

xeCJK 宏包

ctex.org*

2012/05/19 v3.0.5

Contents

1	简介	2
2	基本用法	2
3	用户手册	3
3.1	宏包选项	3
3.2	字体设置与选择	5
3.2.1	X _Y T _E X 的字体名查找	8
3.3	CJK 分区字体设置	9
3.4	设置 CJK 字符范围	10
3.5	设置标点宽度和间距	10
4	已知问题和兼容性	11
5	xeCJK 代码实现	12
5.1	字符类别设定	13
5.2	字符输出规则	19
5.3	段末孤字处理	27
5.4	增加 CJK 子分区	30
5.5	标点处理	32
5.6	后备字体	40
5.7	CJK 字体族声明方式	41
5.8	字体切换	50
5.9	处理等宽字体和抄录环境	55
5.10	xeCJK 其它选项	57
5.11	xeCJK 初始化设置	58
5.12	兼容性修补	59
6	例子	62
6.1	example-autofake.tex	62
6.2	example-fallback.tex	63
6.3	example-SubCJKBlock.tex	64

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

1 简介

xeCJK 是一个 Xe_{La}TeX 宏包,用于排版中日韩(CJK)文字。主要功能:

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

xeCJK 使用了 Xe_{La}TeX 的一些最新特性,需要 Xe_{La}TeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 的依赖 L^AT_EX 3 项目的 expl3 宏包,并依赖 fontspec¹ 宏包来调用系统字体。个别兼容性功能需要 CJK 宏包的支持。

xeCJK 的原始作者是孙文昌,2009 年 5 月起宏包被收入 ctex-kit²项目进行维护,目前主要维护者是刘海洋³ 和李清⁴。

2 基本用法

与其他 L^AT_EX 宏包一样,引入 xecjk 宏包只要在导言区使用

```
\usepackage{xeCJK}
```

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

可以在各种文档类中使用 xecjk 宏包,最简单的示例是:

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

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

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

xeCJK 只提供了字体和标点控制等基本 CJK 语言支持。对于中文文档,可以使用更为高层的 ctex 宏包或文档类,它将自动调用 xecjk 并设置好中文字体,同时提供了进一步的本地化支持。详细内容参看 ctex 宏包⁵的说明。

xeCJK 提供了大量选项,可以在宏包调用时作为宏包选项或用 \xeCJKsetup 命令进行设置,详见 3.1 节。除了 \setCJKmainfont 命令,xecjk 还提供了许多其他命令设置和选择中文字体,详见 3.2 节。其他更详细的功能也都将在下面详细说明。

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

²<http://code.google.com/p/ctex-kit/>

³leoliu.pku@gmail.com

⁴sobenlee@gmail.com

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

3 用户手册

3.1 宏包选项

`xeCJK` 以 $\langle key \rangle = \langle var \rangle$ 的形式提供宏包选项, 你可以在调用宏包的时候直接设置这些选项, 也可以在调用宏包之后使用 `\xeCJKsetup` 来设置这些选项。

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

其中 $\langle key1 \rangle, \langle key2 \rangle$ 是设置选项, 而 $\langle val1 \rangle, \langle val2 \rangle$ 则是对应选项的设置内容。多个选项可以在一个语句中完成设置。例如

```
\usepackage[PunctStyle=kaiming]{xeCJK}
```

等价于

```
\usepackage{xeCJK}
.....
\xeCJKsetup{PunctStyle=kaiming}
```

有些选项或命令后面带有 \star 号, 这表示这个选项或命令只能在导言区中使用, 而 \star 号则表示这个选项或命令只能在导言区使用, 并且只影响随后定义的 `CJK` 字体。其余不带特殊标记的选项或命令, 如果没有特别说明, 可以在导言区或正文中使用。

下列选项使用加粗的形式来表示 `xeCJK` 的默认设置。

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

打开/关闭对中文的特殊处理。

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

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

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

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

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

缺省状态下, `xeCJK` 会忽略 `CJK` 文字之间的空格, 使用这一选项来保留它们之间的空格。

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

是否支持在数学环境中直接输入 `CJK` 字符。使用这个选项后, 可以直接在数学环境中输出 `CJK` 字符。

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

设置 `CJK` 文字之间插入的 `glue`, 上边是 `xeCJK` 的默认值。一般来说, 除非有特殊需要(例如, 改变文字间距等), 否则不需要设置这个选项, 使用默认值即可。如果要设置这个选项, 为了行末的对齐, 设置的 `glue` 最好有一定的弹性。

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

KaiMingPunct	★	KaiMingPunct = { . 。 ? ! }
KaiMingPunct+	★	设置开明(kaiming)标点处理格式时的句末点号,KaiMingPunct 后带的 + 与 - 分别表示从已有的开明句末点号中增加或减少标点。
KaiMingPunct-	★	

LongPunct	★	LongPunct = { — …… — }
LongPunct+	★	设置长标点,例如破折号“——”与省略号“……”,允许在长标点前后断行,但是禁止在它们之间断行。
LongPunct-	★	

MiddlePunct	★	MiddlePunct = { • • • }
MiddlePunct+	★	设置居中显示的标点,例如间隔号“•”。对于在 CJK 文字之间的居中标点,xeCJK 会根据不同的标点处理格式,调整居中标点与前后文字之间的空白,保证其确实居中。对于行末出现的居中标点,允许在其后面断行,但禁止在它前面断行。
MiddlePunct-	★	

PunctWidth	★	PunctWidth = { <dimen> }
缺省状态下,xeCJK 会根据所选择的标点处理格式自动计算标点所占的宽度,如果对缺省设置不满意,可以通过这一选项来改变它。为了使得标点所占的宽度能够适应字体大小的变化,这里设置的 dimen 的单位最好用 em 等相对距离单位,而不建议使用诸如 pt 之类的绝对距离单位。这里的设置对除了 plain 以外的所有标点处理格式都有效。同时,这里的设置对所有的 CJK 标点都生效,如果只要设置部分标点,请使用 3.5 节的 \xeCJKsetwidth。		

AllowBreakBetweenPuncts		AllowBreakBetweenPuncts = { true false }
缺省状态下,xeCJK 禁止在相邻 CJK 右标点和 CJK 左标点之间换行,可以使用这一选项改变这一设置。		

3.2 字体设置与选择

\setCJKmainfont	★	\setCJKmainfont [] { }
设置正文罗马族的 CJK 字体,影响 \rmfamily 和 \textrm 的字体。后面两个参数继承自 fontspec 宏包, 表示字体属性选项, 是字体名。字体名可以是字体族名,也可以是字体的文件名,查找字体名见 3.2.1 节;可用的字体属性选项参见 fontspec 宏包的文档。需要说明的是 xeCJK 修改了 AutoFakeBold 和 AutoFakeSlant 选项,以便配合全局伪粗体和伪斜体的设定。		

AutoFakeBold		AutoFakeBold = { true false 数字 }
AutoFakeSlant		AutoFakeSlant = { true false 数字 }
局部设置当前字体族的伪粗和伪斜属性。如果没有在局部给出这些选项,将使用全局设定。		

\setCJKsansfont	★	\setCJKsansfont [] { }
设置正文无衬线族的 CJK 字体,影响 \sffamily 和 \textsf 的字体。		

\setCJKmonofont	★	\setCJKmonofont [] { }
设置正文等宽族的 CJK 字体,影响 \ttfamily 和 \texttt 的字体。为了有利于等宽字体的代码对齐等情形,xeCJK 在 里增加了 Mono 这个选项。		

<hr/> Mono <hr/>	<p><code>Mono = <Scale> Exspace</code></p> <p>有效值仅为 <code>Scale</code> 或 <code>Exspace</code>。分别表示使用放缩的方式或改变间距的方式,让 CJK 字符的宽度等于西文字体中两个空格的宽度。</p>
<hr/> <code>\CJKfixedspacing</code> <hr/> <hr/> <code>\CJKflexiblespacing</code> <hr/>	<p><code>\CJKfixedspacing</code></p> <p><code>\CJKflexiblespacing</code></p> <p><code>\CJKfixedspacing</code> 命令设置固定字距和不压缩标点格式,符合上面 <code>Mono=Exspace</code> 选项的意义。<code>\CJKflexiblespacing</code> 命令恢复可伸缩的字距和标点压缩。抄录 (verbatim) 环境将自动使用 <code>\CJKfixedspacing</code> 的效果。</p>
<hr/> <code>\setCJKmonoscale</code> <hr/> <hr/> <code>\setCJKmonoexspace</code> <hr/>	<p><code>\setCJKmonoscale</code></p> <p><code>\setCJKmonoexspace</code></p> <p><code>\setCJKmonoscale</code> 和 <code>\setCJKmonoexspace</code> 命令分别计算并设置当前字号下西文等宽字体与 CJK 字体的放缩比例和字距,以符合上面所说的 <code>Mono=Scale</code> 与 <code>Mono=Exspace</code> 选项的意义。在字号改变时,应该使用两个命令之一,重新计算相关参数;但注意两个命令不能同时使用。</p>
<hr/> <code>\setCJKfamilyfont</code> ★ <hr/>	<p><code>\setCJKfamilyfont <{family}> [<{font features}>] <{font name}></code></p> <p>声明新的 CJK 字体族 <i><family></i> 并指定字体。</p>
<hr/> <code>\CJKfamily</code> <hr/>	<p><code>\CJKfamily <{family}></code></p> <p>用于在文档中切换 CJK 字体族, <i><family></i> 必须预先声明。</p>
<hr/> <code>\newCJKfontfamily</code> ★ <hr/>	<p><code>\newCJKfontfamily [<{family}>] \<{font-switch}> [<{font features}>] <{font name}></code></p> <p>声明新的 CJK 字体族 <i><family></i> 并指定字体,并定义 <code>\<{font-switch}></code>,在文档中可以使用它来切换 CJK 字体族。可以不必指定 <i><family></i>,这时候 <i><family></i> 将等于 <i><font-switch></i>。事实上, <code>\newCJKfontfamily</code> 是 <code>\setCJKfamilyfont</code> 和 <code>\CJKfamily</code> 的合并。例如</p> <p style="padding-left: 40px;"><code>\newCJKfontfamily[song]\songti{SimSun}</code></p> <p>等价于</p> <p style="padding-left: 40px;"> <code>\setCJKfamilyfont{song}{SimSun}</code> <code>\newcommand\songti{\CJKfamily{song}}</code> </p>
<hr/> <code>\CJKfontspec</code> <hr/>	<p><code>\CJKfontspec [<{font features}>] <{font name}></code></p> <p>在文档中随机定义新的 CJK 字体族,并马上使用它。</p>
<hr/> <code>\defaultCJKfontfeatures</code> ☆ <hr/>	<p><code>\defaultCJKfontfeatures <{font features}></code></p> <p>全局设置 CJK 字体族的默认选项。例如,使用</p> <p style="padding-left: 40px;"><code>\defaultCJKfontfeatures{Scale=0.962216}</code></p> <p>可以将全部 CJK 字体缩小为 0.962216。</p>

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

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

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

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

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

3.2.1 X_YTeX 的字体名查找

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

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

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

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

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

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

在 fontspec 或 xeCJK 中使用的字体族名是上面列表中冒号前的部分。例如可以使用

```
\setmainfont{Times New Roman}
\setCJKmainfont{SimSun} % 或者 \setCJKmainfont{宋体}
```

来设置字体。

为了方便起见, `fc-list` 命令也可以加上各种选项控制输出格式, 例如如果只要列出所有的中文字体的字体族名, 可以用命令:

```
fc-list -f "%{family}\n" :lang=zh > zhfont.txt
```

这样就把字体列表保存在文件 `zhfont.txt` 中⁶。这样列出的字体列表就比较简明易用, 如 Windows 下预装的中文字体:

```
Arial Unicode MS
FangSong, 仿宋
KaiTi, 楷体
Microsoft YaHei, 微软雅黑
MingLiU, 細明體
NSimSun, 新宋体
PMingLiU, 新細明體
SimHei, 黑体
SimSun, 宋体
```

要列出日文和韩文的字体, 可以把 `:lang=zh` 选项中的 `zh` 改成 `ja` 或 `ko`。

`fontspec` 和 `xeCJK` 也可以使用字体的文件名访问字体。例如 Windows 下的宋体也可以使用命令:

```
\setCJKmainfont{simsun.ttc}
```

来设置。设置字体文件名的相关选项和语法在 `fontspec` 宏包手册中叙述甚详, 这里不再赘述。有个别字体名不规范的中文字体, `xeCJK` 宏包可能无法正确地通过字体名访问, 那么也可以使用这种方式设置。

3.3 CJK 分区字体设置

众所周知, CJK 文字数量极其庞大, 单一的字体不可能涵盖所有的 CJK 文字。 `xeCJK` 可以在同一 CJK 字体族下, 自动使用不同的字体输出 CJK 字符范围内不同区块里的文字。首先要声明 CJK 子分区。

⁶由于汉字编码原因, Windows 下总需要把字体列表输出的文件中防止乱码。

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

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

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

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

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

就声明了 SPUA 和 Ext-B 这两个个子分区。同时在 3.2 节介绍的 CJK 字体设置命令的 ** 里新建了 SPUA 和 Ext-B 这两个选项。新建的这两个选项的使用方法跟 3.2 介绍的 FallBack 类似。可以通过它们来设置字体。例如, 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。

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

在文档中取消对 CJK 分区的声明。

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

在文档中恢复对 CJK 分区的声明。

3.4 设置 CJK 字符范围

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

<class range> 的格式和 3.3 节的 *<block range>* 相同。*<class>* 的有效值见源代码(第 5.1 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。

`\xeCJKResetPunctClass` ★ `\xeCJKResetPunctClass`

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`。

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

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

3.5 设置标点宽度和间距

这里的设置对除了 plain 以外的所有标点处理格式都有效。

`\xeCJKsetwidth` ★ `\xeCJKsetwidth {⟨标点列表⟩} {⟨dimen⟩}`

⟨标点列表⟩ 可以是单个标点,也可以是多个标点。例如,

`\xeCJKsetwidth{。}{0.7em}`

将设置句号和问号所占的宽度为 `0.7em`。

`\xeCJKsetkern` ★ `\xeCJKsetkern {⟨标点 1⟩} {⟨标点 2⟩} {⟨dimen⟩}`

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

`\xeCJKsetkern{:}{“}{0.3em}`

4 已知问题和兼容性

新版本的 `xeCJK` 完全使用 `LATEX3` 的语法来编写,受限于目前 `expl3` 宏包的相关工具的实现方式,在 CJK 文字后面使用 `\outer` 宏时会出现类似

