                 \dim_set_min:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/r/#1 }
1051                 \dim_set:Nc \l_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/l/#2 }
1052                 \dim_set_min:Nc \l_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/r/#2 }
1053                 \dim_set_max:Nn \l_tmpa_dim \l_tmpb_dim
1054             }
1055         }
1056     }
1057 }
1058 \bool_if:nT
1059 { \xeCJK_punct_if_long_p:N {#1} || \xeCJK_punct_if_long_p:N {#2} }
1060 { \dim_set_max:Nn \l_tmpa_dim { .1 em } }
1061 }
1062 \dim_set_max:Nn \l_tmpa_dim \c_zero_dim
1063 }
1064 \cs_generate_variant:Nn \xeCJK_get_kern:NN { V }
1065 (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。

```

1065 \cs_new_nopar:Nn \xeCJK_calc_kern:NnnN
1066 {
1067     \dim_set:Nn #1
1068     {
1069         #2
1070         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1071             \xeCJK_punct_if_right:NNT {#3} { bound } { glue } /l/#3 }
1072         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1073             \xeCJK_punct_if_right:NNT {#4} { glue } { bound } /r/#4 }
1074         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#3 }
1075         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#4 }
1076         \xeCJK_punct_if_both_left_right:NNT {#3} {#4}
1077         { + \tl_use:c { \l_xeCJK_punct_coord_tl/rule/m/#4 } }
1078     }
1079 }
1080 (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。

```

1080 \cs_new_nopar:Nn \xeCJK_get_punct_dimen:N
1081 {
1082     \xeCJK_get_glyph_bounds:nNN { left } {#1} \l_tmpa_dim
1083     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/l/#1 } \l_tmpa_dim
1084     \dim_set_eq:NN \l_tmpb_dim \l_tmpa_dim
1085     \xeCJK_get_glyph_bounds:nNN { right } {#1} \l_tmpa_dim
1086     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/r/#1 } \l_tmpa_dim
1087     \dim_add:Nn \l_tmpb_dim \l_tmpa_dim
1088     \tl_gset:cx { \l_xeCJK_punct_coord_tl/dimen/#1 }
1089     { \dim_eval:n { \tex_fontcharwd:D \tex_font:D `#1 - \l_tmpb_dim } }
1090     \bool_if:nT

```

```

1091 {
1092   \xeCJK_punct_if_long_p:N {#1} &&
1093   ! ( \str_if_eq_p:xx {#1} { \cdots } || \str_if_eq_p:xx {#1} { \cdots } )
1094 }
1095 {
1096   \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
1097   {
1098     \str_if_eq:xxF \c_xeCJK_ps_plain_tl {##1}
1099     {
1100       \tl_gset:cx { \l_xeCJK_font_coors_tl/##1/kern/#1-#1 }
1101       { - \dim_use:N \l_tmpb_dim }
1102     }
1103   }
1104 }
1105 }
(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。

```

1106 \cs_new_nopar:Nn \xeCJK_get_glyph_bounds:nNN
1107 {
1108   \dim_set:Nn #3
1109   {
1110     \XeTeXglyphbounds
1111     \prg_case_str:xxn {#1}
1112     {
1113       { left } { \c_one } { top } { \c_two }
1114       { right } { \c_three } { bottom } { \c_four }
1115     }
1116     { \c_three }
1117     \int_eval:n { \XeTeXcharglyph `#2 }
1118   }
1119 }
(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。

```

1120 \NewDocumentCommand \xeCJKsetkern { m m m }
1121 { \tl_gset:cn { g_xeCJK_punct/kern/#1-#2 } {#3} }
(End definition for \xeCJKsetkern. This function is documented on page 8.)

```

4.6 后备字体

`AutoFallback` 后备字体的宏包选项声明。

```

1122 \keys_define:nn { xeCJK / options }
1123 {
1124   AutoFallback .choice:,
1125   AutoFallback / true .code:n = { \xeCJKenablefallback },
1126   AutoFallback / false .code:n = { \xeCJKdisablefallback },
1127   AutoFallback .default:n = { true },
1128   fallback .meta:n = { AutoFallback = true },
1129 }
(End definition for AutoFallback. This function is documented on page 3.)

```

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

```

\xeCJKdisablefallback
1130 \bool_new:N \l_xeCJK_fallback_bool
1131 \NewDocumentCommand \xeCJKenablefallback { }
1132 {
1133   \bool_if:NF \l_xeCJK_fallback_bool

```

```

1134     {
1135         \bool_set_true:N \l_xeCJK_fallback_bool
1136         \cs_set_eq:NN \xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1137         \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_testsymbol:N
1138     }
1139 }
1140 \NewDocumentCommand \xeCJKdisablefallback { }
1141 {
1142     \bool_if:NT \l_xeCJK_fallback_bool
1143     {
1144         \bool_set_false:N \l_xeCJK_fallback_bool
1145         \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_save_CJKsymbol:N
1146     }
1147 }
1148 (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。

```

1148 \cs_new_nopar:Nn \xeCJK_fallback_testsymbol:N
1149 {
1150     \font_glyph_if_exist:NnTF \tex_font:D {`#1}
1151     { \xeCJK_fallback_save_CJKsymbol:N {#1} }
1152     {
1153         \xeCJK_family_if_exist:nTF { \xeCJK@family/FallBack }
1154         { {
1155             \tl_put_right:Nn \xeCJK@family { /FallBack }
1156             \xeCJK_select_font:
1157             \xeCJK_fallback_testsymbol:N {#1}
1158         } }
1159         {
1160             \xeCJK_warning:nx { fallback } {#1}
1161             \xeCJK_fallback_save_CJKsymbol:N {#1}
1162         }
1163     }
1164 }
1165 \xeCJK_msg_new:nn { fallback }
1166 {
1167     CJKfamily~'\xeCJK@family'~
1168     ( \prop_get:NV \g_xeCJK_family_prop \xeCJK@family )~
1169     does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {`#1}).\\
1170 }
1171 (End definition for \xeCJK_fallback_testsymbol:N.)

```

\setCJKfallbackfamilyfont

```

1171 \xeCJK_set_family_fallback:nnn \NewDocumentCommand \setCJKfallbackfamilyfont { m O{ } m }
1172 {
1173     \exp_args:Nx \tl_if_in:nnTF {#3} { , }
1174     { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} }
1175     { \xeCJK_set_family:nnn { #1/FallBack } {#2} {#3} }
1176 }
1177 \cs_new_nopar:Nn \xeCJK_set_family_fallback:nnn
1178 {
1179     \group_begin:
1180     \tl_set:Nx \l_xeCJK_family_fb_tl {#1}
1181     \prop_get:NVNF \g_xeCJK_family_prop \l_xeCJK_family_fb_tl \l_xeCJK_fontname_tl
1182     { \tl_clear:N \l_xeCJK_fontname_tl }
1183     \clist_map_inline:xn {#3}
1184     {
1185         \tl_put_right:Nn \l_xeCJK_family_fb_tl { /FallBack }

```

```

1186     \xeCJK_get_sub_features:nn \l_xeCJK_family_fb_tl {##1}
1187     \clist_put_left:cx { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist } {#2}
1188     \xeCJK_set_family:ncc \l_xeCJK_family_fb_tl
1189         { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist }
1190         { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontname_tl }
1191     }
1192     \group_end:
1193 }

```

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

4.7 CJK 字体族声明方式

```

1194 \bool_new:N \g_xeCJK_AutoFakeBold_bool
1195 \bool_new:N \g_xeCJK_AutoFakeSlant_bool
1196 \fp_new:N \g_xeCJK_EmboldenFactor_fp
1197 \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
1198 \keys_define:nn { xeCJK / options }
1199 {
1200     AutoFakeBold .choice:,
1201     AutoFakeBold / true .code:n =
1202         { \bool_set_true:N \g_xeCJK_AutoFakeBold_bool },
1203     AutoFakeBold / false .code:n =
1204         { \bool_set_false:N \g_xeCJK_AutoFakeBold_bool },
1205     AutoFakeBold / unknown .code:n =
1206         {
1207             \bool_set_true:N \g_xeCJK_AutoFakeBold_bool
1208             \fp_set:Nn \g_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1209         },
1210     AutoFakeBold .default:n = { true },
1211     AutoFakeSlant .choice:,
1212     AutoFakeSlant / true .code:n =
1213         { \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool },
1214     AutoFakeSlant / false .code:n =
1215         { \bool_set_false:N \g_xeCJK_AutoFakeSlant_bool },
1216     AutoFakeSlant / unknown .code:n =
1217         {
1218             \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool
1219             \fp_set:Nn \g_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1220         },
1221     AutoFakeSlant .default:n = { true },
1222     EmboldenFactor .fp_set:N = \g_xeCJK_EmboldenFactor_fp,
1223     SlantFactor .fp_set:N = \g_xeCJK_SlantFactor_fp,
1224     BoldFont .meta:n = { AutoFakeBold = true },
1225     boldfont .meta:n = { AutoFakeBold = true },
1226     SlantFont .meta:n = { AutoFakeSlant = true },
1227     slantfont .meta:n = { AutoFakeSlant = true },
1228 }

```

(End definition for AutoFakeBold and others. These functions are documented on page 3.)