`! Forbidden control sequence found while scanning use of \token_if_eq_meaning:NNTF`

的错误,在 `\outer` 宏前面加上 `\relax` 可以暂时避免这个错误。目前已知的有 `cprotect` 宏包提供的 `\cprotect`,这时你也可以暂时用 `\icprotect` 代替 `\cprotect`。⁷

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。`xeCJK` 虽然为此作了一些处理,但与它们共同使用时应该小心。

⁷<http://code.google.com/p/ctex-kit/issues/detail?id=87>

5 xeCJK 代码实现

```

1 <*package>
2 \RequirePackage{xparse}
3 \RequirePackage{l3keys2e}
4 \ProvidesExplPackage {xeCJK} {2012/05/19} {3.0.5}
5 {package for typesetting CJK scripts with XeLaTeX}

\XeTeXmsgnew:nn
\XeTeXerror:n
\XeTeXerror:nx
\XeTeXwarning:nx
\XeTeXinfo:nx
\XeTeXinfo:nxx
6 \cs_new_nopar:Npn \XeTeXmsg_new:nn { \msg_new:nnn { xeCJK } }
7 \cs_new_nopar:Npn \XeTeXerror:n { \msg_error:nn { xeCJK } }
8 \cs_new_nopar:Npn \XeTeXerror:nx { \msg_error:nnx { xeCJK } }
9 \cs_new_nopar:Npn \XeTeXwarning:n { \msg_warning:nn { xeCJK } }
10 \cs_new_nopar:Npn \XeTeXwarning:nx { \msg_warning:nnx { xeCJK } }
11 \cs_new_nopar:Npn \XeTeXwarning:nxx { \msg_warning:nnxx { xeCJK } }
12 \cs_new_nopar:Npn \XeTeXinfo:nx { \msg_info:nnx { xeCJK } }
13 \cs_new_nopar:Npn \XeTeXinfo:nxx { \msg_info:nnxx { xeCJK } }

(End definition for \XeTeXmsg_new:nn and others.)
XeCJK 必须使用 XeLaTeX 引擎的支持。
14 \msg_new:nnn {xeCJK} { Require-XeTeX }
15 {
16   The~xeCJK~package~requires~XeTeX~to~function.\\\
17   You~must~change~your~typesetting~engine~to~"xelatex" \\\
18   instead~of~plain~"latex"~or~"pdflatex"~or~"lualatex".
19 }
20 \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。
21 \XeTeXmsg_new:nn { XeTeX-too-old }
22 {
23   \token_to_str:N \XeTeXglyphbouds\c_space_tl is~not~defined.\\\
24   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
25 }
26 \cs_if_free:NT \XeTeXglyphbouds
27 {
28   \XeTeXerror:n { XeTeX-too-old }
29   \AtBeginDocument { \punctstyle { plain } }
30 }
抑制 BOM。
31 \char_set_catcode_ignore:n { "FEFF }

xeCJKactive xeCJK 功能开关。
32 \keys_define:nn { xeCJK / options }
33 {
34   xeCJKactive .choice:,
35   xeCJKactive / true .code:n = { \makexeCJKactive } ,
36   xeCJKactive / false .code:n = { \makexeCJKinactive } ,
37   xeCJKactive .default:n = { true },
38 }
(End definition for xeCJKactive. This function is documented on page 3.)

```

```

\makeXeCJKActive
\makeXeCJKInactive 39 \NewDocumentCommand \makeXeCJKActive { } { \XeTeXinterchartokenstate = \c_one }
40 \NewDocumentCommand \makeXeCJKInactive { } { \XeTeXinterchartokenstate = \c_zero }
(End definition for \makeXeCJKActive and \makeXeCJKInactive.)

```

5.1 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 中预定义类别,xeCJK 新增加了 HalfLeft、HalfRight、NormalSpace。

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

```

\XeCJK_Default_class 这五类是 XeLaTeX 预定义类别。
\XeCJK_CJK_class 41 \int_const:Nn \XeCJK_Default_class { 0 }
\XeCJK_FullLeft_class 42 \int_const:Nn \XeCJK_CJK_class { 1 }
\XeCJK_FullRight_class 43 \int_const:Nn \XeCJK_FullLeft_class { 2 }
\XeCJK_Boundary_class 44 \int_const:Nn \XeCJK_FullRight_class { 3 }
45 \int_const:Nn \XeCJK_Boundary_class { 255 }
(End definition for \XeCJK_Default_class and others.)

```

```

\XeCJK_new_class:n 新建一个字符类别。
\g_xeCJK_class_seq 46 \cs_new_nopar:Nn \XeCJK_new_class:n
47 {
48   \exp_args:Nc \newXeTeXintercharclass { xeCJK_#1_class }
49   \clist_gclear_new:c { g_xeCJK_#1_range_clist }
50   \seq_gput_right:Nv \g_xeCJK_class_seq { xeCJK_#1_class }
51 }
52 \seq_new:N \g_xeCJK_class_seq
(End definition for \XeCJK_new_class:n and \g_xeCJK_class_seq.)

```

```

\XeCJK_HalfLeft_class 新增西文半角左/右标点和前后原始间距的符号类。
\XeCJK_HalfRight_class 53 \XeCJK_new_class:n { HalfLeft }
\XeCJK_NormalSpace_class 54 \XeCJK_new_class:n { HalfRight }
55 \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 56 \clist_set:Nn \g_xeCJK_base_class_clist
\g_xeCJK_CJK_class_prop 57 {
58   Default, CJK, FullLeft, FullRight, Boundary,
59   HalfLeft, HalfRight, NormalSpace
60 }

```

```

61 \clist_map_inline:Nn \g_xeCJK_base_class_clist
62 { \clist_gclear_new:c { g_xeCJK_#1_range_clist } }
63 \clist_new:N \g_xeCJK_CJK_class_clist
64 \prop_new:N \g_xeCJK_CJK_class_prop
65 \cs_new_nopar:Nn \xeCJK_save_CJK_class:n
66 {
67   \clist_gput_right:Nx \g_xeCJK_CJK_class_clist {#1}
68   \prop_gput:Nvx \g_xeCJK_CJK_class_prop { xeCJK_#1_class } {#1}
69 }
70 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
71 \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。

```

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

```

`\tl_if_blank_p:x`

`\tl_if_blank:xTF`

```

73 \prg_set_conditional:Nnn \tl_if_blank:x { p , T , F , TF }
74 {
75   \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
76   \prg_return_true:
77   \else:
78     \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
79     \prg_return_true: \else: \prg_return_false: \fi:
80   \fi:
81 }
(End definition for \tl_if_blank:x.)

```

`\xeCJKDeclareCharClass`

`\xeCJK_setcharclass_aux:Nn`

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。

```

82 \NewDocumentCommand \xeCJKDeclareCharClass { > { \TrimSpaces } m m }
83 {
84   \clist_gput_right:cx { g_xeCJK_#1_range_clist } {#2}
85   \clist_map_inline:xn {#2}
86   {
87     \str_if_eq:xxF {##1} { -> }
88     {
89       \xeCJK_setcharclass_aux:Nn \xeCJKsetcharclass {##1}
90       { \xeCJK_class_num:n {#1} }
91     }
92   }
93 }
94 \NewDocumentCommand \xeCJK_setcharclass_aux:Nn
95 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
96 \cs_generate_variant:Nn \clist_map_inline:nn { x }
(End definition for \xeCJKDeclareCharClass. This function is documented on page 10.)

```

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

```

97 \NewDocumentCommand \xeCJKsetcharclass { m m m }
98 {
99   \xeCJK_check_num_range:nnNN {#1} {#2} \l_tmpa_int \l_tmpb_int

```

```

100 \int_set:Nn \l_tmpc_int {#3}
101 \prop_if_in:NVTF \g_xeCJK_CJK_class_prop \l_tmpc_int
102 { \cs_set_eq:NN \xeCJK_set_catcode:n \char_set_catcode_other:n }
103 { \cs_set_eq:NN \xeCJK_set_catcode:n \use_none:n }
104 \loop
105 \xeCJK_set_catcode:n \l_tmpa_int
106 \XeTeXcharclass \l_tmpa_int = \l_tmpc_int
107 \if_int_compare:w \l_tmpa_int < \l_tmpb_int
108 \int_incr:N \l_tmpa_int
109 \repeat
110 }
(End definition for \xeCJKsetcharclass.)

```

\xeCJK_check_num_range:nnNN

```

111 \cs_new_nopar:Nn \xeCJK_check_num_range:nnNN
112 {
113   \bool_if:nTF { \tl_if_blank_p:x {#1} || \tl_if_blank_p:x {#2} }
114   {
115     \int_set:Nn #3 { \tl_if_blank:xTF {#1} {#2} {#1} }
116     \int_set_eq:NN #3 #4
117   }
118   {
119     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
120     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
121   }
122 }
(End definition for \xeCJK_check_num_range:nnNN.)

```

\xeCJKResetPunctClass 用于重置标点符号所属的字符类。

```

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

```

```

146         "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 , "FF69 ,
147         "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E , "FF9F ,
148     }
149 }

```

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

设置 CJK 文字的范围。

```

150 \xeCJKDeclareCharClass { CJK }
151 {

```

- Hangul Jamo (谚文字母)

```

152         "1100 -> "11FF ,

```

- CJK Radicals Supplement (中日韩部首补充)

```

153         "2E80 -> "2EFF ,

```

- Kangxi Radicals (康熙字典部首)

```

154         "2F00 -> "2FDF ,

```

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

```

155         "2FF0 -> "2FFF ,

```

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

```

156         "3000 -> "303F ,

```

- Hiragana (日文平假名)

```

157         "3040 -> "309F ,

```

- Katakana (日文片假名)

```

158         "30A0 -> "30FF ,

```

- Bopomofo (注音字母)

```

159         "3100 -> "312F ,

```

- Hangul Compatibility Jamo (谚文兼容字母)

```

160         "3130 -> "318F ,

```

- Kanbun (象形字注释标志)

```

161         "3190 -> "319F ,

```

- Bopomofo Extended (注音字母扩展)

```

162         "31A0 -> "31BF ,

```

- CJK Strokes (中日韩笔画)

```

163         "31C0 -> "31EF ,

```

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

```

164         "31F0 -> "31FF ,

```

- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
165 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
166 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
167 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
168 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
169 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
170 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
171 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
172 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
173 "AC00 -> "D7AF ,
- Kana Supplement (日文假名增补)
174 "B000 -> "B0FF ,
- Hangul Jamo Extended-B (谚文扩展 B)
175 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
176 "F900 -> "FAFF ,
- CJK Compatibility Forms (中日韩兼容形式)
177 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)
178 "FF00 -> "FFEF ,
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)
179 "20000 -> "2A6DF ,
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)
180 "2A700 -> "2B73F ,

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

```
181 "2B740 -> "2B81F ,
```

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

```
182 "2F800 -> "2FA1F ,
```

```
183 }
```

重置标点符号的字符类。

```
184 \xeCJKResetPunctClass
```

`\normalspacedchars` 声明前后不加间距的字符。

```
185 \NewDocumentCommand \normalspacedchars { m }
186 {
187   \tl_map_inline:nn {#1}
188   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
189 }
190 \normalspacedchars{/}
(End definition for \normalspacedchars. This function is documented on page 10.)
```

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

```
\xeCJK_inter_class_toks:nnc
191 \cs_new_nopar:Nn \xeCJK_inter_class_toks:nnn
192 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = {#3} }
193 \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。

```
194 \cs_new_nopar:Nn \xeCJK_clear_inter_class_toks:nn
195 {
196   \XeTeXinterchartoks
197   \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = { \c_empty_tl }
198 }
(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。

```
199 \cs_new_nopar:Nn \xeCJK_pre_inter_class_toks:nnn
200 {
201   \use:x
202   {
203     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
204     {
205       \exp_not:n {#3} \tex_the:D \XeTeXinterchartoks
206       \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2}
207     }
208   }
209 }
(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
210 \cs_new_nopar:Nn \xeCJK_app_inter_class_toks:nnn
211 {
212   \use:x
213   {
214     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
215     {
216       \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1}
217       \xeCJK_class_num:n {#2} \exp_not:n {#3}
218     }
219   }
220 }
221 \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。
222 \cs_new_nopar:Nn \xeCJK_get_inter_class_toks:nn
223 { \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} }
(End definition for \xeCJK_get_inter_class_toks:nn.)

\xeCJK_copy_inter_class_toks:nnnn 将 #3 和 #4 之间的内容复制到 #1 和 #2 之间。
224 \cs_new_nopar:Nn \xeCJK_copy_inter_class_toks:nnnn
225 {
226   \use:x
227   {
228     \xeCJK_inter_class_toks:nnn {#1} {#2}
229     { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
230   }
231 }
(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。
232 \cs_new_nopar:Nn \xeCJK_clear_CJK_toks:
233 {
234   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
235   { \xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
236 }
(End definition for \xeCJK_clear_CJK_toks:.)

```

5.2 字符输出规则

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

```

237 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
238 {
239   \clist_map_inline:nn { CJK, FullLeft, FullRight }
240   {
241     \xeCJK_inter_class_toks:nnn {#1} {##1}
242     {
243       \c_group_begin_token
244       \xeCJK_select_font:
245       \xeCJK_clear_inter_class_toks:nn {#1} {##1}
246       \xeCJK_clear_CJK_toks:
247     }
248     \xeCJK_inter_class_toks:nnn {##1} {#1} { \c_group_end_token }
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 }
254 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
255 {
256   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
257   {
258     \int_compare:nNnTF \tex_lastkern:D = \c_one { \CJKecglue }
259     { \int_compare:nNnT \tex_lastkern:D = \c_two \xeCJK_space_or_xecglue: }
260   }
261   \str_if_eq:xxF {#1} { NormalSpace }
262   { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
263 }
264 \clist_map_inline:nn { Default, HalfRight, NormalSpace }
265 {
266   \xeCJK_inter_class_toks:nnn {#1} { Boundary }
267   {
268     \peek_catcode:NTF \c_space_token
269     { { \xeCJK_default_space_kern: } } { { \xeCJK_default_kern: } }
270   }
271   \str_if_eq:xxF {#1} { NormalSpace }
272   { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
273 }
274 \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }

```

\xeCJK_Boundary_and_CJK:

```

275 \cs_new_nopar:Nn \xeCJK_Boundary_and_CJK:
276 {
277   {
278     \bool_if:nTF
279     {
280       \int_compare_p:nNn \tex_lastkern:D = \c_one ||
281       \int_compare_p:nNn \tex_lastkern:D = \c_two
282     }
283     { \CJKglue }
284     {
285       \bool_if:nTF

```

```

286     {
287         \int_compare_p:nNn \tex_lastkern:D = \c_three      ||
288         \int_compare_p:nNn \etex_lastnodetype:D = \c_ten   ||
289         ( \int_compare_p:nNn \etex_lastnodetype:D = \c_twelve &&
290           \g_xeCJK_itscorr_bool )
291     }
292     { \CJKecglue }
293     {
294         \bool_if:nT
295         {
296             \l_xeCJK_xecglue_bool &&
297             \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
298             ( \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl ||
299               \int_compare_p:nNn \tex_lastskip:D = \l_xeCJK_ecglue_dim )
300         }
301         {
302             \skip_set_eq:NN \l_tmpa_skip \tex_lastskip:D
303             \tex_unskip:D
304             \bool_if:nTF
305             {
306                 \int_compare_p:nNn \tex_lastkern:D = \c_four      ||
307                 \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
308                 \int_compare_p:nNn \tex_lastkern:D = \c_three     ||
309                 \g_xeCJK_itscorr_bool
310             }
311             { \CJKecglue }
312             {
313                 \bool_if:nT
314                 {
315                     ! ( \int_compare_p:nNn \tex_lastkern:D = \c_one ) ||
316                     ! ( \int_compare_p:nNn \tex_lastkern:D = \c_two ) ||
317                     \l_xeCJK_space_bool
318                 }
319                 { \skip_horizontal:N \l_tmpa_skip }
320             }
321         }
322     }
323 }
324 \bool_gset_false:N \g_xeCJK_itscorr_bool
325 }
326 \c_group_begin_token
327 \xeCJK_clear_CJK_toks:
328 \xeCJK_select_font:
329 \CJKsymbol
330 }
331 \tl_set:Nn \l_xeCJK_space_glue_tl
332 {
333     \tex_fontdimen:D \c_two \tex_font:D
334     \@plus \tex_fontdimen:D \c_three \tex_font:D
335     \@minus \tex_fontdimen:D \c_four \tex_font:D
336 }
(End definition for \xeCJK_Boundary_and_CJK:.)

```

```

337 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
338 {
339   \c_group_end_token
340   { \xeCJK_CJK_kern: }
341   \xeCJK_ignorespaces:
342 }
343 \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }

```

`\xeCJK_CJK_and_CJK:N`

```

344 \cs_new_nopar:Nn \xeCJK_CJK_and_CJK:N { \CJKglue \CJKsymbol {#1} }
(End definition for \xeCJK_CJK_and_CJK:N)

345 \clist_map_inline:nn { FullLeft, FullRight }
346 {
347   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
348   {
349     \c_group_begin_token
350     \xeCJK_clear_CJK_toks:
351     \xeCJK_select_font:
352     \use:c { xeCJK_CJK_and_#1:N }
353   }
354   \xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
355   \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \c_group_end_token \tex_ignorespaces:D }
356 }

357 \clist_map_inline:Nn \g_xeCJK_base_class_clist
358 {
359   \bool_if:nF
360     { \str_if_eq_p:xx {#1} { FullLeft } || \str_if_eq_p:xx {#1} { FullRight } }
361     {
362       \xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \nobreak }
363       \xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \xeCJK_after_FullRight: }
364     }
365 }

366 \clist_map_inline:nn { CJK, FullLeft, FullRight }
367 {
368   \clist_map_inline:nn { FullLeft, FullRight }
369     { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
370 }

```

`\xeCJK_punct_rule:n` 用于抹去标点符号的左/右空白。

```

371 \cs_new_nopar:Nn \xeCJK_punct_rule:n
372 {
373   \tex_vrule:D
374   \@width \dim_eval:c { \l_xeCJK_punct_coors_tl/rule/#1 }
375   \@depth \c_zero_dim
376   \@height \c_zero_dim \scan_stop:
377 }
(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。

```

378 \cs_new_nopar:Nn \xeCJK_punct_glue:n
379 {

```

```

380 \skip_horizontal:n
381 {
382   \dim_eval:c { \l_xeCJK_punct_coortl/glue/#1 }
383   \@plus \dim_eval:n { \tl_use:c { \l_xeCJK_punct_coortl/glue/#1 } /4 }
384   \@minus \dim_eval:n { \tl_use:c { \l_xeCJK_punct_coortl/glue/#1 } /2 }
385 }
386 }

```

(End definition for \xeCJK_punct_glue:n.)

\xeCJK_punct_kern:n 相邻两个标点之间的间距。

```

387 \cs_new_nopar:Nn \xeCJK_punct_kern:n
388 { \tex_kern:D \dim_eval:c { \l_xeCJK_punct_coortl/kern/#1 } }
(End definition for \xeCJK_punct_kern:n.)

```

\dim_eval:c

```

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

```

\xeCJK_before_FullLeft:N

```

390 \cs_new_nopar:Nn \xeCJK_before_FullLeft:N
391 {
392   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
393   \xeCJK_punct_rule:n { 1/#1 }
394   \CJKpunctsymbol {#1}
395 }
(End definition for \xeCJK_before_FullLeft:N.)

```

\xeCJK_after_FullRight:

```

396 \cs_new_nopar:Nn \xeCJK_after_FullRight:
397 {
398   \xeCJK_get_punct_bounds:nV { r } \l_xeCJK_lastpunct_tl
399   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
400   \xeCJK_punct_glue:n { r/\l_xeCJK_lastpunct_tl }
401 }
(End definition for \xeCJK_after_FullRight:.)

```

\xeCJK_CJK_and_FullLeft:N

```

402 \cs_new_nopar:Nn \xeCJK_CJK_and_FullLeft:N
403 {
404   \xeCJK_get_punct_bounds:nN { 1 } {#1}
405   \int_compare:nNf \etex_lastnodetype:D = \c_one
406     { \xeCJK_punct_glue:n { 1/#1 } }
407   \xeCJK_before_FullLeft:N {#1}
408   \xeCJK_punct_if_middle:NT {#1}
409   {
410     \xeCJK_punct_rule:n { m/#1 }
411     \xeCJK_punct_glue:n { 1/#1 }
412   }
413 }
(End definition for \xeCJK_CJK_and_FullLeft:N.)

```

\xeCJK_CJK_and_FullRight:N

```
414 \cs_new_nopar:Nn \xeCJK_CJK_and_FullRight:N
415 {
416   \xeCJK_get_punct_bounds:nN { r } {#1}
417   \xeCJK_punct_if_long:NTF {#1} \CJKglue
418   {
419     \nobreak
420     \xeCJK_punct_if_middle:NT {#1}
421     {
422       \xeCJK_punct_glue:n { r/#1 }
423       \xeCJK_punct_rule:n { m/#1 }
424     }
425   }
426   \tl_gset:Nx \l_xeCJK_lastpunct_tl {#1}
427   \CJKpunctsymbol {#1}
428 }
```

(End definition for \xeCJK_CJK_and_FullRight:N.)

\xeCJK_FullLeft_and_FullLeft:N

```
429 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullLeft:N
430 {
431   \nobreak
432   \xeCJK_get_punct_bounds:nN { l } {#1}
433   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
434   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
435   \xeCJK_before_FullLeft:N {#1}
436 }
```

(End definition for \xeCJK_FullLeft_and_FullLeft:N.)

\xeCJK_FullLeft_and_FullRight:N

```
437 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullRight:N
438 {
439   \nobreak
440   \xeCJK_get_punct_bounds:nN { r } {#1}
441   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
442   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
443   \nobreak
444   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
445   \CJKpunctsymbol {#1}
446 }
```

(End definition for \xeCJK_FullLeft_and_FullRight:N.)

\xeCJK_FullRight_and_FullLeft:N

```
447 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullLeft:N
448 {
449   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
450   \xeCJK_get_punct_bounds:nN { l } {#1}
451   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
452   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
453   \xeCJK_punct_nobreak:
454   \xeCJK_before_FullLeft:N {#1}
455 }
```

(End definition for \xeCJK_FullRight_and_FullLeft:N.)

\xeCJK_FullRight_and_FullRight:N

```

456 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullRight:N
457 {
458   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
459   \xeCJK_get_punct_bounds:nN { r } {#1}
460   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
461   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
462   \nobreak
463   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
464   \CJKpunctsymbol {#1}
465 }

```

(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: 456 \cs_new_nopar:Nn \xeCJK_CJK_kern: { \tex_kern:D -1 sp \tex_kern:D 1 sp }
\xeCJK_default_kern: 457 \cs_new_nopar:Nn \xeCJK_CJK_space_kern: { \tex_kern:D -2 sp \tex_kern:D 2 sp }
\xeCJK_default_space_kern: 458 \cs_new_nopar:Nn \xeCJK_default_kern: { \tex_kern:D -3 sp \tex_kern:D 3 sp }
459 \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。

```

470 \keys_define:nn { xeCJK / options } { CJKglue .tl_set_x:N = \CJKglue }
(End definition for CJKglue. This function is documented on page 3.)

```

CJKecglue CJK 与西文和数学行内数学公式之间自动添加的空白。

```

xCJKecglue 471 \keys_define:nn { xeCJK / options }
472 {
473   CJKecglue .code:n =
474   {
475     \cs_set:Npx \CJKecglue {#1}
476     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
477   },
478   xCJKecglue .choice:,
479   xCJKecglue / true .code:n =
480   {
481     \bool_set_true:N \l_xeCJK_xecglue_bool
482     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
483   },
484   xCJKecglue / false .code:n =
485   {
486     \bool_set_false:N \l_xeCJK_xecglue_bool
487     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_token
488   },
489   xCJKecglue / unknown .code:n =
490   {
491     \bool_set_true:N \l_xeCJK_xecglue_bool
492     \cs_set:Npx \CJKecglue {#1}
493     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
494     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
495   },

```

```

496     xCJKecglue .default:n = { true } ,
497   }
498   \dim_new:N \l_xeCJK_ecglue_dim
499   \bool_new:N \l_xeCJK_xecglue_bool

```

(End definition for CJKecglue and xCJKecglue. These functions are documented on page 4.)

CJKspace 是否保留 CJK 文字间的空白,默认不保留。

```

500   \keys_define:nn { xeCJK / options }
501   {
502     CJKspace .choice:,
503     CJKspace / true .code:n = { \CJKspace },
504     CJKspace / false .code:n = { \CJKnospace },
505     CJKspace .default:n = { true },
506     space .meta:n = { CJKspace = true },
507     nospace .meta:n = { CJKspace = false },
508   }

```

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

\CJKspace 是否保留 CJK 文字间的空白,默认不保留。

```

\CJKnospace
509   \NewDocumentCommand \CJKspace { }
510   {
511     \bool_set_true:N \l_xeCJK_space_bool
512     \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_peek_math:
513   }
514   \NewDocumentCommand \CJKnospace { }
515   {
516     \bool_set_false:N \l_xeCJK_space_bool
517     \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_ignore_spaces:
518   }

```

(End definition for \CJKspace and \CJKnospace.)

\xeCJK_ignore_spaces:

```

\xeCJK_peek_math:
519   \cs_new_nopar:Nn \xeCJK_ignore_spaces:
520   {
521     \peek_catcode:NTF \c_space_token
522     {
523       \int_compare:nNt \tex_lastkern:D = \c_one
524       { \tex_unkern:D \tex_unkern:D { \xeCJK_CJK_space_kern: } }
525       \peek_catcode_ignore_spaces:NTF \c_math_toggle_token
526       { \xeCJK_space_or_xecglue: }
527       { \bool_if:nT { \token_if_cs_p:N \l_peek_token } \xeCJK_space_or_xecglue: }
528     }
529     {
530       \token_if_math_toggle:NTF \l_peek_token \CJKecglue
531       { \cs_if_exist:NF \l_peek_token \exp_not:N }
532     }
533   }
534   \cs_new_nopar:Nn \xeCJK_peek_math:
535   {
536     \peek_catcode:NTF \c_space_token
537     {
538       \int_compare:nNt \tex_lastkern:D = \c_one

```

```

539         { \tex_unkern:D \tex_unkern:D { \xeCJK_CJK_space_kern: } }
540     \peek_catcode_ignore_spaces:NF \c_space_token { \xeCJK_space_or_xecglue: }
541 }
542 {
543     \token_if_math_toggle:NTF \l_peek_token \CJKeckglue
544     { \cs_if_exist:NF \l_peek_token \exp_not:N }
545 }
546 }

```

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

5.3 段末孤字处理

CheckSingle 孤字处理功能选项。

```

CheckSingleglue 547 \keys_define:nn { xeCJK / options }
548 {
549     CheckSingle .choice:,
550     CheckSingle / true .code:n = { \xeCJKenablechecksingle },
551     CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
552     CheckSingle .default:n = { true },
553     CJKchecksingle .meta:n = { CheckSingle = true },
554     CheckSingleglue .tl_set_x:N = { \CJKchecksingleglue },
555 }

```

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

\xeCJKenablechecksingle 孤字处理功能开关。

```

\xeCJKdisablechecksingle 556 \bool_new:N \l_xeCJK_checksingle_bool
557 \NewDocumentCommand \xeCJKenablechecksingle { }
558 {
559     \bool_if:NF \l_xeCJK_checksingle_bool
560     {
561         \bool_set_true:N \l_xeCJK_checksingle_bool
562         \cs_set_eq:NN \xeCJK_checksingle_save_CJKsymbol:N \xeCJK_CJK_and_CJK:N
563         \cs_set_eq:NN \xeCJK_checksingle_save_CJKglue \CJKglue
564         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle:N
565         \cs_set_nopar:Npn \CJKglue { \CJKchecksingleglue }
566     }
567 }
568 \NewDocumentCommand \xeCJKdisablechecksingle { }
569 {
570     \bool_if:NT \l_xeCJK_checksingle_bool
571     {
572         \bool_set_false:N \l_xeCJK_checksingle_bool
573         \cs_set_eq:NN \CJKglue \xeCJK_checksingle_save_CJKglue
574         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle_save_CJKsymbol:N
575     }
576 }

```

(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle.)

```

\xeCJK_checksingle:N
\xeCJK_checksingle:NN 577 \cs_new_nopar:Nn \xeCJK_checksingle:N
\xeCJK_checksingle:NNN 578 {
579     \peek_catcode:NTF \c_catcode_other_token