\keys_set_known:nxN

```

1229 \cs_generate_variant:Nn \keys_set_known:nnN { nx }

```

(End definition for \keys_set_known:nxN.)

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

```

\g_xeCJK_sub_key_clist
1230 \clist_new:N \g_xeCJK_sub_key_clist
1231 \cs_new_nopar:Nn \xeCJK_new_sub_key:n

```

```

1232 {
1233   \clist_gput_right:Nx \g_xeCJK_sub_key_clist {#1}
1234   \keys_define:nn { xeCJK / features }
1235   {
1236     #1 .code:n =
1237     {
1238       \tl_if_blank:xTF {##1}
1239       {
1240         \bool_set_false:c { l_xeCJK_copy_#1_bool }
1241         \bool_set_false:c { l_xeCJK_add_#1_bool }
1242         \tl_put_right:Nx \l_xeCJK_family_tl { /#1 }
1243       }
1244       {
1245         \clist_put_right:Nx \l_xeCJK_sub_key_clist {#1}
1246         \str_if_eq:xxTF {##1} *
1247         {
1248           \bool_set_true:c { l_xeCJK_copy_#1_bool }
1249           \bool_set_false:c { l_xeCJK_add_#1_bool }
1250         }
1251         {
1252           \bool_set_false:c { l_xeCJK_copy_#1_bool }
1253           \bool_set_true:c { l_xeCJK_add_#1_bool }
1254           \xeCJK_get_sub_features:nn {#1} {##1}
1255         }
1256       }
1257     },
1258     #1 .default:n = \c_empty_tl,
1259   }
1260 }
1261 \cs_generate_variant:Nn \xeCJK_new_sub_key:n { x }
1262 \cs_new_nopar:Nn \xeCJK_get_sub_features:nn
1263 {
1264   \tl_set:Nx \l_tmpa_tl {#2}
1265   \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1266   \xeCJK_gobble_brace:N \l_tmpa_tl
1267   \exp_args:No \tl_if_head_eq_charcode:nNTF \l_tmpa_tl [ % ]
1268   {
1269     \exp_after:wN \xeCJK_get_sub_features_aux:n \l_tmpa_tl \c_empty_tl
1270     \tl_if_blank:xT \l_xeCJK_sub_fontname_tl
1271     {
1272       \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_tmpa_tl
1273       \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1274     }
1275   }
1276   { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_tmpa_tl }
1277   \tl_if_blank:xTF \l_xeCJK_sub_fontname_tl
1278   { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_fontname_tl }
1279   { \tl_replace_all:Nnx \l_xeCJK_sub_fontname_tl * \l_xeCJK_fontname_tl }
1280   \clist_set:cx { l_xeCJK_#1_fontfeat_clist } \l_xeCJK_sub_fontfeat_clist
1281   \tl_set:cx { l_xeCJK_#1_fontname_tl } \l_xeCJK_sub_fontname_tl
1282 }

```

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

```

\xeCJK_gobble_brace:N
\xeCJK_get_sub_features:nn 1283 \cs_new_nopar:Nn \xeCJK_gobble_brace:N
\xeCJK_get_sub_features_aux:n 1284 {
1285   \exp_args:No \tl_if_head_group:nT #1
1286   {

```

```

1287         \exp_after:wN \tl_set:Nn \exp_after:wN #1 #1
1288         \xeCJK_gobble_brace:N #1
1289     }
1290 }
1291 \cs_new_nopar:Npn \xeCJK_get_sub_features_aux:n [#1] #2
1292 {
1293     \clist_set:Nx \l_xeCJK_sub_fontfeat_clist {#1}
1294     \tl_set:Nx \l_xeCJK_sub_fontname_tl {#2}
1295 }
(End definition for \xeCJK_gobble_brace:N, \xeCJK_get_sub_features:nn, and \xeCJK_get_sub_features_aux:n.)

```

FallBack

```

1296 \xeCJK_new_sub_key:n { FallBack }
(End definition for FallBack. This function is documented on page 6.)

```

.clist_set_x:N

```

.clist_set_x:c
1297 \cs_set_protected:cpn { \c_keys_props_root_tl .clist_set_x:N } #1
1298 { \keys_variable_set:NnN #1 { clist } x }
1299 \cs_set_protected:cpn { \c_keys_props_root_tl .clist_set_x:c } #1
1300 { \keys_variable_set:cnN {#1} { clist } x }
(End definition for .clist_set_x:N and .clist_set_x:c.)

```

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

```

    BoldFont
    ItalicFont
    BoldItalicFont
    SlantedFont
    BoldSlantedFont
    BoldFeatures
    ItalicFeatures
    BoldItalicFeatures
    SlantedFeatures
    BoldSlantedFeatures
1301 \cs_new_nopar:Nn \xeCJK_map_features_id:n
1302 { \prop_get:Nn \g_xeCJK_features_id_prop { #1 } }
1303 \prop_new:N \g_xeCJK_features_id_prop
1304 \prop_put:Nnn \g_xeCJK_features_id_prop { bf } { Bold }
1305 \prop_put:Nnn \g_xeCJK_features_id_prop { it } { Italic }
1306 \prop_put:Nnn \g_xeCJK_features_id_prop { bfit } { BoldItalic }
1307 \prop_put:Nnn \g_xeCJK_features_id_prop { sl } { Slanted }
1308 \prop_put:Nnn \g_xeCJK_features_id_prop { bfsl } { BoldSlanted }
1309 \prop_map_inline:Nn \g_xeCJK_features_id_prop
1310 {
1311     \keys_define:nn { xeCJK / features }
1312     {
1313         #2Font .tl_set_x:c = { l_xeCJK_fontname_#1_tl },
1314         #2Features .clist_set_x:c = { l_xeCJK_fontfeat_#1_clist } ,
1315     }
1316 }
(End definition for \g_xeCJK_features_id_prop and others.)

```

AutoFakeBold

```

AutoFakeSlant
Mono
1317 \keys_define:nn { xeCJK / features }
1318 {
1319     AutoFakeBold .choice:,
1320     AutoFakeBold / false .code:n =
1321     { \bool_set_false:N \l_xeCJK_AutoFakeBold_bool },
1322     AutoFakeBold / unknown .code:n =
1323     {
1324         \bool_set_true:N \l_xeCJK_AutoFakeBold_bool
1325         \fp_set:Nn \l_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1326     },
1327     AutoFakeBold .default:n = { \g_xeCJK_EmboldenFactor_fp },
1328     AutoFakeSlant .choice:,
1329     AutoFakeSlant / false .code:n =
1330     { \bool_set_false:N \l_xeCJK_AutoFakeSlant_bool },
1331     AutoFakeSlant / unknown .code:n =

```

```

1332     {
1333         \bool_set_true:N \l_xeCJK_AutoFakeSlant_bool
1334         \fp_set:Nn \l_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1335     },
1336     AutoFakeSlant .default:n = { \g_xeCJK_SlantFactor_fp },
1337     Mono .choice:,
1338     Mono / Exspace .code:n = { \xeCJK_set_monoexspace: },
1339     Mono / Scale .code:n =
1340     {
1341         \xeCJK_set_monoscale:
1342         \tl_set:Nx \l_xeCJK_monoscale_tl
1343         { Scale = { \fp_use:N \g_xeCJK_monoscale_fp } }
1344     }
1345 }

```

(End definition for AutoFakeBold, AutoFakeSlant, and Mono.)