```

```

580     { \xeCJK_checksingle:NN {#1} }
581     {
582         \bool_if:nTF
583         {
584             \tl_if_empty_p:x { \token_get_arg_spec:N \l_peek_token } &&
585             \exp_args:No \tl_if_single_token_p:n \l_peek_token      &&
586             \exp_after:wN \token_if_other_p:N \l_peek_token
587         }
588         { \xeCJK_checksingle:NN {#1} }
589         { \xeCJK_checksingle_save_CJKsymbol:N {#1} }
590     }
591 }
592 \cs_new_nopar:Nn \xeCJK_checksingle:NN
593 {
594     \peek_catcode:NTF \c_catcode_other_token
595     { \xeCJK_checksingle_save_CJKsymbol:N {#1} #2 }
596     {
597         \bool_if:nTF
598         {
599             \token_if_space_p:N      \l_peek_token ||
600             \token_if_math_toggle_p:N \l_peek_token ||
601             \token_if_cs_p:N         \l_peek_token
602         }
603         {
604             \bool_if:nTF { \token_if_space_p:N \l_peek_token }
605             {
606                 \peek_catcode_ignore_spaces:NTF \c_catcode_other_token
607                 { \xeCJK_checksingle_save_CJKsymbol:N {#1} #2 \c_space_token }
608                 {
609                     \bool_if:nTF
610                     {
611                         \token_if_math_toggle_p:N \l_peek_token ||
612                         \token_if_cs_p:N         \l_peek_token
613                     }
614                     { \xeCJK_checksingle:NNN {#1} {#2} \c_space_token }
615                     { \xeCJK_checksingle_save_CJKsymbol:N {#1} #2 \c_space_token }
616                 }
617             }
618             { \xeCJK_checksingle:NNN {#1} {#2} { } }
619         }
620         { \xeCJK_checksingle_save_CJKsymbol:N {#1} #2 }
621     }
622 }
623 \cs_new_nopar:Nn \xeCJK_checksingle:NNN
624 {
625     \bool_if:nTF
626     {
627         \token_if_eq_meaning_p:NN \l_peek_token \par ||
628         \token_if_eq_meaning_p:NN \l_peek_token \[ % \
629     }
630     { \CJKsymbol {#1} \nobreak #2 }
631     { \xeCJK_checksingle_env:NNN {#1} {#2} {#3} }

```

632 }
 (End definition for \xeCJK_checksinglingle:N, \xeCJK_checksinglingle:NN, and \xeCJK_checksinglingle:NNN.)

\xeCJK_checksinglingle_env:NNN
 \xeCJK_checksinglingle_env:NNNNn 633 \cs_new_nopar:Nn \xeCJK_checksinglingle_env:NNN
 634 {
 635 \peek_catcode_remove:NTF \c_math_toggle_token
 636 {
 637 \peek_catcode:NTF \c_math_toggle_token
 638 { \CJKsymbol {#1} \nobreak #2 \c_math_toggle_token }
 639 { \xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2#3 \c_math_toggle_token }
 640 }
 641 {
 642 \peek_meaning_remove:NTF \begin
 643 { \xeCJK_checksinglingle_env:NNNNn {#1} {#2} {#3} \begin }
 644 {
 645 \peek_meaning_remove:NTF \end
 646 { \xeCJK_checksinglingle_env:NNNNn {#1} {#2} {#3} \end }
 647 { \xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2#3 }
 648 }
 649 }
 650 }
 651 \cs_new_nopar:Nn \xeCJK_checksinglingle_env:NNNNn
 652 {
 653 \clist_if_in:NnTF \g_xeCJK_inline_env_clist {#5}
 654 { \xeCJK_checksinglingle_save_CJKsymbol:N {#1} #2#3 }
 655 { \CJKsymbol {#1} \nobreak #2 }
 656 \scan_stop: #4 {#5}
 657 }
 (End definition for \xeCJK_checksinglingle_env:NNN and \xeCJK_checksinglingle_env:NNNNn.)

\g_xeCJK_inline_env_clist

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

InlineEnv

InlineEnv+ 659 \keys_define:nn { xeCJK / options }

InlineEnv- 660 {
 661 InlineEnv .clist_gset:N = \g_xeCJK_inline_env_clist ,
 662 InlineEnv+ .code:n =
 663 {
 664 \clist_gput_right:Nx \g_xeCJK_inline_env_clist {#1}
 665 \clist_gremove_duplicates:N \g_xeCJK_inline_env_clist
 666 },
 667 InlineEnv- .code:n =
 668 {
 669 \clist_map_inline:xn {#1}
 670 { \clist_gremove_all:Nn \g_xeCJK_inline_env_clist {##1} }
 671 },
 672 }

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

5.4 增加 CJK 子分区

```
\xeCJK_UL_subclass_patch_tl
\g_xeCJK_CJK_subclass_clist 673 \tl_new:N \xeCJK_UL_subclass_patch_tl
674 \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: 675 \cs_new_nopar:Nn \xeCJK_erase_CJKsymbol:
676 {
677   \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
678   \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
679 }
680 \cs_new_nopar:Nn \xeCJK_restore_CJKsymbol:
681 {
682   \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block
683   \CJKsymbol
684 }
(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。
685 \NewDocumentCommand \xeCJKDeclareSubCJKBlock { m m }
686 { \xeCJKDeclareSubCharClass { CJK } {#1} {#2} }
687 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock. This function is documented on page 10.)

\xeCJKCancelSubCJKBlock 取消和恢复对 CJK 子区的声明。
\xeCJKRestoreSubCJKBlock 688 \bool_new:N \l_xeCJK_sub_cancel_bool
689 \NewDocumentCommand \xeCJKCancelSubCJKBlock { m }
690 {
691   \bool_if:NF \l_xeCJK_sub_cancel_bool
692   {
693     \bool_set_true:N \l_xeCJK_sub_cancel_bool
694     \xeCJK_sub_restore_or_cancel:n {#1}
695   }
696 }
697 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { m }
698 {
699   \bool_if:NT \l_xeCJK_sub_cancel_bool
700   {
701     \bool_set_false:N \l_xeCJK_sub_cancel_bool
702     \xeCJK_sub_restore_or_cancel:n {#1}
703   }
704 }
705 \cs_new_nopar:Nn \xeCJK_sub_restore_or_cancel:n
706 {
707   \clist_map_inline:xn {#1}
708   {
709     \cs_if_free:cTF { xeCJK_ CJK/##1 _class }
710     { \xeCJK_error:nx { SubBlock-undefined } {##1} }
711     {
712       \xeCJKDeclareCharClass
```

```

713             { CJK \bool_if:NF \l_xeCJK_sub_cancel_bool { /##1 } }
714             { \clist_use:c { g_xeCJK_CJK/##1_range_clist } }
715         }
716     }
717 }
718 \xeCJK_msg_new:nn { SubBlock-undefined }
719 {
720     The~CJK~sub~block~'#1'~is~undefined.\\
721     Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \c_space_tl
722     to~declare~it.
723 }

```

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

\xeCJKDeclareSubCharClass

```

724 \NewDocumentCommand \xeCJKDeclareSubCharClass
725 { > { \TrimSpaces } m > { \TrimSpaces } m m }
726 {
727     \cs_if_free:cT { xeCJK_ #1/#2 _class }
728     {
729         \xeCJK_new_class:n { #1/#2 }
730         \xeCJK_set_sub_class_toks:nn {#1} {#2}
731         \xeCJK_new_sub_key:n {#2}
732         \xeCJK_UL_subclass_patch:nn {#1} {#2}
733     }
734     \xeCJKDeclareCharClass { #1/#2 } {#3}
735 }
736 \@onlypreamble \xeCJKDeclareSubCharClass

```

(End definition for \xeCJKDeclareSubCharClass.)

\xeCJK_set_sub_class_toks:nn

```

737 \cs_new_nopar:Nn \xeCJK_set_sub_class_toks:nn
738 {
739     \clist_map_inline:Nn \g_xeCJK_base_class_clist
740     {
741         \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
742         \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
743         \str_if_eq:xxTF {##1} { CJK }
744         {
745             \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
746             { \xeCJK_switch_font:nn {#1} {#2} }
747         }
748         {
749             \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
750             { \xeCJK_erase_CJKsymbol: }
751             \xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
752             { \xeCJK_switch_font:nn {#1} {#2} \xeCJK_restore_CJKsymbol: }
753         }
754     }
755     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
756     \clist_if_empty:NF \g_xeCJK_CJK_subclass_clist
757     {

```

```

758 \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
759 {
760   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
761   \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
762   \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
763   { \xeCJK_switch_font:nn {#2} {##1} }
764   \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
765   { \xeCJK_switch_font:nn {##1} {#2} }
766 }
767 }
768 \clist_gput_right:Nn \g_xeCJK_CJK_subclass_clist {#2}
769 \xeCJK_save_CJK_class:n { #1/#2 }
770 \clist_map_inline:nn { CJK, FullLeft, FullRight }
771 {
772   \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
773   { \xeCJK_switch_font:nn {#2} {#1} }
774 }
775 }

```

(End definition for \xeCJK_set_sub_class_toks:nn.)

\xeCJK_UL_subclass_patch:nn

```

776 \cs_new_nopar:Nn \xeCJK_UL_subclass_patch:nn
777 {
778   \tl_put_right:Nn \xeCJK_UL_subclass_patch_tl
779   {
780     \clist_map_inline:nn { Default, HalfLeft, HalfRight }
781     {
782       \str_if_eq:xxTF {##1} { HalfLeft }
783       { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKecglue \CJKsymbol } }
784       { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
785       \xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
786       {
787         \xeCJK_erase_CJKsymbol:
788         \xeCJK_Boundary_and_CJK:
789         \xeCJK_switch_font:nn {#1} {#2}
790         \xeCJK_restore_CJKsymbol:
791       }
792     }
793   }
794 }

```

(End definition for \xeCJK_UL_subclass_patch:nn.)

5.5 标点处理

\g_xeCJK_punctstyle_clist 标点处理方式。

```

795 \clist_new:N \g_xeCJK_punctstyle_clist
796 \clist_set:Nn \g_xeCJK_punctstyle_clist
797 { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
798 \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
799 { \tl_const:cn { c_xeCJK_ps_#1_tl } {#1} }

```

(End definition for \g_xeCJK_punctstyle_clist.)

```

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

PunctStyle 相关选项声明。
AllowBreakBetweenPuncts 803 \keys_define:nn { xeCJK / options }
KaiMingPunct 804 {
  LongPunct 805   AllowBreakBetweenPuncts .choice:,
  MiddlePunct 806   AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
  PunctWidth 807   AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
                        808   AllowBreakBetweenPuncts .default:n = { true },
                        809   KaiMingPunct .code:n = { \xeCJK_set_special_punct:nn { mixedwidth } {#1} },
                        810   KaiMingPunct+ .code:n = { \xeCJK_add_special_punct:nn { mixedwidth } {#1} },
                        811   KaiMingPunct- .code:n = { \xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
                        812   LongPunct .code:n = { \xeCJK_set_special_punct:nn { long } {#1} },
                        813   LongPunct+ .code:n = { \xeCJK_add_special_punct:nn { long } {#1} },
                        814   LongPunct- .code:n = { \xeCJK_sub_special_punct:nn { long } {#1} },
                        815   MiddlePunct+ .code:n = { \xeCJK_add_special_punct:nn { middle } {#1} },
                        816   MiddlePunct .code:n = { \xeCJK_set_special_punct:nn { middle } {#1} },
                        817   MiddlePunct- .code:n = { \xeCJK_sub_special_punct:nn { middle } {#1} },
                        818   PunctWidth .tl_gset:N = \g_xeCJK_punct_width_tl ,
                        819   PunctStyle .choice_code:n =
                        820   {
                        821     \tl_set:Nx \l_xeCJK_punctstyle_tl { \l_keys_choice_tl }
                        822     \tl_if_eq:NNT \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
                        823     { \xeCJKallowbreakbetweenpuncts }
                        824   },
                        825   PunctStyle .generate_choices:V = \g_xeCJK_punctstyle_clist,
                        826   PunctStyle / banjiao .meta:n = { PunctStyle = halfwidth },
                        827   PunctStyle / quanjiao .meta:n = { PunctStyle = fullwidth },
                        828   PunctStyle / kaiming .meta:n = { PunctStyle = mixedwidth },
                        829   PunctStyle / hangmobanjiao .meta:n = { PunctStyle = marginkerning },
                        830   PunctStyle / unknown .code:n =
                        831   { \xeCJK_error:nx { PunctStyle-undefined } \l_keys_value_tl },
                        832   }
                        833 \xeCJK_msg_new:nn { PunctStyle-undefined }
                        834 {
                        835   Punctstyle~"#1"~is~not~available. \\\
                        836   You~can~only~use~one~of\\\
                        837   "\g_xeCJK_punctstyle_clist".\\
                        838   }
                        (End definition for PunctStyle and others. These functions are documented on page 5.)

\xeCJKallowbreakbetweenpuncts
\xeCJKnobreakbetweenpuncts 839 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
                        840 { \cs_set_eq:NN \xeCJK_punct_nobreak: \xeCJK_zero_glue: }
                        841 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
                        842 { \cs_set_eq:NN \xeCJK_punct_nobreak: \nobreak }
                        843 \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 844 \clist_set:Nn \g_xeCJK_special_ps_clist { mixedwidth , long , middle }
\XeCJK_add_special_punct:nn 845 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
\XeCJK_sub_special_punct:nn 846 {
847     \tl_new:c { l_xeCJK_#1_punct_tl }
848     \prop_new:c { l_xeCJK_#1_punct_prop }
849 }
850 \cs_new_nopar:Nn \XeCJK_set_special_punct:nn
851 {
852     \tl_set:cx { l_xeCJK_#1_punct_tl } {#2}
853     \prop_clear:c { l_xeCJK_#1_punct_prop }
854     \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
855     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
856 }
857 \cs_new_nopar:Nn \XeCJK_add_special_punct:nn
858 {
859     \tl_put_right:cx { l_xeCJK_#1_punct_tl } {#2}
860     \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
861     { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
862 }
863 \cs_new_nopar:Nn \XeCJK_sub_special_punct:nn
864 {
865     \tl_map_inline:xn {#2}
866     {
867         \tl_remove_all:cn { l_xeCJK_#1_punct_tl } {##1}
868         \prop_del:cn { l_xeCJK_#1_punct_prop } {##1}
869     }
870 }
871 \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 872 \cs_generate_variant:Nn \dim_set:Nn { Nc }
\dim_set_min:Nc 873 \cs_generate_variant:Nn \dim_add:Nn { Nc }
874 \cs_generate_variant:Nn \dim_set_min:Nn { Nc }
(End definition for \dim_set:Nc, \dim_add:Nc, and \dim_set_min:Nc.)

\l_xeCJK_punct_coor_tl 记录当前字体状态下的标点处理方式。
875 \tl_set:Nn \l_xeCJK_punct_coor_tl { \l_xeCJK_font_coor_tl/\l_xeCJK_punctstyle_tl }
(End definition for \l_xeCJK_punct_coor_tl.)

\XeCJK_get_punct_bounds:nN #1 为 l/r, #2 为标点字符, 返回标点的实际左/右空白的负值和根据标点处理方式决定添加的左/右空白。全角方式直接添加标点的实际左/右空白; 半角方式添加标点左右空白中的最小值; 开明方式与半角相同, 只在开明标点再增加实际左/右空白的一半。
876 \cs_new_nopar:Nn \XeCJK_get_punct_bounds:nN
877 {
878     \tl_if_exist:cF { \l_xeCJK_punct_coor_tl/rule/#1/#2 }
879     {
880         \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
881         {
882             \clist_map_inline:nn { glue , rule , bound }
883             { \tl_gset:cV { \l_xeCJK_punct_coor_tl/##1/#1/#2 } \c_zero_dim }

```

```

884 \tl_gset:cV { \l_xeCJK_punct_coor_tl/rule/m/#2 } \c_zero_dim
885 }
886 {
887 \tl_if_exist:cF { \l_xeCJK_punct_coor_tl/bound/#1/#2 }
888 { { \xeCJK_select_font: \xeCJK_get_punct_dimen:N {#2} } }
889 \xeCJK_punct_if_long:NTF {#2}
890 {
891 \dim_zero:N \l_tmpa_dim
892 \dim_zero:N \l_tmpb_dim
893 \dim_zero:N \l_tmpc_dim
894 }
895 {
896 \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/#1/#2 }
897 \dim_set:Nc \l_tmpb_dim
898 { \l_xeCJK_punct_coor_tl/bound/\str_if_eq:xxTF {#1} lrl/#2 }
899 \dim_set:Nn \l_tmpc_dim
900 {
901 \cs_if_exist_use:cF { g_xeCJK_punct_width/#2 }
902 {
903 \tl_if_blank:xTF \g_xeCJK_punct_width_tl
904 { \c_zero_dim }
905 { \g_xeCJK_punct_width_tl }
906 }
907 }
908 \xeCJK_punct_if_middle:NTF {#2}
909 {
910 \dim_set:Nn \l_tmpc_dim
911 {
912 \dim_compare:nNnTF \l_tmpc_dim > \c_zero_dim
913 { \l_tmpc_dim }
914 {
915 \prg_case_tl:Nnn \l_xeCJK_punctstyle_tl
916 {
917 \c_xeCJK_ps_halfwidth_tl { .5 em }
918 \c_xeCJK_ps_mixedwidth_tl { .5 em }
919 \c_xeCJK_ps_CCT_tl { .7 em }
920 \c_xeCJK_ps_fullwidth_tl { 1 em }
921 }
922 { 1 em }
923 }
924 - \tl_use:c { \l_xeCJK_punct_coor_tl/dimen/#2 }
925 }
926 \dim_set:Nn \l_tmpc_dim { .5 \l_tmpc_dim }
927 }
928 {
929 \dim_compare:nNnTF \l_tmpc_dim > \c_zero_dim
930 {
931 \dim_set:Nn \l_tmpc_dim
932 {
933 \l_tmpc_dim - \l_tmpb_dim
934 - \tl_use:c { \l_xeCJK_punct_coor_tl/dimen/#2 }
935 }

```

```

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

```

\xeCJKsetwidth

```

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

```

\xeCJK_punct_if_right_p:N

\xeCJK_punct_if_both_left_right_p:N

\xeCJK_punct_if_long_p:N

\xeCJK_punct_if_middle_p:N

\xeCJK_punct_if_mixedwidth_p:N

\xeCJK_punct_if_right:N

\xeCJK_punct_if_both_left_right:N

\xeCJK_punct_if_long:N

\xeCJK_punct_if_middle:N

\xeCJK_punct_if_mixedwidth:N

```

975 \prg_new_conditional:Nnn \xeCJK_punct_if_right:N { p , T , F , TF }
976 {
977     \tl_if_exist:cTF { \l_xeCJK_punct_coors_tl/glue/r/#1 }
978     \prg_return_true: \prg_return_false:
979 }
980 \prg_new_conditional:Nnn \xeCJK_punct_if_both_left_right:N { p , T , F , TF }
981 {

```

```

982     \bool_if:nTF
983     {
984         ( \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/r/#1 } &&
985           \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/r/#2 } ) ||
986         ( \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/l/#1 } &&
987           \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/l/#2 } )
988     }
989     \prg_return_true: \prg_return_false:
990 }
991 \cs_generate_variant:Nn \prg_new_conditional:Nnn { c }
992 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
993 {
994     \prg_new_conditional:cnn { xeCJK_punct_if_#1:N } { p , T , F , TF }
995     {
996         \prop_if_in:cnTF { l_xeCJK_#1_punct_prop } {##1}
997         \prg_return_true: \prg_return_false:
998     }
999 }

```

(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。

```

1000 \cs_new_nopar:Nn \xeCJK_get_kern:NN
1001 {
1002     \tl_if_exist:cF { \l_xeCJK_punct_coord_tl/kern/#1-#2 }
1003     {
1004         \dim_zero:N \l_tmpa_dim
1005         \tl_if_eq:NnF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1006         {
1007             \tl_if_exist:cTF { g_xeCJK_punct/kern/#1-#2 }
1008             { \dim_set:Nc \l_tmpa_dim { g_xeCJK_punct/kern/#1-#2 } }
1009             { \xeCJK_calc_kern:NN {#1} {#2} }
1010         }
1011         \tl_gset:cV { \l_xeCJK_punct_coord_tl/kern/#1-#2 } \l_tmpa_dim
1012     }
1013 }
1014 \cs_new_nopar:Nn \xeCJK_calc_kern:NN
1015 {
1016     \tl_if_eq:NnTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_marginkerning_tl
1017     {
1018         \xeCJK_punct_if_right:NT {#1}
1019         { \dim_add:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/r/#1 } }
1020         \xeCJK_punct_if_right:NF {#2}
1021         { \dim_add:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#2 } }
1022     }
1023     {
1024         \tl_if_eq:NnTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_fullwidth_tl
1025         { \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1.5 em } {#1} {#2} }
1026         {
1027             \xeCJK_punct_if_both_left_right:NnTF {#1} {#2}
1028             {
1029                 \xeCJK_punct_if_mixedwidth:NT {#1}

```

```

1030         {
1031             \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1 em } {#1} {#2}
1032             \dim_compare:nNnT \l_tmpa_dim < { .1 em }
1033             { \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#1 } }
1034         }
1035     }
1036     {
1037         \bool_if:nTF
1038         {
1039             \xeCJK_punct_if_mixedwidth_p:N {#1} &&
1040             ! ( \tl_if_eq_p:NN \l_xeCJK_punctstyle_tl \c_xeCJK_ps_halfwidth_tl )
1041         }
1042         {
1043             \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/glue/r/#1 }
1044             \dim_set:Nn \l_tmpa_dim { .7 \l_tmpa_dim }
1045         }
1046         {
1047             \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1 em } {#1} {#2}
1048             \dim_compare:nNnT \l_tmpa_dim < { .1 em }
1049             {
1050                 \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#1 }
1051                 \dim_set_min:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/r/#1 }
1052                 \dim_set:Nc \l_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/l/#2 }
1053                 \dim_set_min:Nc \l_tmpb_dim { \l_xeCJK_punct_coord_tl/bound/r/#2 }
1054                 \dim_set_max:Nn \l_tmpa_dim \l_tmpb_dim
1055             }
1056         }
1057     }
1058 }
1059 \bool_if:nT
1060 { \xeCJK_punct_if_long_p:N {#1} || \xeCJK_punct_if_long_p:N {#2} }
1061 { \dim_set_max:Nn \l_tmpa_dim { .1 em } }
1062 }
1063 \dim_set_max:Nn \l_tmpa_dim \c_zero_dim
1064 }
1065 \cs_generate_variant:Nn \xeCJK_get_kern:NN { V }
(End definition for \xeCJK_get_kern:NN and \xeCJK_calc_kern:NN.)

```

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

```

1066 \cs_new_nopar:Nn \xeCJK_calc_kern:NnNN
1067 {
1068     \dim_set:Nn #1
1069     {
1070         #2
1071         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1072             \xeCJK_punct_if_right:NNT {#3} { bound } { glue } /l/#3 }
1073         - \tl_use:c { \l_xeCJK_punct_coord_tl/
1074             \xeCJK_punct_if_right:NNT {#4} { glue } { bound } /r/#4 }
1075         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#3 }
1076         - \tl_use:c { \l_xeCJK_punct_coord_tl/dimen/#4 }
1077         \xeCJK_punct_if_both_left_right:NNT {#3} {#4}

```

```

1078         { + \tl_use:c { \l_xeCJK_punct_coord_tl/rule/m/#4 } }
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。

```

1081 \cs_new_nopar:Nn \xeCJK_get_punct_dimen:N
1082 {
1083     \xeCJK_get_glyph_bounds:nNN { left } {#1} \l_tmpa_dim
1084     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/l/#1 } \l_tmpa_dim
1085     \dim_set_eq:NN \l_tmpb_dim \l_tmpa_dim
1086     \xeCJK_get_glyph_bounds:nNN { right } {#1} \l_tmpa_dim
1087     \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/r/#1 } \l_tmpa_dim
1088     \dim_add:Nn \l_tmpb_dim \l_tmpa_dim
1089     \tl_gset:cx { \l_xeCJK_punct_coord_tl/dimen/#1 }
1090     { \dim_eval:n { \tex_fontcharwd:D \tex_font:D `#1 - \l_tmpb_dim } }
1091     \bool_if:nT
1092     {
1093         \xeCJK_punct_if_long_p:N {#1} &&
1094         ! ( \str_if_eq_p:xx {#1} { \cdots } || \str_if_eq_p:xx {#1} { \dots } )
1095     }
1096     {
1097         \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
1098         {
1099             \str_if_eq:xxF \c_xeCJK_ps_plain_tl {##1}
1100             {
1101                 \tl_gset:cx { \l_xeCJK_font_coord_tl/##1/kern/#1-#1 }
1102                 { - \dim_use:N \l_tmpb_dim }
1103             }
1104         }
1105     }
1106 }
(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。

```

1107 \cs_new_nopar:Nn \xeCJK_get_glyph_bounds:nNN
1108 {
1109     \dim_set:Nn #3
1110     {
1111         \XeTeXglyphbounds
1112         \prg_case_str:xxn {#1}
1113         {
1114             { left } { \c_one } { top } { \c_two }
1115             { right } { \c_three } { bottom } { \c_four }
1116         }
1117         { \c_three }
1118         \int_eval:n { \XeTeXcharglyph `#2 }
1119     }
1120 }
(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。

```
1121 \NewDocumentCommand \xeCJKsetkern { m m m }
1122 { \tl_gset:cn { g_xeCJK_punct/kern/#1-#2 } {#3} }
(End definition for \xeCJKsetkern. This function is documented on page 11.)
```

5.6 后备字体

`AutoFallback` 后备字体的宏包选项声明。

```
1123 \keys_define:nn { xeCJK / options }
1124 {
1125   AutoFallback .choice:,
1126   AutoFallback / true .code:n = { \xeCJKenablefallback },
1127   AutoFallback / false .code:n = { \xeCJKdisablefallback },
1128   AutoFallback .default:n = { true },
1129   fallback .meta:n = { AutoFallback = true },
1130 }
(End definition for AutoFallback. This function is documented on page 4.)
```

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

```
\xeCJKdisablefallback 1131 \bool_new:N \l_xeCJK_fallback_bool
1132 \NewDocumentCommand \xeCJKenablefallback { }
1133 {
1134   \bool_if:NF \l_xeCJK_fallback_bool
1135   {
1136     \bool_set_true:N \l_xeCJK_fallback_bool
1137     \cs_set_eq:NN \xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1138     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_testsymbol:N
1139   }
1140 }
1141 \NewDocumentCommand \xeCJKdisablefallback { }
1142 {
1143   \bool_if:NT \l_xeCJK_fallback_bool
1144   {
1145     \bool_set_false:N \l_xeCJK_fallback_bool
1146     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_save_CJKsymbol:N
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。

```
1149 \cs_new_nopar:Nn \xeCJK_fallback_testsymbol:N
1150 {
1151   \font_glyph_if_exist:NnTF \tex_font:D {`#1}
1152   { \xeCJK_fallback_save_CJKsymbol:N {#1} }
1153   {
1154     \xeCJK_family_if_exist:nTF { \xeCJK@family/FallBack }
1155     { {
1156       \tl_put_right:Nn \xeCJK@family { /FallBack }
1157       \xeCJK_select_font:
1158       \xeCJK_fallback_testsymbol:N {#1}
1159     } }
1160   }
}
```

```

1160      {
1161        \xeCJK_warning:nx { fallback } {#1}
1162        \xeCJK_fallback_save_CJKsymbol:N {#1}
1163      }
1164    }
1165  }
1166  \xeCJK_msg_new:nn { fallback }
1167  {
1168    CJKfamily~'\xeCJK@family'~
1169    ( \prop_get:NV \g_xeCJK_family_prop \xeCJK@family )~
1170    does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {'#1'}).\
1171  }
  (End definition for \xeCJK_fallback_testsymbol:N.)