\xeCJK_set_init:

```

1346 \cs_new_nopar:Nn \xeCJK_set_init:
1347 {
1348     \tl_clear:N \l_xeCJK_monoscale_tl
1349     \int_gincr:N \g_xeCJK_fontspec_int
1350     \prop_clear:N \l_xeCJK_add_font_prop
1351     \prop_map_inline:Nn \g_xeCJK_features_id_prop
1352     {
1353         \tl_clear:c { l_xeCJK_fontname_##1_tl }
1354         \clist_clear:c { l_xeCJK_fontfeat_##1_clist }
1355     }
1356     \clist_clear:N \l_xeCJK_pass_features_clist
1357     \clist_clear:N \l_xeCJK_sub_key_clist
1358     \clist_map_inline:Nn \g_xeCJK_sub_key_clist
1359     {
1360         \bool_set_false:c { l_xeCJK_copy_##1_bool }
1361         \bool_set_false:c { l_xeCJK_add_##1_bool }
1362     }
1363     \bool_set_eq:NN \l_xeCJK_AutoFakeBold_bool \g_xeCJK_AutoFakeBold_bool
1364     \bool_set_eq:NN \l_xeCJK_AutoFakeSlant_bool \g_xeCJK_AutoFakeSlant_bool
1365     \fp_set_eq:NN \l_xeCJK_EmboldenFactor_fp \g_xeCJK_EmboldenFactor_fp
1366     \fp_set_eq:NN \l_xeCJK_SlantFactor_fp \g_xeCJK_SlantFactor_fp
1367 }

```

(End definition for \xeCJK_set_init:.)

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

```

1368 \cs_new_protected_nopar:Nn \xeCJK_set_family:nnn
1369 {
1370     \group_begin:
1371     \xeCJK_set_init:
1372     \tl_set:Nx \l_xeCJK_family_tl {#1}
1373     \clist_set:Nx \l_xeCJK_fontoptions_clist {#2}
1374     \tl_set:Nx \l_xeCJK_fontname_tl {#3}
1375     \keys_set_known:nxN { xeCJK / features }
1376     { \g_xeCJK_default_features_clist, #2 } \l_xeCJK_pass_features_clist
1377     \xeCJK_parse_features:
1378     \xeCJK_pass_features:
1379     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1380     {
1381         \exp_not:n { \fontspec_set_family:Nnn \l_xeCJK_fontspec_family_tl }
1382         { \l_xeCJK_pass_features_clist } { \l_xeCJK_fontname_tl }
1383         \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop

```

```

1384         { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_fontspec_family_tl
1385     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1386     \exp_not:n
1387     { {
1388         \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1389         \exp_not:N \fontfamily { \l_xeCJK_fontspec_family_tl }
1390         \exp_not:N \selectfont
1391     } }
1392     }
1393     \xeCJK_save_family_info:
1394     \xeCJK_add_sub_block:
1395     \group_end:
1396     }
1397     \cs_generate_variant:Nn \xeCJK_set_family:nnn { ncc }
1398     (End definition for \xeCJK_set_family:nnn.)

```

\xeCJK_add_font:nn

\xeCJK_add_font_if_new:nn

```

1398 \cs_new_nopar:Nn \xeCJK_add_font:nn
1399 {
1400     \prop_put:Nxx \l_xeCJK_add_font_prop
1401     { \xeCJK_map_features_id:n {#1} Font } {#2}
1402 }
1403 \cs_new_nopar:Nn \xeCJK_add_font_if_new:nn
1404 {
1405     \prop_put_if_new:Nxx \l_xeCJK_add_font_prop
1406     { \xeCJK_map_features_id:n {#1} Font } {#2}
1407 }
1408 \cs_generate_variant:Nn \xeCJK_add_font:nn { nx, nc, nV, nv }
1409 \cs_generate_variant:Nn \xeCJK_add_font_if_new:nn { nx, nc, nV, nv }
1410 (End definition for \xeCJK_add_font:nn and \xeCJK_add_font_if_new:nn.)
1410 \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1411 \cs_generate_variant:Nn \prop_get:Nn { NV }
1412 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1413 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
1414 \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1415 \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

\xeCJK_add_fake_bold:n

\xeCJK_add_fake_slant:n

```

1416 \cs_new_nopar:Nn \xeCJK_add_fake_bold:n
1417 {
1418     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1419     { FakeBold = { \fp_use:N \l_xeCJK_EmboldenFactor_fp } }
1420 }
1421 \cs_new_nopar:Nn \xeCJK_add_fake_slant:n
1422 {
1423     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1424     { FakeSlant = { \fp_use:N \l_xeCJK_SlantFactor_fp } }
1425 }
1426 (End definition for \xeCJK_add_fake_bold:n and \xeCJK_add_fake_slant:n.)

```

\xeCJK_parse_features:

\xeCJK_parse_features:n

```

1426 \cs_new_nopar:Nn \xeCJK_parse_features:
1427 { \prop_map_inline:Nn \g_xeCJK_features_id_prop { \xeCJK_parse_features:n {##1} } }
1428 \cs_new_nopar:Nn \xeCJK_parse_features:n
1429 {
1430     \xeCJK_if_font_select:nTF {#1}
1431     {

```

```

1432 \xeCJK_add_font:nv {#1} { l_xeCJK_fontname_#1_tl }
1433 \xeCJK_if_it_or_sl:nTF {#1}
1434 {
1435   \xeCJK_if_font_select:nF {bf#1}
1436   { \xeCJK_add_font_if_new:nv {bf#1} { l_xeCJK_fontname_#1_tl } }
1437 }
1438 {
1439   \str_if_eq:xxT {#1} {bf}
1440   {
1441     \clist_map_inline:nn { it , sl }
1442     {
1443       \xeCJK_if_font_select:nF {bf##1}
1444       { \xeCJK_add_font:nv {bf##1} \l_xeCJK_fontname_bf_tl }
1445     }
1446   }
1447 }
1448 }
1449 { \xeCJK_set_fake:n {#1} }
1450 }

```

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

```

\xeCJK_if_font_select_p:n
  \xeCJK_if_it_or_sl_p:n 1451 \prg_new_conditional:Nnn \xeCJK_if_font_select:n { p, T, F, TF }
\xeCJK_if_font_select:nTF 1452 {
  \xeCJK_if_it_or_sl:nTF 1453   \exp_args:Nc
1454   \tl_if_blank:xTF { l_xeCJK_fontname_#1_tl } \prg_return_false: \prg_return_true:
1455 }
1456 \prg_new_conditional:Nnn \xeCJK_if_it_or_sl:n { p, T, F, TF }
1457 {
1458   \bool_if:nTF { \str_if_eq_p:xx { it } {#1} || \str_if_eq_p:xx { sl } {#1} }
1459   \prg_return_true: \prg_return_false:
1460 }

```

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

\xeCJK_set_fake:n

```

1461 \cs_new_nopar:Nn \xeCJK_set_fake:n
1462 {
1463   \str_if_eq:xxTF {#1} { bf }
1464   { \bool_if:NT \l_xeCJK_AutoFakeBold_bool { \xeCJK_add_fake_bold:n {#1} } }
1465   {
1466     \bool_if:NTF \l_xeCJK_AutoFakeSlant_bool
1467     {
1468       \bool_if:nT
1469       { \xeCJK_if_it_or_sl_p:n {#1} ||
1470         ( \str_if_eq_p:xx {#1} {bfit} && ! ( \xeCJK_if_font_select_p:n {it} ) ) ||
1471         ( \str_if_eq_p:xx {#1} {bfsf} && ! ( \xeCJK_if_font_select_p:n {sl} ) )
1472       }
1473       { \xeCJK_add_fake_slant:n {#1} }
1474     }
1475     { \xeCJK_if_it_or_sl:nT {#1} { \xeCJK_map_it_sl:n {#1} } }
1476     \bool_if:nT
1477     { \l_xeCJK_AutoFakeBold_bool
1478       && ! ( \xeCJK_if_it_or_sl_p:n {#1} )
1479       && ! ( \xeCJK_if_font_select_p:n {bf} )
1480     }
1481     { \xeCJK_add_fake_bold:n {#1} }
1482   }
1483   \xeCJK_add_font_if_new:nn {#1} *

```

```

1484 }
      (End definition for \xeCJK_set_fake:n.)

```

\xeCJK_map_it_sl:n

```

1485 \cs_new_nopar:Nn \xeCJK_map_it_sl:n
1486 {
1487   \xeCJK_if_map_font_select:nT {#1}
1488   {
1489     \xeCJK_add_font:nx {#1} { \xeCJK_get_map_font:n {#1} }
1490     \xeCJK_if_font_select:nF {bf#1}
1491     { \xeCJK_add_font_if_new:nx {bf#1} { \xeCJK_get_map_font:n {#1} } }
1492   }
1493 }
      (End definition for \xeCJK_map_it_sl:n.)

```

\xeCJK_get_map_font:n

```

1494 \cs_new_nopar:Nn \xeCJK_get_map_font:n
1495 { \tl_use:c { l_xeCJK_fontname_\str_if_eq:xxTF {#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
1496 \prg_new_conditional:Nnn \xeCJK_if_map_font_select:n { p, T, F, TF }
1497 {
1498   \tl_if_blank:xTF { \xeCJK_get_map_font:n {#1} }
1499   \prg_return_false: \prg_return_true:
1500 }
      (End definition for \xeCJK_if_map_font_select:n.)

```

\xeCJK_pass_features:

```

1501 \cs_new_nopar:Nn \xeCJK_pass_features:
1502 {
1503   \prop_map_inline:Nn \g_xeCJK_features_id_prop
1504   {
1505     \clist_if_empty:cF { l_xeCJK_fontfeat_##1_clist }
1506     {
1507       \clist_put_right:Nx \l_xeCJK_pass_features_clist
1508       { ##2Features = { \clist_use:c { l_xeCJK_fontfeat_##1_clist } } }
1509     }
1510   }
1511   \prop_map_inline:Nn \l_xeCJK_add_font_prop
1512   { \clist_put_right:Nx \l_xeCJK_pass_features_clist { ##1 = { ##2 } } }
1513   \tl_if_blank:xF \l_xeCJK_monoscale_tl
1514   { \clist_put_right:Nx \l_xeCJK_pass_features_clist \l_xeCJK_monoscale_tl }
1515 }
      (End definition for \xeCJK_pass_features:.)

```

\g_xeCJK_family_prop

\g_xeCJK_family_name_prop

\g_xeCJK_family_options_prop

```

1516 \prop_new:N \g_xeCJK_family_prop
1517 \prop_new:N \g_xeCJK_family_name_prop
1518 \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:

```
1519 \cs_new_nopar:Nn \xeCJK_save_family_info:
1520 {
1521   \group_begin:
1522   \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1523   {
1524     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist {##1}
1525     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist { ##1 = }
1526   }
1527   \prop_gput:NVV \g_xeCJK_family_prop
1528     \l_xeCJK_family_tl \l_xeCJK_fontname_tl
1529   \prop_gput:NVV \g_xeCJK_family_options_prop
1530     \l_xeCJK_family_tl \l_xeCJK_fontoptions_clist
1531   \group_end:
1532 }
(End definition for \xeCJK_save_family_info:.)
```

\xeCJK_add_sub_block:

```
1533 \cs_new_nopar:Nn \xeCJK_add_sub_block:
1534 {
1535   \clist_remove_duplicates:N \l_xeCJK_sub_key_clist
1536   \clist_map_inline:Nn \l_xeCJK_sub_key_clist
1537   {
1538     \tl_set:Nx \l_tmpa_tl { \l_xeCJK_family_tl/##1 }
1539     \bool_if:cT { l_xeCJK_copy_##1_bool }
1540     {
1541       \tl_map_inline:nn { \g_xeCJK_family_prop \g_xeCJK_family_options_prop }
1542       {
1543         \prop_get:NVNT #####1 \l_xeCJK_family_tl \l_tmpb_tl
1544         { \prop_gput:NVV #####1 \l_tmpa_tl \l_tmpb_tl }
1545       }
1546       \prop_get:NVNT \g_xeCJK_family_options_prop \l_tmpa_tl \l_tmpa_clist
1547       {
1548         \clist_remove_all:Nn \l_tmpa_clist { ##1 = * }
1549         \prop_gput:NVV \g_xeCJK_family_options_prop \l_tmpa_tl \l_tmpa_clist
1550       }
1551       \cs_gset_protected_nopar:cpx { xeCJK/family/\l_tmpa_tl }
1552       {
1553         \exp_not:N \xeCJK_family_if_exist:nT { \l_xeCJK_family_tl }
1554         {
1555           \prop_get:NnNT \exp_not:N \g_xeCJK_family_name_prop
1556             { \l_xeCJK_family_tl } \exp_not:N \l_tmpa_tl
1557           {
1558             \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1559               { \l_tmpa_tl } \exp_not:N \l_tmpa_tl
1560             \cs_gset_protected_nopar:cpx { xeCJK/family/\l_tmpa_tl }
1561             \exp_not:n
1562             { {
1563               \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1564               \exp_not:N \fontfamily { \l_tmpa_tl }
1565               \exp_not:N \selectfont
1566             } }
1567           }
1568         }
1569       }
1570     }
1571     \bool_if:cT { l_xeCJK_add_##1_bool }
1572     {
```

```

1573         \xeCJK_set_family:ncc \l_tmpa_tl
1574         { l_xeCJK_##1_fontfeat_clist } { l_xeCJK_##1_fontname_tl }
1575     }
1576 }
1577 }
(End definition for \xeCJK_add_sub_block:.)

```

\xeCJK_copy_family:nn

```

1578 \cs_new_nopar:Nn \xeCJK_copy_family:nn
1579 {
1580     \xeCJK_family_if_exist:nT {#2}
1581     {
1582         \tl_map_inline:nn
1583         { \g_xeCJK_family_prop \g_xeCJK_family_options_prop \g_xeCJK_family_name_prop }
1584         { \prop_get:NnNT ##1 {#2} \l_tmpa_tl { \prop_gput:NnV ##1 {#1} \l_tmpa_tl } }
1585         \cs_gset_eq:cc { xeCJK/family/#1 } { xeCJK/family/#2 }
1586     }
1587 }
1588 \cs_generate_variant:Nn \xeCJK_copy_family:nn { xx }
(End definition for \xeCJK_copy_family:nn.)

```

4.8 字体切换

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

```

\l_xeCJK_font_coor_tl
\l_xeCJK_select_font:
\l_xeCJK_switch_font:nn
1589 \tl_set:Nn \l_xeCJK_font_coor_tl { xeCJK/\xeCJK@family/\f@series/\f@shape/\f@size }
1590 \cs_new_nopar:Nn \xeCJK_select_font:
1591 {
1592     \cs_if_exist_use:cF \l_xeCJK_font_coor_tl
1593     {
1594         \tl_set:Nx \l_xeCJK_current_coor_tl \l_xeCJK_font_coor_tl
1595         \cs_if_exist_use:cT { xeCJK/family/\xeCJK@family }
1596         { \exp_last_unbraced:NNV \cs_gset_eq:cN \l_xeCJK_current_coor_tl \font@name }
1597     }
1598 }
1599 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
1600 \cs_new_nopar:Nn \xeCJK_switch_font:nn
1601 {
1602     \str_if_eq:xxF {#1} {#2}
1603     {
1604         \xeCJK_info:nxx { CJK-block } {#1} {#2}
1605         \tl_remove_all:Nn \xeCJK@family { /#1 }
1606         \str_if_eq:xxF {#2} { CJK }
1607         {
1608             \tl_set:Nx \l_tmpa_tl \xeCJK@family
1609             \tl_put_right:Nx \xeCJK@family { /#2 }
1610             \xeCJK_family_if_exist:nF \xeCJK@family
1611             {
1612                 \xeCJK_family_if_exist:nTF { \CJKfamilydefault/#2 }
1613                 { \xeCJK_copy_family:xx \xeCJK@family { \CJKfamilydefault/#2 } }
1614                 { \xeCJK_copy_family:xx \xeCJK@family \l_tmpa_tl }
1615             }
1616         }
1617     }
1618     \xeCJK_select_font:
1619 }
1620 \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:nTF

```
1621 \prg_new_conditional:Nnn \xeCJK_family_if_exist:n { T, F, TF }
1622 {
1623   \cs_if_free:cTF { xeCJK/family/#1 } \prg_return_false:
1624   {
1625     \prop_if_in:NxF
1626     \g_xeCJK_family_name_prop {#1} { \use:c { xeCJK/family/#1 } }
1627     \prg_return_true:
1628   }
1629 }
(End definition for \xeCJK_family_if_exist:nTF.)
```

\CJKfamily 用于切换 CJK 字体族。