```

\setCJKfallbackfamilyfont

```

\xecjk_set_family_fallback:nnn 1172 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1173 {
1174   \exp_args:Nx \tl_if_in:nnTF {#3} { , }
1175   { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} }
1176   { \xeCJK_set_family:nnn { #1/FallBack } {#2} {#3} }
1177 }
1178 \cs_new_nopar:Nn \xeCJK_set_family_fallback:nnn
1179 {
1180   \group_begin:
1181   \tl_set:Nx \l_xeCJK_family_fb_tl {#1}
1182   \prop_get:NVNF \g_xeCJK_family_prop \l_xeCJK_family_fb_tl \l_xeCJK_fontname_tl
1183   { \tl_clear:N \l_xeCJK_fontname_tl }
1184   \clist_map_inline:xn {#3}
1185   {
1186     \tl_put_right:Nn \l_xeCJK_family_fb_tl { /FallBack }
1187     \xeCJK_get_sub_features:nn \l_xeCJK_family_fb_tl {##1}
1188     \clist_put_left:cx { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist } {#2}
1189     \xeCJK_set_family:ncc \l_xeCJK_family_fb_tl
1190     { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist }
1191     { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontname_tl }
1192   }
1193   \group_end:
1194 }

```

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

5.7 CJK 字体族声明方式

```

1195 \bool_new:N \g_xeCJK_AutoFakeBold_bool
1196 \bool_new:N \g_xeCJK_AutoFakeSlant_bool
1197 \fp_new:N \g_xeCJK_EmboldenFactor_fp
1198 \fp_new:N \g_xeCJK_SlantFactor_fp

```

AutoFakeBold 伪粗体和伪斜体的宏包选项声明。

```

AutoFakeSlant 1199 \keys_define:nn { xeCJK / options }
EmboldenFactor 1200 {
SlantFactor 1201   AutoFakeBold .choice:,
1202   AutoFakeBold / true .code:n =

```

```

1203     { \bool_set_true:N \g_xeCJK_AutoFakeBold_bool },
1204     AutoFakeBold / false .code:n =
1205     { \bool_set_false:N \g_xeCJK_AutoFakeBold_bool },
1206     AutoFakeBold / unknown .code:n =
1207     {
1208         \bool_set_true:N \g_xeCJK_AutoFakeBold_bool
1209         \fp_set:Nn \g_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1210     },
1211     AutoFakeBold .default:n = { true },
1212     AutoFakeSlant .choice:,
1213     AutoFakeSlant / true .code:n =
1214     { \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool },
1215     AutoFakeSlant / false .code:n =
1216     { \bool_set_false:N \g_xeCJK_AutoFakeSlant_bool },
1217     AutoFakeSlant / unknown .code:n =
1218     {
1219         \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool
1220         \fp_set:Nn \g_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1221     },
1222     AutoFakeSlant .default:n = { true },
1223     EmboldenFactor .fp_set:N = \g_xeCJK_EmboldenFactor_fp,
1224     SlantFactor .fp_set:N = \g_xeCJK_SlantFactor_fp,
1225     BoldFont .meta:n = { AutoFakeBold = true },
1226     boldfont .meta:n = { AutoFakeBold = true },
1227     SlantFont .meta:n = { AutoFakeSlant = true },
1228     slantfont .meta:n = { AutoFakeSlant = true },
1229 }

```

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

\keys_set_known:nxN

```

1230 \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 1231 \clist_new:N \g_xeCJK_sub_key_clist
1232 \cs_new_nopar:Nn \xeCJK_new_sub_key:n
1233 {
1234     \clist_gput_right:Nx \g_xeCJK_sub_key_clist {#1}
1235     \keys_define:nn { xeCJK / features }
1236     {
1237         #1 .code:n =
1238         {
1239             \tl_if_blank:xTF {##1}
1240             {
1241                 \bool_set_false:c { l_xeCJK_copy_#1_bool }
1242                 \bool_set_false:c { l_xeCJK_add_#1_bool }
1243                 \tl_put_right:Nx \l_xeCJK_family_tl { /#1 }
1244             }
1245             {
1246                 \clist_put_right:Nx \l_xeCJK_sub_key_clist {#1}
1247                 \str_if_eq:xxTF {##1} *
1248                 {

```

```

1249         \bool_set_true:c { l_xeCJK_copy_#1_bool }
1250         \bool_set_false:c { l_xeCJK_add_#1_bool }
1251     }
1252     {
1253         \bool_set_false:c { l_xeCJK_copy_#1_bool }
1254         \bool_set_true:c { l_xeCJK_add_#1_bool }
1255         \xeCJK_get_sub_features:nn {#1} {##1}
1256     }
1257 }
1258 },
1259 #1 .default:n = \c_empty_tl,
1260 }
1261 }
1262 \cs_generate_variant:Nn \xeCJK_new_sub_key:n { x }
1263 \cs_new_nopar:Nn \xeCJK_get_sub_features:nn
1264 {
1265     \tl_set:Nx \l_tmpa_tl {#2}
1266     \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1267     \xeCJK_gobble_brace:N \l_tmpa_tl
1268     \exp_args:No \tl_if_head_eq_charcode:nNTF \l_tmpa_tl [ % ]
1269     {
1270         \exp_after:wN \xeCJK_get_sub_features_aux:n \l_tmpa_tl \c_empty_tl
1271         \tl_if_blank:xT \l_xeCJK_sub_fontname_tl
1272         {
1273             \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_tmpa_tl
1274             \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1275         }
1276     }
1277     { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_tmpa_tl }
1278     \tl_if_blank:xTF \l_xeCJK_sub_fontname_tl
1279     { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_fontname_tl }
1280     { \tl_replace_all:Nnx \l_xeCJK_sub_fontname_tl * \l_xeCJK_fontname_tl }
1281     \clist_set:cx { l_xeCJK_#1_fontfeat_clist } \l_xeCJK_sub_fontfeat_clist
1282     \tl_set:cx { l_xeCJK_#1_fontname_tl } \l_xeCJK_sub_fontname_tl
1283 }

```

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

```

\xeCJK_gobble_brace:N
\xeCJK_get_sub_features:nn 1284 \cs_new_nopar:Nn \xeCJK_gobble_brace:N
\xeCJK_get_sub_features_aux:n 1285 {
1286     \exp_args:No \tl_if_head_group:nT #1
1287     {
1288         \exp_last_unbraced:NNo \tl_set:Nn #1 #1
1289         \xeCJK_gobble_brace:N #1
1290     }
1291 }
1292 \cs_new_nopar:Npn \xeCJK_get_sub_features_aux:n [#1] #2
1293 {
1294     \clist_set:Nx \l_xeCJK_sub_fontfeat_clist {#1}
1295     \tl_set:Nx \l_xeCJK_sub_fontname_tl {#2}
1296 }

```

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

FallBack

```
1297 \xeCJK_new_sub_key:n { FallBack }  
(End definition for FallBack. This function is documented on page 7.)
```

.clist_set_x:N

```
.clist_set_x:c 1298 \cs_set_protected:cpn { \c_keys_props_root_tl .clist_set_x:N } #1  
1299 { \keys_variable_set:NnN #1 { clist } x }  
1300 \cs_set_protected:cpn { \c_keys_props_root_tl .clist_set_x:c } #1  
1301 { \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 1302 \cs_new_nopar:Nn \xeCJK_map_features_id:n  
    ItalicFont 1303 { \prop_get:Nn \g_xeCJK_features_id_prop { #1 } }  
    BoldItalicFont 1304 \prop_new:N \g_xeCJK_features_id_prop  
    SlantedFont 1305 \prop_put:Nnn \g_xeCJK_features_id_prop { bf } { Bold }  
    BoldSlantedFont 1306 \prop_put:Nnn \g_xeCJK_features_id_prop { it } { Italic }  
    BoldFeatures 1307 \prop_put:Nnn \g_xeCJK_features_id_prop { bfit } { BoldItalic }  
    ItalicFeatures 1308 \prop_put:Nnn \g_xeCJK_features_id_prop { sl } { Slanted }  
    BoldItalicFeatures 1309 \prop_put:Nnn \g_xeCJK_features_id_prop { bfs1 } { BoldSlanted }  
    SlantedFeatures 1310 \prop_map_inline:Nn \g_xeCJK_features_id_prop  
    BoldSlantedFeatures 1311 {  
1312     \keys_define:nn { xeCJK / features }  
1313     {  
1314         #2Font .tl_set_x:c = { l_xeCJK_fontname_#1_tl },  
1315         #2Features .clist_set_x:c = { l_xeCJK_fontfeat_#1_clist } ,  
1316     }  
1317 }  
(End definition for \g_xeCJK_features_id_prop and others.)
```

AutoFakeBold

```
AutoFakeSlant 1318 \keys_define:nn { xeCJK / features }  
    Mono 1319 {  
1320     AutoFakeBold .choice:,  
1321     AutoFakeBold / false .code:n =  
1322     { \bool_set_false:N \l_xeCJK_AutoFakeBold_bool },  
1323     AutoFakeBold / unknown .code:n =  
1324     {  
1325         \bool_set_true:N \l_xeCJK_AutoFakeBold_bool  
1326         \fp_set:Nn \l_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }  
1327     },  
1328     AutoFakeBold .default:n = { \g_xeCJK_EmboldenFactor_fp },  
1329     AutoFakeSlant .choice:,  
1330     AutoFakeSlant / false .code:n =  
1331     { \bool_set_false:N \l_xeCJK_AutoFakeSlant_bool },  
1332     AutoFakeSlant / unknown .code:n =  
1333     {  
1334         \bool_set_true:N \l_xeCJK_AutoFakeSlant_bool  
1335         \fp_set:Nn \l_xeCJK_SlantFactor_fp { \l_keys_value_tl }  
1336     },  
1337     AutoFakeSlant .default:n = { \g_xeCJK_SlantFactor_fp },  
1338     Mono .choice:,
```

```

1339 Mono / Exspace .code:n = { \xeCJK_set_monoexspace: },
1340 Mono / Scale .code:n =
1341 {
1342     \xeCJK_set_monoscale:
1343     \tl_set:Nx \l_xeCJK_monoscale_tl
1344     { Scale = { \fp_use:N \g_xeCJK_monoscale_fp } }
1345 }
1346 }

```

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

\xeCJK_set_init:

```

1347 \cs_new_nopar:Nn \xeCJK_set_init:
1348 {
1349     \tl_clear:N \l_xeCJK_monoscale_tl
1350     \int_gincr:N \g_xeCJK_fontspec_int
1351     \prop_clear:N \l_xeCJK_add_font_prop
1352     \prop_map_inline:Nn \g_xeCJK_features_id_prop
1353     {
1354         \tl_clear:c { l_xeCJK_fontname_##1_tl }
1355         \clist_clear:c { l_xeCJK_fontfeat_##1_clist }
1356     }
1357     \clist_clear:N \l_xeCJK_pass_features_clist
1358     \clist_clear:N \l_xeCJK_sub_key_clist
1359     \clist_map_inline:Nn \g_xeCJK_sub_key_clist
1360     {
1361         \bool_set_false:c { l_xeCJK_copy_##1_bool }
1362         \bool_set_false:c { l_xeCJK_add_##1_bool }
1363     }
1364     \bool_set_eq:NN \l_xeCJK_AutoFakeBold_bool \g_xeCJK_AutoFakeBold_bool
1365     \bool_set_eq:NN \l_xeCJK_AutoFakeSlant_bool \g_xeCJK_AutoFakeSlant_bool
1366     \fp_set_eq:NN \l_xeCJK_EmboldenFactor_fp \g_xeCJK_EmboldenFactor_fp
1367     \fp_set_eq:NN \l_xeCJK_SlantFactor_fp \g_xeCJK_SlantFactor_fp
1368 }

```

(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。

```

1369 \cs_new_protected_nopar:Nn \xeCJK_set_family:nnn
1370 {
1371     \group_begin:
1372     \xeCJK_set_init:
1373     \tl_set:Nx \l_xeCJK_family_tl {#1}
1374     \clist_set:Nx \l_xeCJK_fontoptions_clist {#2}
1375     \tl_set:Nx \l_xeCJK_fontname_tl {#3}
1376     \prop_gpop:NVT \g_xeCJK_family_name_prop \l_xeCJK_family_tl \l_tmpa_tl
1377     { \xeCJK_warning:nxx { CJKfamily-redef } \l_xeCJK_family_tl \l_tmpa_tl }
1378     \keys_set_known:nxN { xeCJK / features }
1379     { \g_xeCJK_default_features_clist, #2 } \l_xeCJK_pass_features_clist
1380     \xeCJK_parse_features:
1381     \xeCJK_pass_features:
1382     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1383     {
1384         \exp_not:n { \fontspec_set_family:Nnn \l_xeCJK_fontspec_family_tl }

```

```

1385     { \l_xeCJK_pass_features_clist } { \l_xeCJK_fontname_tl }
1386     \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1387     { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_fontspec_family_tl
1388     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1389     \exp_not:n
1390     { {
1391         \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1392         \exp_not:N \fontfamily { \l_xeCJK_fontspec_family_tl }
1393         \exp_not:N \selectfont
1394     } }
1395     }
1396     \xeCJK_save_family_info:
1397     \xeCJK_add_sub_block:
1398     \group_end:
1399     }
1400     \cs_generate_variant:Nn \xeCJK_set_family:nnn { ncc }
1401     \xeCJK_msg_new:nn { CJKfamily-redef } { Redefining~CJKfamily~`#1'~(#2). }
1402     (End definition for \xeCJK_set_family:nnn.)