```
1630 \NewDocumentCommand \CJKfamily { m }
1631 {
1632   \xeCJK_family_if_exist:nTF {#1}
1633   { \tl_set:Nx \xeCJK@family {#1} }
1634   {
1635     \cs_if_free:cT { xeCJK_warned_aux_#1 }
1636     {
1637       \xeCJK_warning:nx { CJKfamily-Unknown } {#1}
1638       \tl_new:c { xeCJK_warned_aux_#1 }
1639     }
1640   }
1641 }
1642 \xeCJK_msg_new:nn { CJKfamily-Unknown }
1643 {
1644   Unknown~CJK~family~'#1'~is~ignored.\\\
1645   Try~to~use~
1646   \prg_case_str:xxn {#1}
1647   {
1648     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1649     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1650     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1651   }
1652   { \token_to_str:N \setCJKfamilyfont {#1} }
1653   [...]{}~to~define~it.
1654 }
```

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

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

```
\newCJKfontfamily
\CJKfontspec
1655 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
1656 { \xeCJK_set_family:nnn {#1} {#2} {#3} }
1657 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1658 {
1659   \IfNoValueTF {#1}
1660   { \tl_set:Nx \l_xeCJK_family_tl { \cs_to_str:N #2 } }
1661   { \tl_set:Nx \l_xeCJK_family_tl {#1} }
1662   \xeCJK_set_family:nnn \l_xeCJK_family_tl {#3} {#4}
1663   \cs_new_protected_nopar:Npx #2
1664   { \exp_not:N \CJKfamily { \l_xeCJK_family_tl } }
1665 }
1666 \int_new:N \g_xeCJK_fonts_spec_int
1667 \int_gdecr:N \g_xeCJK_fonts_spec_int
1668 \NewDocumentCommand \CJKfontspec { O{} m }
1669 {
1670   \tl_set:Nx \l_xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1671   \cs_if_free:cT \l_xeCJK_CJKfontspec_id_tl
```

```

1672     {
1673       \int_gincr:N \g_xeCJK_fontspec_int
1674       \tl_gset:Nx \g_xeCJK_family_spec_tl
1675       { CJKfontspec ( \int_use:N \g_xeCJK_fontspec_int ) }
1676       \xeCJK_set_family:nnn \g_xeCJK_family_spec_tl {#1} {#2}
1677       \tl_gset:cx \l_xeCJK_CJKfontspec_id_tl \g_xeCJK_family_spec_tl
1678     }
1679     \exp_args:Nv \CJKfamily \l_xeCJK_CJKfontspec_id_tl
1680     \tex_ignorespaces:D
1681   }

```

(End definition for \setCJKfamilyfont, \newCJKfontfamily, and \CJKfontspec. These functions are documented on page 5.)

\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
1682 \clist_new:N \g_xeCJK_default_features_clist
1683 \NewDocumentCommand \defaultCJKfontfeatures { m }
1684 { \clist_gset:Nn \g_xeCJK_default_features_clist {#1} }
1685 \@onlypreamble \defaultCJKfontfeatures
1686 \NewDocumentCommand \addCJKfontfeatures { m }
1687 {
1688   \prop_if_in:NVTF \g_xeCJK_family_prop \xeCJK@family
1689   {
1690     \group_begin:
1691     \clist_set:Nx \l_tmpa_clist {#1}
1692     \prop_get:NVN \g_xeCJK_family_prop \xeCJK@family \l_tmpa_tl
1693     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1694     {
1695       \clist_if_in:NnT \l_tmpa_clist {##1}
1696       {
1697         \clist_remove_all:Nn \l_tmpa_clist {##1}
1698         \prop_get:NxNF \g_xeCJK_family_prop
1699         { \xeCJK@family/##1 } \l_tmpb_tl
1700         {
1701           \prop_get:NxNF \g_xeCJK_family_prop
1702           { \CJKfamilydefault/##1 } \l_tmpb_tl
1703           { \tl_set:Nx \l_tmpb_tl \l_tmpa_tl }
1704         }
1705         \clist_set:Nx \l_tmpa_clist
1706         { ##1 = { [ \l_tmpa_clist ] { \l_tmpb_tl } } }
1707         \clist_map_break:
1708       }
1709     }
1710     \prop_get:NVNF \g_xeCJK_family_options_prop \xeCJK@family
1711     \l_tmpb_clist { \clist_clear:N \l_tmpb_clist }
1712     \clist_clear:N \l_tmpc_clist
1713     \clist_map_inline:Nn \l_tmpb_clist
1714     {
1715       \clist_if_in:NnF \l_tmpa_clist {##1}
1716       { \clist_put_right:Nn \l_tmpc_clist {##1} }
1717     }
1718     \clist_put_left:Nv \l_tmpa_clist \l_tmpc_clist
1719     \use:x
1720     { \group_end: \exp_not:N \CJKfontspec [ \l_tmpa_clist ] { \l_tmpa_tl } }
1721   }
1722   { \xeCJK_warning:n { addCJKfontfeature-ignored } }
1723   \tex_ignorespaces:D
1724 }
1725 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

```

1726 \xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1727 {
1728   \tl_to_str:N \addCJKfontfeatures (s)~ignored.\\\
1729   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1730 }
(End definition for \defaultCJKfontfeatures and \addCJKfontfeatures. These functions are documented on page 5.)

```

\setCJKmainfont 设置文档的 CJK 普通字体、无衬线字体。

```

\setCJKsansfont 1731 \NewDocumentCommand \setCJKmainfont { O{ } m }
1732 { \xeCJK_set_family:nnn \CJKrmdefault {#1} {#2} }
1733 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1734 \NewDocumentCommand \setCJKsansfont { O{ } m }
1735 { \xeCJK_set_family:nnn \CJKsfdefault {#1} {#2} }
(End definition for \setCJKmainfont and \setCJKsansfont. These functions are documented on page 4.)

```

```

\CJKrmdefault
\CJKsfdefault 1736 \cs_if_free:NT \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
\CJKttdefault 1737 \cs_if_free:NT \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 1738 \cs_if_free:NT \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1739 \cs_if_free:NT \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1740 \tl_new:c { xeCJK/family/\CJKfamilydefault }
(End definition for \CJKrmdefault and others. These functions are documented on page 5.)

```

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

```

1741 \AtEndOfPackage
1742 {
1743   \etex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1744   \etex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }
1745   \etex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1746   \etex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1747   \cs_set_eq:NN \reset@font \normalfont
1748 }

```

\setCJKmathfont 设置 CJK 数学字体。

```

1749 \NewDocumentCommand \setCJKmathfont { O{ } m }
1750 { \xeCJK_set_family:nnn \c_xeCJK_math_tl {#1} {#2} }
(End definition for \setCJKmathfont. This function is documented on page 5.)

```

CJKmath 数学公式内的 CJK 字体。

```

1751 \tl_const:Nn \c_xeCJK_math_tl { CJKmath }
1752 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g_xeCJK_math_bool }
(End definition for CJKmath. This function is documented on page 2.)

```

\xeCJKsetmathcode

```

1753 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1754 {
1755   \xeCJK_check_num_range:nnNN {#1} {#2} \l_tmpa_int \l_tmpb_int
1756   \int_set:Nn \l_tmpc_int { \xeCJK_math_type:n {#3} }
1757   \loop
1758     \XeTeXmathcode \l_tmpa_int = \l_tmpc_int \use:c { sym #4 } \l_tmpa_int
1759     \if_int_compare:w \l_tmpa_int < \l_tmpb_int
1760       \int_incr:N \l_tmpa_int
1761   \repeat
1762 }
1763 \cs_set_eq:NN \xeCJK_math_type:n \mathchar@type

```

(End definition for \xeCJKsetmathcode.)

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