```

\xeCJK_add_font:nn

\xeCJK_add_font_if_new:nn

```

1402     \cs_new_nopar:Nn \xeCJK_add_font:nn
1403     {
1404         \prop_put:Nxx \l_xeCJK_add_font_prop
1405         { \xeCJK_map_features_id:n {#1} Font } {#2}
1406     }
1407     \cs_new_nopar:Nn \xeCJK_add_font_if_new:nn
1408     {
1409         \prop_put_if_new:Nxx \l_xeCJK_add_font_prop
1410         { \xeCJK_map_features_id:n {#1} Font } {#2}
1411     }
1412     \cs_generate_variant:Nn \xeCJK_add_font:nn { nx, nc, nV, nv }
1413     \cs_generate_variant:Nn \xeCJK_add_font_if_new:nn { nx, nc, nV, nv }
1414     (End definition for \xeCJK_add_font:nn and \xeCJK_add_font_if_new:nn.)
1415     \cs_generate_variant:Nn \prop_gpop:NnNT { NV }
1416     \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1417     \cs_generate_variant:Nn \prop_get:Nn { NV }
1418     \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1419     \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
1420     \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1421     \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

\xeCJK_add_fake_bold:n

\xeCJK_add_fake_slant:n

```

1421     \cs_new_nopar:Nn \xeCJK_add_fake_bold:n
1422     {
1423         \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1424         { FakeBold = { \fp_use:N \l_xeCJK_EmboldenFactor_fp } }
1425     }
1426     \cs_new_nopar:Nn \xeCJK_add_fake_slant:n
1427     {
1428         \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1429         { FakeSlant = { \fp_use:N \l_xeCJK_SlantFactor_fp } }
1430     }

```

(End definition for \xeCJK_add_fake_bold:n and \xeCJK_add_fake_slant:n.)

\xeCJK_parse_features:

```

1431 \cs_new_nopar:Nn \xeCJK_parse_features:
1432 { \prop_map_inline:Nn \g_xeCJK_features_id_prop { \xeCJK_parse_features:n {##1} } }
1433 \cs_new_nopar:Nn \xeCJK_parse_features:n
1434 {
1435   \xeCJK_if_font_select:nTF {#1}
1436   {
1437     \xeCJK_add_font:nv {#1} { l_xeCJK_fontname_#1_tl }
1438     \xeCJK_if_it_or_sl:nTF {#1}
1439     {
1440       \xeCJK_if_font_select:nF {bf#1}
1441       { \xeCJK_add_font_if_new:nv {bf#1} { l_xeCJK_fontname_#1_tl } }
1442     }
1443     {
1444       \str_if_eq:xxT {#1} {bf}
1445       {
1446         \clist_map_inline:nn { it , sl }
1447         {
1448           \xeCJK_if_font_select:nF {bf##1}
1449           { \xeCJK_add_font:nV {bf##1} \l_xeCJK_fontname_bf_tl }
1450         }
1451       }
1452     }
1453   }
1454   { \xeCJK_set_fake:n {#1} }
1455 }

```

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

\xeCJK_if_font_select_p:n

```

1456 \xeCJK_if_it_or_sl_p:n \prg_new_conditional:Nnn \xeCJK_if_font_select:n { p, T, F, TF }
1457 {
1458   \exp_args:Nc
1459   \tl_if_blank:xTF { l_xeCJK_fontname_#1_tl } \prg_return_false: \prg_return_true:
1460 }
1461 \prg_new_conditional:Nnn \xeCJK_if_it_or_sl:n { p, T, F, TF }
1462 {
1463   \bool_if:nTF { \str_if_eq_p:xx { it } {#1} || \str_if_eq_p:xx { sl } {#1} }
1464   \prg_return_true: \prg_return_false:
1465 }

```

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

\xeCJK_set_fake:n

```

1466 \cs_new_nopar:Nn \xeCJK_set_fake:n
1467 {
1468   \str_if_eq:xxTF {#1} { bf }
1469   { \bool_if:NT \l_xeCJK_AutoFakeBold_bool { \xeCJK_add_fake_bold:n {#1} } }
1470   {
1471     \bool_if:NTF \l_xeCJK_AutoFakeSlant_bool
1472     {
1473       \bool_if:nT

```

```

1474         { \xeCJK_if_it_or_sl_p:n {#1} ||
1475         ( \str_if_eq_p:xx {#1} {bfit} && ! ( \xeCJK_if_font_select_p:n {it} ) ) ||
1476         ( \str_if_eq_p:xx {#1} {bfs1} && ! ( \xeCJK_if_font_select_p:n {sl} ) )
1477         }
1478         { \xeCJK_add_fake_slant:n {#1} }
1479     }
1480     { \xeCJK_if_it_or_sl:nT {#1} { \xeCJK_map_it_sl:n {#1} } }
1481 \bool_if:nT
1482     {
1483         \l_xeCJK_AutoFakeBold_bool
1484         && ! ( \xeCJK_if_it_or_sl_p:n {#1} )
1485         && ! ( \xeCJK_if_font_select_p:n {bf} )
1486     }
1487     { \xeCJK_add_fake_bold:n {#1} }
1488 \xeCJK_add_font_if_new:nn {#1} *
1489 }
(End definition for \xeCJK_set_fake:n.)

```

\xeCJK_map_it_sl:n

```

1490 \cs_new_nopar:Nn \xeCJK_map_it_sl:n
1491 {
1492     \xeCJK_if_map_font_select:nT {#1}
1493     {
1494         \xeCJK_add_font:nx {#1} { \xeCJK_get_map_font:n {#1} }
1495         \xeCJK_if_font_select:nF {bf#1}
1496         { \xeCJK_add_font_if_new:nx {bf#1} { \xeCJK_get_map_font:n {#1} } }
1497     }
1498 }
(End definition for \xeCJK_map_it_sl:n.)

```

\xeCJK_get_map_font:n

```

1499 \cs_new_nopar:Nn \xeCJK_get_map_font:n
1500 { \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 1501 \prg_new_conditional:Nnn \xeCJK_if_map_font_select:n { p, T, F, TF }
1502 {
1503     \tl_if_blank:xTF { \xeCJK_get_map_font:n {#1} }
1504     \prg_return_false: \prg_return_true:
1505 }
(End definition for \xeCJK_if_map_font_select:n.)

```

\xeCJK_pass_features:

```

1506 \cs_new_nopar:Nn \xeCJK_pass_features:
1507 {
1508     \prop_map_inline:Nn \g_xeCJK_features_id_prop
1509     {
1510         \clist_if_empty:cF { l_xeCJK_fontfeat_###_clist }
1511         {
1512             \clist_put_right:Nx \l_xeCJK_pass_features_clist
1513             { ##2Features = { \clist_use:c { l_xeCJK_fontfeat_###_clist } } }

```

```

1514     }
1515   }
1516   \prop_map_inline:Nn \l_xeCJK_add_font_prop
1517     { \clist_put_right:Nx \l_xeCJK_pass_features_clist { ##1 = { ##2 } } }
1518   \tl_if_blank:xF \l_xeCJK_monoscale_tl
1519     { \clist_put_right:Nx \l_xeCJK_pass_features_clist \l_xeCJK_monoscale_tl }
1520   }
  (End definition for \xeCJK_pass_features:.)

```

```

\g_xeCJK_family_prop
\g_xeCJK_family_name_prop 1521 \prop_new:N \g_xeCJK_family_prop
\g_xeCJK_family_options_prop 1522 \prop_new:N \g_xeCJK_family_name_prop
1523 \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:
1524 \cs_new_nopar:Nn \xeCJK_save_family_info:
1525   {
1526     \group_begin:
1527     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1528       {
1529         \clist_remove_all:Nn \l_xeCJK_fontoptions_clist {##1}
1530         \clist_remove_all:Nn \l_xeCJK_fontoptions_clist { ##1 = }
1531       }
1532     \prop_gput:NVV \g_xeCJK_family_prop
1533       \l_xeCJK_family_tl \l_xeCJK_fontname_tl
1534     \prop_gput:NVV \g_xeCJK_family_options_prop
1535       \l_xeCJK_family_tl \l_xeCJK_fontoptions_clist
1536     \group_end:
1537   }
  (End definition for \xeCJK_save_family_info:.)

```

```

\xeCJK_add_sub_block:
1538 \cs_new_nopar:Nn \xeCJK_add_sub_block:
1539   {
1540     \clist_remove_duplicates:N \l_xeCJK_sub_key_clist
1541     \clist_map_inline:Nn \l_xeCJK_sub_key_clist
1542       {
1543         \tl_set:Nx \l_tmpa_tl { \l_xeCJK_family_tl/##1 }
1544         \bool_if:cT { l_xeCJK_copy_##1_bool }
1545         {
1546           \tl_map_inline:nn { \g_xeCJK_family_prop \g_xeCJK_family_options_prop }
1547             {
1548               \prop_get:NVNT #####1 \l_xeCJK_family_tl \l_tmpb_tl
1549               { \prop_gput:NVV #####1 \l_tmpa_tl \l_tmpb_tl }
1550             }
1551           \prop_get:NVNT \g_xeCJK_family_options_prop \l_tmpa_tl \l_tmpa_clist
1552           {
1553             \clist_remove_all:Nn \l_tmpa_clist { ##1 = * }
1554             \prop_gput:NVV \g_xeCJK_family_options_prop \l_tmpa_tl \l_tmpa_clist
1555           }
1556           \cs_gset_protected_nopar:cpx { xeCJK/family/\l_tmpa_tl }

```

```

1557         {
1558             \exp_not:N \xeCJK_family_if_exist:nT { \l_xeCJK_family_tl }
1559             {
1560                 \prop_get:NnNT \exp_not:N \g_xeCJK_family_name_prop
1561                 { \l_xeCJK_family_tl } \exp_not:N \l_tmpa_tl
1562                 {
1563                     \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1564                     { \l_tmpa_tl } \exp_not:N \l_tmpa_tl
1565                     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_tmpa_tl }
1566                     \exp_not:n
1567                     { {
1568                         \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1569                         \exp_not:N \fontfamily { \l_tmpa_tl }
1570                         \exp_not:N \selectfont
1571                     } }
1572                 }
1573             }
1574         }
1575     }
1576     \bool_if:cT { l_xeCJK_add_##1_bool }
1577     {
1578         \xeCJK_set_family:ncc \l_tmpa_tl
1579         { l_xeCJK_##1_fontfeat_clist } { l_xeCJK_##1_fontname_tl }
1580     }
1581 }
1582 }

```

(End definition for \xeCJK_add_sub_block:.)

\xeCJK_copy_family:nn

```

1583 \cs_new_nopar:Nn \xeCJK_copy_family:nn
1584 {
1585     \xeCJK_family_if_exist:nT {#2}
1586     {
1587         \tl_map_inline:nn
1588         { \g_xeCJK_family_prop \g_xeCJK_family_options_prop \g_xeCJK_family_name_prop }
1589         { \prop_get:NnNT ##1 {#2} \l_tmpa_tl { \prop_gput:NnV ##1 {#1} \l_tmpa_tl } }
1590         \cs_gset_eq:cc { xeCJK/family/#1 } { xeCJK/family/#2 }
1591     }
1592 }

```

\cs_generate_variant:Nn \xeCJK_copy_family:nn { xx }

(End definition for \xeCJK_copy_family:nn.)

5.8 字体切换

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

```

\l_xeCJK_select_font: 1594 \tl_set:Nn \l_xeCJK_font_coors_tl { xeCJK/\xeCJK@family/\f@series/\f@shape/\f@size }
\l_xeCJK_switch_font:nn 1595 \cs_new_nopar:Nn \xeCJK_select_font:
1596 {
1597     \cs_if_exist_use:cF \l_xeCJK_font_coors_tl
1598     {
1599         \tl_set:Nx \l_xeCJK_current_coors_tl \l_xeCJK_font_coors_tl
1600         \cs_if_exist_use:cT { xeCJK/family/\xeCJK@family }

```

```

1601      { \exp_last_unbraced:NNV \cs_gset_eq:cN \l_xeCJK_current_coor_tl \font@name }
1602    }
1603  }
1604  \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
1605  \cs_new_nopar:Nn \xeCJK_switch_font:nn
1606  {
1607    \str_if_eq:xxF {#1} {#2}
1608    {
1609      \xeCJK_info:nxx { CJK-block } {#1} {#2}
1610      \tl_remove_all:Nn \xeCJK@family { /#1 }
1611      \str_if_eq:xxF {#2} { CJK }
1612      {
1613        \tl_set:Nx \l_tmpa_tl \xeCJK@family
1614        \tl_put_right:Nx \xeCJK@family { /#2 }
1615        \xeCJK_family_if_exist:nF \xeCJK@family
1616        {
1617          \xeCJK_family_if_exist:nTF { \CJKfamilydefault/#2 }
1618            { \xeCJK_copy_family:xx \xeCJK@family { \CJKfamilydefault/#2 } }
1619            { \xeCJK_copy_family:xx \xeCJK@family \l_tmpa_tl }
1620        }
1621      }
1622    }
1623    \xeCJK_select_font:
1624  }
1625  \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

```

1626  \prg_new_conditional:Nnn \xeCJK_family_if_exist:n { T, F, TF }
1627  {
1628    \cs_if_free:cTF { xeCJK/family/#1 } \prg_return_false:
1629    {
1630      \prop_if_in:NxF
1631      \g_xeCJK_family_name_prop {#1} { \use:c { xeCJK/family/#1 } }
1632      \prg_return_true:
1633    }
1634  }

```

(End definition for \xeCJK_family_if_exist:nTF.)

\CJKfamily 用于切换 CJK 字体族。

```

1635  \NewDocumentCommand \CJKfamily { m }
1636  {
1637    \xeCJK_family_if_exist:nTF {#1}
1638      { \tl_set:Nx \xeCJK@family {#1} }
1639      {
1640        \cs_if_free:cT { xeCJK_warned_aux_#1 }
1641        {
1642          \xeCJK_warning:nx { CJKfamily-Unknown } {#1}
1643          \tl_new:c { xeCJK_warned_aux_#1 }
1644        }
1645      }
1646  }

```

```

1647 \xeCJK_msg_new:nn { CJKfamily-Unknown }
1648 {
1649   Unknown~CJK~family~'#1'~is~ignored.\\\
1650   Try~to~use~
1651   \prg_case_str:xxn {#1}
1652   {
1653     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1654     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1655     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1656   }
1657   { \token_to_str:N \setCJKfamilyfont {#1} }
1658   [...]{}~to~define~it.
1659 }

```

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

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

```

\newCJKfontfamily 1660 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
\CJKfontspec 1661 { \xeCJK_set_family:nnn {#1} {#2} {#3} }
1662 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1663 {
1664   \IfNoValueTF {#1}
1665   { \tl_set:Nx \l_xeCJK_family_tl { \cs_to_str:N #2 } }
1666   { \tl_set:Nx \l_xeCJK_family_tl {#1} }
1667   \xeCJK_set_family:nnn \l_xeCJK_family_tl {#3} {#4}
1668   \cs_new_protected_nopar:Npx #2
1669   { \exp_not:N \CJKfamily { \l_xeCJK_family_tl } }
1670 }
1671 \int_new:N \g_xeCJK_fontspec_int
1672 \int_gdecr:N \g_xeCJK_fontspec_int
1673 \NewDocumentCommand \CJKfontspec { O{} m }
1674 {
1675   \tl_set:Nx \l_xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1676   \cs_if_free:cT \l_xeCJK_CJKfontspec_id_tl
1677   {
1678     \int_gincr:N \g_xeCJK_fontspec_int
1679     \tl_gset:Nx \g_xeCJK_family_spec_tl
1680     { CJKfontspec ( \int_use:N \g_xeCJK_fontspec_int ) }
1681     \xeCJK_set_family:nnn \g_xeCJK_family_spec_tl {#1} {#2}
1682     \tl_gset:cx \l_xeCJK_CJKfontspec_id_tl \g_xeCJK_family_spec_tl
1683   }
1684   \exp_args:Nv \CJKfamily \l_xeCJK_CJKfontspec_id_tl
1685   \tex_ignorespaces:D
1686 }