```
1764 \tl_put_left:Nn \document
1765 {
1766   \group_end:
1767   \prop_if_empty:NTF \g_xeCJK_family_prop
1768   { \xeCJK_warning:n { no-CJKfamily } }
1769   {
1770     \bool_if:NT \g_xeCJK_math_bool
1771     {
1772       \xeCJK_family_if_exist:nTF \c_xeCJK_math_tl
1773       { \tl_set:Nx \l_tmpa_tl \c_xeCJK_math_tl }
1774       {
1775         \xeCJK_family_if_exist:nTF \CJKfamilydefault
1776         { \tl_set:Nx \l_tmpa_tl \CJKfamilydefault }
1777         { \tl_set_eq:NN \l_tmpa_tl \q_no_value }
1778       }
1779       \prop_get:NVNT \g_xeCJK_family_name_prop \l_tmpa_tl \l_xeCJK_math_family_tl
1780       {
1781         \DeclareSymbolFont \c_xeCJK_math_tl \g_fontspec_encoding_tl
1782         \l_xeCJK_math_family_tl \mddefault \shapedefault
1783         \SetSymbolFont \c_xeCJK_math_tl { bold } \g_fontspec_encoding_tl
1784         \l_xeCJK_math_family_tl \bfdefault \shapedefault
1785         \clist_map_inline:nn { CJK, FullLeft, FullRight }
1786         {
1787           \clist_map_inline:cn { g_xeCJK_#1_range_clist }
1788           {
1789             \xeCJK_setcharclass_aux:Nn \xeCJKsetmathcode {##1}
1790             \mathalpha \c_xeCJK_math_tl
1791           }
1792         }
1793       }
1794     }
1795     \normalfont
1796   }
1797   \group_begin:
1798 }
1799 \xeCJK_msg_new:nn { no-CJKfamily }
1800 {
1801   It~seems~that~you~have~not~declare~a~CJKfamily.\\
1802   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
1803   \token_to_str:N~\xeCJKmainfont[...]{...}\\
1804   in~the~preamble~to~declare~the~main~CJKfamily.\\
1805 }
```

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。

```
1806 \@onlypreamble \setCJKmainfont
1807 \@onlypreamble \setCJKmathfont
1808 \@onlypreamble \setCJKsansfont
1809 \@onlypreamble \setCJKmonofont
1810 \@onlypreamble \setCJKromanfont
```

4.9 处理等宽字体和抄录环境

```
\g_xeCJK_monoscale_fp
\g_xeCJK_exspace_dim
1811 \fp_new:N \g_xeCJK_monoscale_fp
1812 \fp_gset_eq:NN \g_xeCJK_monoscale_fp \c_one_fp
1813 \dim_new:N \g_xeCJK_exspace_dim
```

(End definition for \g_xeCJK_monoscale_fp and \g_xeCJK_exspace_dim.)

\setCJKmonoscale

```

1814 \NewDocumentCommand \setCJKmonoscale { }
1815 {
1816   \xeCJK_set_monoscale:
1817   \addCJKfontfeatures { Scale = \fp_use:N \g_xeCJK_monoscale_fp }
1818 }
1819 \cs_new_nopar:Nn \xeCJK_set_monoscale:
1820 {
1821   \dim_gzero:N \g_xeCJK_exspace_dim
1822   \group_begin:
1823   \fontfamily \ttdefault \selectfont
1824   \fp_gset_from_dim:Nn \g_xeCJK_monoscale_fp
1825   { \c_two \tex_fontdimen:D \c_two \tex_font:D }
1826   \fp_gdiv:Nn \g_xeCJK_monoscale_fp \f@size
1827   \group_end:
1828 }

```

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

\setCJKmonoexspace

```

1829 \NewDocumentCommand \setCJKmonoexspace { } { \xeCJK_set_monoexspace: }
1830 \cs_new_nopar:Nn \xeCJK_set_monoexspace:
1831 {
1832   \fp_gset_eq:NN \g_xeCJK_monoscale_fp \c_one_fp
1833   \group_begin:
1834   \fontfamily \ttdefault \selectfont
1835   \dim_gset:Nn \g_xeCJK_exspace_dim
1836   { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size \p@ }
1837   \group_end:
1838 }

```

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

\xeCJK_fixed_ecglue:

\xeCJK_fixed_cjkg glue:

\xeCJK_flexible_ecglue:

\xeCJK_flexible_cjkg glue:

\xeCJK_flexible_cjkg glue:

```

1839 \cs_new_nopar:Nn \xeCJK_fixed_ecglue: { \skip_horizontal:n { .5\g_xeCJK_exspace_dim } }
1840 \cs_new_nopar:Nn \xeCJK_fixed_cjkg glue: { \skip_horizontal:N \g_xeCJK_exspace_dim }
1841 \AtEndOfPackage
1842 {
1843   \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1844   \cs_new_eq:NN \xeCJK_flexible_ecglue: \CJKecglue
1845   \cs_new_eq:NN \xeCJK_flexible_cjkg glue: \CJKglue
1846 }

```

(End definition for \xeCJK_fixed_ecglue: and others.)

\CJKfixedspacing

```

1847 \NewDocumentCommand \CJKfixedspacing { }
1848 {
1849   \tl_if_eq:NNF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1850   {
1851     \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1852     \punctstyle { plain }
1853   }
1854   \cs_if_eq:NNF \CJKecglue \xeCJK_fixed_ecglue:
1855   {
1856     \cs_set_eq:NN \xeCJK_flexible_ecglue: \CJKecglue
1857     \cs_set_eq:NN \CJKecglue \xeCJK_fixed_ecglue:
1858   }

```

```

1859 \cs_if_eq:NNF \CJKglue \xeCJK_fixed_cjkglue:
1860 {
1861   \cs_set_eq:NN \xeCJK_flexible_cjkglue: \CJKglue
1862   \cs_set_eq:NN \CJKglue \xeCJK_fixed_cjkglue:
1863 }
1864 }
1865 \AtBeginDocument { \tl_put_right:Nn \verbatim@font \CJKfixedspacing }
(End definition for \CJKfixedspacing. This function is documented on page 4.)

```

\CJKflexiblespacing

```

1866 \NewDocumentCommand \CJKflexiblespacing { m }
1867 {
1868   \cs_set_eq:NN \l_xeCJK_punctstyle_tl \l_xeCJK_flexible_punctstyle_tl
1869   \cs_set_eq:NN \CJKecglue \xeCJK_flexible_ecglue:
1870   \cs_set_eq:NN \CJKglue \xeCJK_flexible_cjkglue:
1871 }
(End definition for \CJKflexiblespacing. This function is documented on page 4.)

```

\setCJKmonofont 设置文档的 CJK 等宽字体族。

```

1872 \NewDocumentCommand \setCJKmonofont { s t+ O{} m }
1873 {
1874   \IfBooleanTF {#1}
1875   { \xeCJK_set_family:nnn \CJKttdefault { Mono = Scale, #3 } {#4} }
1876   {
1877     \IfBooleanTF {#2}
1878     { \xeCJK_set_family:nnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
1879     { \xeCJK_set_family:nnn \CJKttdefault {#3} {#4} }
1880   }
1881 }
(End definition for \setCJKmonofont. This function is documented on page 4.)

```

4.10 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

```

1882 \bool_new:N \g_xeCJK_indent_bool
1883 \keys_define:nn { xeCJK / options }
1884 {
1885   CJKnumber      .bool_set:N = \g_xeCJK_number_bool ,
1886   indentfirst    .bool_set:N = \g_xeCJK_indent_bool ,
1887   normalindentfirst .meta:n = { indentfirst = true } ,
1888   quiet .code:n =
1889   {
1890     \msg_redirect_module:nnn { xeCJK } { warning } { info }
1891     \msg_redirect_module:nnn { xeCJK } { info } { none }
1892     \PassOptionsToPackage { quiet } { fontspec }
1893   },
1894   silent .code:n =
1895   {
1896     \msg_redirect_module:nnn { xeCJK } { warning } { none }
1897     \msg_redirect_module:nnn { xeCJK } { info } { none }
1898     \PassOptionsToPackage { silent } { fontspec }
1899   },
1900   unknown .code:n =
1901   {
1902     \@ifpackageloaded { fontspec }
1903     { \xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }

```

```

1904         { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
1905     },
1906 }
1907 \xeCJK_msg_new:nn { key-unknown }
1908 {
1909     Sorry,~but~\l_keys_module_tl\c_space_tl does~not~have~a~key~called~'#1'.\\\
1910     The~key~'#1'~is~being~ignored.
1911 }

```

(End definition for CJKnumber and indentfirst. These functions are documented on page 2.)

4.11 xeCJK 初始化设置

xeCJK 宏包的初始化设置。

```

1912 \keys_set:nn { xeCJK / options }
1913 {
1914     CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
1915     CJKecglue = \c_space_token ,
1916     CheckSingleglue = \CJKglue ,
1917     xCJKecglue = false ,
1918     CheckSingle = false ,
1919     CJKspace = false ,
1920     CJKmath = false ,
1921     xeCJKactive = true ,
1922     indentfirst = true ,
1923     EmboldenFactor = 4 ,
1924     SlantFactor = .167 ,
1925     PunctStyle = quanjiao ,
1926     KaiMingPunct = { . 。 ? ! } ,
1927     LongPunct = { — … — } ,
1928     MiddlePunct = { · · · } ,
1929     AllowBreakBetweenPuncts = false ,
1930 }

```

\CJKsymbol

```

\CJKpunctsymbol 1931 \cs_new_nopar:Npn \CJKsymbol #1 {#1}
1932 \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。

```

1933 \ProcessKeysOptions { xeCJK / options }
1934 \RequirePackage { fontspec } [ 2011/09/13 ]
1935 \bool_if:NT \g_xeCJK_indent_bool
1936 {
1937     \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
1938     \afterindenttrue
1939 }

```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```

1940 \NewDocumentCommand \xeCJKsetup { m }
1941 {
1942     \keys_set:nn { xeCJK / options } {#1}
1943     \tex_ignorespaces:D
1944 }

```

(End definition for \xeCJKsetup. This function is documented on page 1.)

```

\XeCJKsetemboldenfactor
\XeCJKsetslantfactor
1945 \NewDocumentCommand \XeCJKsetEmboldenFactor { m }
1946 { \XeCJKsetup { EmboldenFactor = {#1} } }
1947 \NewDocumentCommand \XeCJKsetSlantFactor { m }
1948 { \XeCJKsetup { SlantFactor = {#1} } }
(End definition for \XeCJKsetemboldenfactor and \XeCJKsetslantfactor.)

\punctstyle
\XeCJKplainchr
1949 \NewDocumentCommand \punctstyle { m } { \XeCJKsetup { PunctStyle = {#1} } }
1950 \cs_new_nopar:Npn \XeCJKplainchr { \punctstyle { plain } }
(End definition for \punctstyle and \XeCJKplainchr.)

\CJKsetecglue
1951 \NewDocumentCommand \CJKsetecglue { m } { \XeCJKsetup { CJKecglue = {#1} } }
1952 \cs_new_eq:NN \XeCJKsetecglue \CJKsetecglue
(End definition for \CJKsetecglue.)

```

4.12 兼容性修补

∨ 修复倾斜校正。

```

1953 \cs_new_eq:NN \XeCJK_italiccorr_aux \/
1954 \bool_new:N \g_xeCJK_italiccorr_bool
1955 \cs_set_protected_nopar:Npn \/
1956 {
1957   \scan_stop:
1958   \int_compare:nNt \tex_lastkern:D = \c_three
1959   {
1960     \tex_unkern:D \tex_unkern:D
1961     \bool_gset_true:N \g_xeCJK_italiccorr_bool
1962   }
1963   \XeCJK_italiccorr_aux
1964 }
1965 \cs_set_eq:NN \@italiccorr \/
(End definition for \/. )

```

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

```

\XeCJK_set_others_toks:n
\XeCJK_Others_class
1966 \cs_new_nopar:Nn \XeCJK_compatibility_hook:
1967 {
1968   \int_compare:nNf
1969   { \c_three + \seq_length:N \g_xeCJK_class_seq } = \xe@alloc@intercharclass
1970   {
1971     \int_set_eq:NN \l_tmpa_int \c_three
1972     \int_until_do:nNnn \l_tmpa_int = \xe@alloc@intercharclass
1973     {
1974       \int_incr:N \l_tmpa_int
1975       \seq_if_in:NVF \g_xeCJK_class_seq \l_tmpa_int
1976       { \XeCJK_set_others_toks:n \l_tmpa_int }
1977     }
1978   }
1979 }
1980 \cs_new_nopar:Nn \XeCJK_set_others_toks:n
1981 {
1982   \int_set:Nn \XeCJK_Others_class {#1}
1983   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
1984   {
1985     \XeCJK_copy_inter_class_toks:n {##1} { Others } {##1} { NormalSpace }
1986     \XeCJK_copy_inter_class_toks:n { Others } {##1} { NormalSpace } {##1}

```

```

1987 \exp_args:Nnnx \xeCJK_app_inter_class_toks:nnn {##1} { Others }
1988 { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
1989 \exp_args:Nnnx \xeCJK_pre_inter_class_toks:nnn { Others } {##1}
1990 { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
1991 \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
1992 {
1993   \xeCJK_copy_inter_class_toks:nnnn
1994   { Others } { Boundary } { Default } { Boundary }
1995 }
1996 \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
1997 {
1998   \xeCJK_copy_inter_class_toks:nnnn
1999   { Boundary } { Others } { Boundary } { Default }
2000 }
2001 }
2002 }
2003 \int_new:N \xeCJK_Others_class
2004 \tl_put_right:Nn \document { \xeCJK_compatibility_hook: \tex_ignorespaces:D }
(End definition for \xeCJK_compatibility_hook:, \xeCJK_set_others_toks:n, and \xeCJK_Others_class.)

```

`\xeCJK_patch:Nnn` 给已有宏内容前后附加补丁。

```

2005 \cs_new_nopar:Nn \xeCJK_patch:Nnn
2006 { \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。

```

2007 \AtBeginDocument
2008 {
2009   \tl_map_inline:nn
2010   {
2011     \textellipsis \textendash \textperiodcentered \textcentereddot
2012     \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2013   }
2014   { \xeCJK_patch:Nnn #1 { \group_begin: \makexeCJKinactive } { \group_end: } }
2015   \xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2016   \cs_set_eq:NN \xeCJK_aux_r \r
2017   \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \xeCJK_aux_r {#1} } }
2018   \ifpackageloaded { pifont }
2019   {
2020     \RenewDocumentCommand \Pifont { m }
2021     {
2022       \fontencoding {U} \fontfamily {#1} \fontseries {m} \fontshape {n}
2023       \selectfont \makexeCJKinactive
2024     }
2025   } { }
2026 }

```

禁止在 `xeCJK` 宏包后再载入 `CJK` 宏包。

```

2027 \tl_set:cn { ver@CJK.sty } { 2050/01/01 }

```

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

```

2028 \cs_if_free:NT \CJK@ifundefined
2029 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2030 \NewDocumentCommand \xeCJKcaption { o m }
2031 {
2032   \IfValueT {#1} { \XeTeXdefaultencoding "#1" }
2033   \cs_set_nopar:Nx \xeCJK_reset_at_catcode:
2034   {

```

```

2035      \exp_not:n { \char_set_catcode:nn { '\@ } }
2036      { \char_value_catcode:n { '\@ } }
2037    }
2038    \char_set_catcode_letter:N \@
2039    \file_input:n { #2.cpx }
2040    \xeCJK_reset_at_catcode:
2041    \XeTeXdefaultencoding "UTF-8"
2042  }

```

(End definition for \xeCJKcaption.)

\xeCJK_ULprepunctchar:n

\xeCJK_ULpostpunctchar:n

\xeCJK_ULroutines:

```

2043 \cs_new_protected_nopar:Nn \xeCJK_ULprepunctchar:n
2044 {
2045   { \makexeCJKinactive \CJKpunctsymbol {#1} \nobreak }
2046   \tex_ignorespaces:D
2047 }
2048 \cs_new_protected_nopar:Nn \xeCJK_ULpostpunctchar:n
2049 {
2050   { \makexeCJKinactive \CJKpunctsymbol {#1} }
2051   \xeCJK_ignorespaces:
2052 }
2053 \cs_new_protected_nopar:Nn \xeCJK_ULroutines:
2054 {
2055   \xeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKecglue \CJKsymbol }
2056   \xeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2057   \xeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKecglue \CJKsymbol }
2058   \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }
2059   \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2060   {
2061     \xeCJK_inter_class_toks:nnn {##1} { FullLeft } { \xeCJK_ULprepunctchar:n }
2062     \xeCJK_inter_class_toks:nnn {##1} { FullRight } { \xeCJK_ULpostpunctchar:n }
2063   }
2064   \xeCJK_UL_subclass_patch_tl
2065 }

```

(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。

```

2066 \AtBeginDocument
2067 {
2068   \cs_if_exist:NT \UL@hook
2069   {
2070     \addto@hook \UL@hook
2071     {
2072       \cs_set_eq:NN \xeCJK_UL_CJKsymbol \CJKsymbol
2073       \cs_set_eq:NN \xeCJK_UL_CJKpunctsymbol \CJKpunctsymbol
2074       \cs_set_nopar:Npn \CJKsymbol #1
2075       {
2076         { \xeCJK_select_font: \xeCJK_UL_CJKsymbol {#1} }
2077         \xeCJK_CJK_kern: \xeCJK_ignorespaces:
2078       }
2079       \cs_set_nopar:Npn \CJKpunctsymbol #1
2080       { { \xeCJK_select_font: \xeCJK_UL_CJKpunctsymbol {#1} } }
2081       \xeCJK_ULroutines:
2082     }
2083   }
2084   \cs_if_exist:NT \XeTeX@CJKfntef@hook
2085   {
2086     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2087     { \xeCJK_select_font: \makexeCJKinactive }

```

```

2088     }
2089   }
    使用 CJKnumb 宏包。
2090 \bool_if:NT \g_xeCJK_number_bool
2091 {
2092   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2093   \cs_set_nopar:Npn \CJKaddEncHook #1#2 { \cs_set_nopar:cpn { xeCJK_enc_#1 } {#2} }
2094   \cs_set_nopar:Npn \Unicode #1#2
2095     { \tex_char:D \int_eval:n { #1 * \c_two_hundred_fifty_six + #2 } }
2096   \RequirePackage { CJKnumb }
2097   \use:c { xeCJK_enc_ \CJK@UnicodeEnc }
2098   \tl_set:Nn \CJK@tenthousand { 万 }
2099   \tl_set:Nn \CJK@hundredmillion { 亿 }
2100 }
2101 </package>

```

5 例子

5.1 example-autofake.tex

```

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

```

5.2 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 & 𠂇 & 3401 & 𠂈 & 3402 & 𠂉 & 3403 & \\\
73 𠂊 & 3404 & 𠂋 & 3405 & 𠂌 & 3406 & 𠂍 & 3407 & \\\
74 𠂎 & 3408 & 𠂏 & 3409 & 𠂐 & 340A & 𠂑 & 340B & \\\
75 𠂒 & 340C & 𠂓 & 340D & 𠂔 & 340E & 𠂕 & 341F & \\\
76 𠂖 & 3410 & 𠂗 & 3411 & 𠂘 & 3412 & 𠂙 & 3413 & \\\
77 𠂚 & 3414 & 𠂛 & 3415 & 𠂜 & 3416 & 𠂝 & 3417 & \\\
78 𠂞 & 3418 & 𠂟 & 3419 & 𠂠 & 341A & 𠂡 & 341B & \\\
79 𠂢 & 341C & 𠂣 & 341D & 𠂤 & 341E & 𠂥 & 341F & \\\[1ex]
80 𠂦 & 20000 & 𠂧 & 20001 & 𠂨 & 20002 & 𠂩 & 20003 & \\\
81 𠂪 & 20004 & 𠂫 & 20005 & 𠂬 & 20006 & 𠂭 & 20007 & \\\
82 𠂮 & 20008 & 𠂯 & 20009 & 𠂰 & 2000A & 𠂱 & 2000B & \\\
83 𠂲 & 2000C & 𠂳 & 2000D & 𠂴 & 2000E & 𠂵 & 2000F & \\\
84 𠂶 & 20010 & 𠂷 & 20011 & 𠂸 & 20012 & 𠂹 & 20013 & \\\
85 𠂺 & 20014 & 𠂻 & 20015 & 𠂼 & 20016 & 𠂽 & 20017 & \\\
86 𠂾 & 20018 & 𠂿 & 20019 & 𠃀 & 2001A & 𠃁 & 2001B & \\\
87 𠃂 & 2001C & 𠃃 & 2001D & 𠃄 & 2001E & 𠃅 & 2001F & \\\
88 \end{tabular}
89 \end{table}
90 \end{document}
91 \

```

5.3 example-SubCJKBlock.tex

```
92 <*ex-block>
93 \documentclass{article}
94 \usepackage{xCJK}
95 \usepackage{array}
96 \xeCJKDeclareSubCJKBlock{Ext-A} { "3400 -> "4DBF }
97 \xeCJKDeclareSubCJKBlock{Ext-B} { "20000 -> "2A6DF }
98 \xeCJKDeclareSubCJKBlock{Kana} { "3040 -> "309F, "30A0 -> "30FF, "31F0 -> "31FF, }
99 \xeCJKDeclareSubCJKBlock{Hangul}{ "1100 -> "11FF, "3130 -> "318F, "A960 -> "A97F, "AC00 -
> "D7AF }
100 \setCJKmainfont[Ext-A=SimHei,Ext-B=SimSun-ExtB]{SimSun}
101 \setCJKmainfont[Kana]{Meiryo}
102 \setCJKmainfont[Hangul]{Malgun Gothic}
103 \parindent=2em
104 \begin{document}
105 \long\def\showtext{%
106 中日韩越统一表意文字（英語：CJKV Unified Ideographs），旧称中日韩统一表意文字（英
語：CJK Unified Ideographs），也称统一汉字（英語：Unihan），目的是要把分别来自中
文、日文、韩文、越文、壮文中，对于相同起源、本义相同、形状一样或稍异的表意文字
（主要为汉字，但也有仿汉字如：方块壮字、日文汉字（かんじ / kanji）、韩文汉字（한
자 / hanja）、越南的喃字（Ch Nôm）与越文汉字（Ch Nho，在越南也称作儒字），应赋予
其在 ISO 10646 及统一码标准中有相同编码。此计划原本只包含中文、日文及韩文中所使用的
汉字，是以旧称中日韩统一表意文字（CJK）。后来，此计划加入了越文的喃字，所以合称中
日韩越统一表意文字（CJKV）。
107
108 CJK 統合漢字（シージェーケーとうごうかんじ、CJK Unified Ideographs）は、ISO/IEC 10646 (Uni-
versal Multiple-Octet Coded Character Set, 略称 UCS) および Unicode にて採用されてい
る符号化用漢字集合およびその符号表である。CJK 統合漢字の名称は、中国語 (Chinese)、
日本語 (Japanese)、韓国語 (Korean)で使われている漢字をひとまとめにしたことからき
ている。CJK 統合漢字の初版である Unified Repertoire and Ordering (URO) 第二版は
1992 年に制定されたが、1994 年にベトナムで使われていた漢字も含めることにしたため、
CJKV と呼ばれる事もある。CJKV は、中国語・日本語・韓国語・ベトナム語（英語：Chi-
nese-Japanese-Korean-Vietnamese）の略。特に、その 4 言語で共通して使われる、または
使われていた文字体系である漢字（チュノムを含む）のこと。ソフトウェアの国際化、中
でも文字コードに関する分野で用いられる。
109
110 \CJKspace
111 한중일월 통합 한자 (또는 한중일 통합 한자)는 유니코드에 담겨 있는 한자들의 집합으
로, 한국, 중국, 일본에서 쓰이는 한자를 묶은 것이기 때문에 머리 글자를 따서 한중일
(CJK) 통합 한자라고 불렀는데, 최근에는 베트남에서 쓰이는 한자도 추가되었기에 한중일
월 (CJKV) 통합 한자로 부르게 되었다.
112
113 처음에 유니코드에는 65,536(216)자만 들어갈 수 있었기 때문에, 가장 많은 문자
가 배당되는 한자를 위해서 한국, 중국, 일본에서 사용하는 한자 중에 모양이 유사하
며 그 뜻이 같은 글자를 같은 코드로 통합했다. 따라서 문자 코드만으로 그 한자가 사용
되는 언어를 알아 낼 수 없는데, 다만 중국의 간체자나 번체자, 일본의 구자체나 신자
체 등 분명하게 모양이 다른 글자는 별도의 부호를 할당하고 있다. 이런 문자 할당 정
책에 반발하여 TRON 과 같은 인코딩이 만들어지기도 했으나, 실제로 통합된 한자의 차이
가 별로 크지 않기 때문에 문제가 되지 않는다는 의견도 있다.
114 \CJKnospace}
115 \showtext
116
117 \bigskip
118 \xeCJKCancelSubCJKBlock{Kana, Hangul}
119 \showtext
120
121 \bigskip
```

```

122 \xeCJKRestoreSubCJKBlock{Hangul}
123 \showtext
124
125 \begin{table}[ht]
126 \caption{生僻字测试}
127 \medskip\centering
128 \begin{tabular}{*4{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 & \\
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 𠂻 & 20014 & 𠂼 & 20015 & 𠂽 & 20016 & 𠂾 & 20017 & \\
143 𠂿 & 20018 & 𠃀 & 20019 & 𠃁 & 2001A & 𠃂 & 2001B & \\
144 𠃃 & 2001C & 𠃄 & 2001D & 𠃅 & 2001E & 𠃆 & 2001F & \\
145 \end{tabular}
146 \end{table}
147 \end{document}
148 </ex-block>

```