```

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

\defaultCJKfontfeatures 分别用于设置 CJK 字体的默认属性和增加当前 CJK 字体的属性。

```

\addCJKfontfeatures 1687 \clist_new:N \g_xeCJK_default_features_clist
1688 \NewDocumentCommand \defaultCJKfontfeatures { m }
1689 { \clist_gset:Nn \g_xeCJK_default_features_clist {#1} }
1690 \@onlypreamble \defaultCJKfontfeatures
1691 \NewDocumentCommand \addCJKfontfeatures { m }

```

```

1692 {
1693   \prop_if_in:NVTF \g_xeCJK_family_prop \xeCJK@family
1694   {
1695     \group_begin:
1696     \clist_set:Nx \l_tmpa_clist {#1}
1697     \prop_get:NVN \g_xeCJK_family_prop \xeCJK@family \l_tmpa_tl
1698     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1699     {
1700       \clist_if_in:NnT \l_tmpa_clist {##1}
1701       {
1702         \clist_remove_all:Nn \l_tmpa_clist {##1}
1703         \prop_get:NxNF \g_xeCJK_family_prop
1704         { \xeCJK@family/##1 } \l_tmpb_tl
1705         {
1706           \prop_get:NxNF \g_xeCJK_family_prop
1707           { \CJKfamilydefault/##1 } \l_tmpb_tl
1708           { \tl_set:Nx \l_tmpb_tl \l_tmpa_tl }
1709         }
1710         \clist_set:Nx \l_tmpa_clist
1711         { ##1 = { [ \l_tmpa_clist ] { \l_tmpb_tl } } }
1712         \clist_map_break:
1713       }
1714     }
1715     \prop_get:NVNF \g_xeCJK_family_options_prop \xeCJK@family
1716     \l_tmpb_clist { \clist_clear:N \l_tmpb_clist }
1717     \clist_clear:N \l_tmpc_clist
1718     \clist_map_inline:Nn \l_tmpb_clist
1719     {
1720       \clist_if_in:NnF \l_tmpa_clist {##1}
1721       { \clist_put_right:Nn \l_tmpc_clist {##1} }
1722     }
1723     \clist_put_left:NV \l_tmpa_clist \l_tmpc_clist
1724     \use:x
1725     { \group_end: \exp_not:N \CJKfontspec [ \l_tmpa_clist ] { \l_tmpa_tl } }
1726   }
1727   { \xeCJK_warning:n { addCJKfontfeature-ignored } }
1728   \tex_ignorespaces:D
1729 }
1730 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
1731 \xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1732 {
1733   \tl_to_str:N \addCJKfontfeatures (s)~ignored.\\\
1734   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1735 }

```

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

\setCJKmainfont 设置文档的 CJK 普通字体、无衬线字体。

```

\setCJKsansfont 1736 \NewDocumentCommand \setCJKmainfont { O{ m }
1737   { \xeCJK_set_family:nnn \CJKrmdefault {#1} {#2} }
1738 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1739 \NewDocumentCommand \setCJKsansfont { O{ m }
1740   { \xeCJK_set_family:nnn \CJKsfdefault {#1} {#2} }

```

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

```
\CJKrmdefault
\CJKsfdefault 1741 \cs_if_free:NT \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
\CJKttdefault 1742 \cs_if_free:NT \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 1743 \cs_if_free:NT \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1744 \cs_if_free:NT \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1745 \tl_new:c { xeCJK/family/\CJKfamilydefault }
```

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

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

```
1746 \AtEndOfPackage
1747 {
1748   \tex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1749   \tex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }
1750   \tex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1751   \tex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1752   \cs_set_eq:NN \reset@font \normalfont
1753 }
```

\setCJKmathfont 设置 CJK 数学字体。

```
1754 \NewDocumentCommand \setCJKmathfont { O{} m }
1755 { \xeCJK_set_family:nnn \c_xeCJK_math_tl {#1} {#2} }
```

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

CJKmath 数学公式内的 CJK 字体。

```
1756 \tl_const:Nn \c_xeCJK_math_tl { CJKmath }
1757 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g_xeCJK_math_bool }
```

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

\xeCJKsetmathcode

```
1758 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1759 {
1760   \xeCJK_check_num_range:nnNN {#1} {#2} \l_tmpa_int \l_tmpb_int
1761   \int_set:Nn \l_tmpc_int { \xeCJK_math_type:n {#3} }
1762   \loop
1763     \XeTeXmathcode \l_tmpa_int = \l_tmpc_int \use:c { sym #4 } \l_tmpa_int
1764     \if_int_compare:w \l_tmpa_int < \l_tmpb_int
1765       \int_incr:N \l_tmpa_int
1766   \repeat
1767 }
1768 \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。

```
1769 \tl_put_left:Nn \document
1770 {
1771   \group_end:
1772   \prop_if_empty:NTF \g_xeCJK_family_prop
1773     { \xeCJK_warning:n { no-CJKfamily } }
1774     {
1775       \bool_if:NT \g_xeCJK_math_bool
```

```

1776     {
1777         \xeCJK_family_if_exist:nTF \c_xeCJK_math_tl
1778         { \tl_set:Nx \l_tmpa_tl \c_xeCJK_math_tl }
1779         {
1780             \xeCJK_family_if_exist:nTF \CJKfamilydefault
1781             { \tl_set:Nx \l_tmpa_tl \CJKfamilydefault }
1782             { \tl_set_eq:NN \l_tmpa_tl \q_no_value }
1783         }
1784         \prop_get:NVNT \g_xeCJK_family_name_prop \l_tmpa_tl \l_xeCJK_math_family_tl
1785         {
1786             \DeclareSymbolFont \c_xeCJK_math_tl \g_fontspec_encoding_tl
1787             \l_xeCJK_math_family_tl \mddefault \shapedefault
1788             \SetSymbolFont \c_xeCJK_math_tl { bold } \g_fontspec_encoding_tl
1789             \l_xeCJK_math_family_tl \bfdefault \shapedefault
1790             \clist_map_inline:nn { CJK, FullLeft, FullRight }
1791             {
1792                 \clist_map_inline:cn { g_xeCJK_#1_range_clist }
1793                 {
1794                     \xeCJK_setcharclass_aux:Nn \xeCJKsetmathcode {##1}
1795                     \mathalpha \c_xeCJK_math_tl
1796                 }
1797             }
1798         }
1799     }
1800     \normalfont
1801 }
1802 \group_begin:
1803 }
1804 \xeCJK_msg_new:nn { no-CJKfamily }
1805 {
1806     It~seems~that~you~have~not~declare~a~CJKfamily.\\
1807     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
1808     \token_to_str:N~\xeCJKmainfont[...]{...}\\
1809     in~the~preamble~to~declare~the~main~CJKfamily.\\
1810 }

```

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。

```

1811 \@onlypreamble \setCJKmainfont
1812 \@onlypreamble \setCJKmathfont
1813 \@onlypreamble \setCJKsansfont
1814 \@onlypreamble \setCJKmonofont
1815 \@onlypreamble \setCJKromanfont

```

5.9 处理等宽字体和抄录环境

```

\g_xeCJK_monoscale_fp
\g_xeCJK_exspace_dim
1816 \fp_new:N \g_xeCJK_monoscale_fp
1817 \fp_gset_eq:NN \g_xeCJK_monoscale_fp \c_one_fp
1818 \dim_new:N \g_xeCJK_exspace_dim
(End definition for \g_xeCJK_monoscale_fp and \g_xeCJK_exspace_dim.)

\setCJKmonoscale

```

```

1819 \NewDocumentCommand \setCJKmonoscale { }
1820 {
1821   \xeCJK_set_monoscale:
1822   \addCJKfontfeatures { Scale = \fp_use:N \g_xeCJK_monoscale_fp }
1823 }
1824 \cs_new_nopar:Nn \xeCJK_set_monoscale:
1825 {
1826   \dim_gzero:N \g_xeCJK_exspace_dim
1827   \group_begin:
1828     \fontfamily \ttdefault \selectfont
1829     \fp_gset_from_dim:Nn \g_xeCJK_monoscale_fp
1830     { \c_two \tex_fontdimen:D \c_two \tex_font:D }
1831     \fp_gdiv:Nn \g_xeCJK_monoscale_fp \f@size
1832   \group_end:
1833 }

```

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

`\setCJKmonoexspace`

```

1834 \NewDocumentCommand \setCJKmonoexspace { } { \xeCJK_set_monoexspace: }
1835 \cs_new_nopar:Nn \xeCJK_set_monoexspace:
1836 {
1837   \fp_gset_eq:NN \g_xeCJK_monoscale_fp \c_one_fp
1838   \group_begin:
1839     \fontfamily \ttdefault \selectfont
1840     \dim_gset:Nn \g_xeCJK_exspace_dim
1841     { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size \p@ }
1842   \group_end:
1843 }

```

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

`\xeCJK_fixed_ecglue:`

```

\xeCJK_fixed_cjkglue: 1844 \cs_new_nopar:Nn \xeCJK_fixed_ecglue: { \skip_horizontal:n { .5\g_xeCJK_exspace_dim } }
\xeCJK_flexible_ecglue: 1845 \cs_new_nopar:Nn \xeCJK_fixed_cjkglue: { \skip_horizontal:N \g_xeCJK_exspace_dim }
\xeCJK_flexible_ecglue: 1846 \AtEndOfPackage
\xeCJK_flexible_cjkglue: 1847 {
1848   \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1849   \cs_new_eq:NN \xeCJK_flexible_ecglue: \CJKecglue
1850   \cs_new_eq:NN \xeCJK_flexible_cjkglue: \CJKglue
1851 }

```

(End definition for \xeCJK_fixed_ecglue: and others.)

`\CJKfixedspacing`

```

1852 \NewDocumentCommand \CJKfixedspacing { }
1853 {
1854   \tl_if_eq:NNF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1855   {
1856     \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1857     \punctstyle { plain }
1858   }
1859   \cs_if_eq:NNF \CJKecglue \xeCJK_fixed_ecglue:
1860   {
1861     \cs_set_eq:NN \xeCJK_flexible_ecglue: \CJKecglue

```

```

1862     \cs_set_eq:NN \CJKecglue \xeCJK_fixed_ecglue:
1863   }
1864   \cs_if_eq:NNF \CJKglue \xeCJK_fixed_cjkg glue:
1865   {
1866     \cs_set_eq:NN \xeCJK_flexible_cjkg glue: \CJKglue
1867     \cs_set_eq:NN \CJKglue \xeCJK_fixed_cjkg glue:
1868   }
1869 }
1870 \AtBeginDocument { \tl_put_right:Nn \verbatim@font \CJKfixedspacing }
(End definition for \CJKfixedspacing. This function is documented on page 6.)

```

\CJKflexiblespacing

```

1871 \NewDocumentCommand \CJKflexiblespacing { m }
1872 {
1873   \cs_set_eq:NN \l_xeCJK_punctstyle_tl \l_xeCJK_flexible_punctstyle_tl
1874   \cs_set_eq:NN \CJKecglue \xeCJK_flexible_ecglue:
1875   \cs_set_eq:NN \CJKglue \xeCJK_flexible_cjkg glue:
1876 }
(End definition for \CJKflexiblespacing. This function is documented on page 6.)

```

\setCJKmonofont 设置文档的 CJK 等宽字体族。

```

1877 \NewDocumentCommand \setCJKmonofont { s t+ O{ } m }
1878 {
1879   \IfBooleanTF {#1}
1880   { \xeCJK_set_family:nnn \CJKttdefault { Mono = Scale, #3 } {#4} }
1881   {
1882     \IfBooleanTF {#2}
1883     { \xeCJK_set_family:nnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
1884     { \xeCJK_set_family:nnn \CJKttdefault {#3} {#4} }
1885   }
1886 }
(End definition for \setCJKmonofont. This function is documented on page 5.)

```

5.10 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。

```

1887 \bool_new:N \g_xeCJK_indent_bool
1888 \keys_define:nn { xeCJK / options }
1889 {
1890   CJKnumber .bool_set:N = \g_xeCJK_number_bool ,
1891   indentfirst .bool_set:N = \g_xeCJK_indent_bool ,
1892   normalindentfirst .meta:n = { indentfirst = false } ,
1893   quiet .code:n =
1894   {
1895     \msg_redirect_module:nnn { xeCJK } { warning } { info }
1896     \msg_redirect_module:nnn { xeCJK } { info } { none }
1897     \PassOptionsToPackage { quiet } { fontspec }
1898   },
1899   silent .code:n =
1900   {
1901     \msg_redirect_module:nnn { xeCJK } { warning } { none }

```

```

1902      \msg_redirect_module:nnn { xeCJK } { info } { none }
1903      \PassOptionsToPackage { silent } { fontspec }
1904    },
1905    unknown .code:n =
1906    {
1907      \@ifpackageloaded { fontspec }
1908      { \xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
1909      { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
1910    },
1911  }
1912  \xeCJK_msg_new:nn { key-unknown }
1913  {
1914    Sorry,~but~\l_keys_module_tl\c_space_tl does~not~have~a~key~called~'#1'.\\
1915    The~key~'#1'~is~being~ignored.
1916  }

```

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

5.11 xeCJK 初始化设置

xeCJK 宏包的初始化设置。

```

1917 \keys_set:nn { xeCJK / options }
1918 {
1919   CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
1920   CJKecglue = \c_space_token ,
1921   CheckSingleglue = \CJKglue ,
1922   xCJKecglue = false ,
1923   CheckSingle = false ,
1924   CJKspace = false ,
1925   CJKmath = false ,
1926   xeCJKactive = true ,
1927   indentfirst = true ,
1928   EmboldenFactor = 4 ,
1929   SlantFactor = .167 ,
1930   PunctStyle = quanjiao ,
1931   KaiMingPunct = { . 。 ? ! } ,
1932   LongPunct = { — … † — } ,
1933   MiddlePunct = { • • • } ,
1934   AllowBreakBetweenPuncts = false ,
1935 }

```

\CJKsymbol

```

\CJKpunctsymbol 1936 \cs_new_nopar:Npn \CJKsymbol #1 {#1}
1937 \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。

```

1938 \ProcessKeysOptions { xeCJK / options }
1939 \RequirePackage { fontspec } [ 2011/09/13 ]
1940 \bool_if:NT \g_xeCJK_indent_bool
1941 {
1942   \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
1943   \@afterindenttrue
1944 }

```

`\xeCJKsetup` 在导言区或文档中设置 `\xeCJK` 的接口。

```
1945 \NewDocumentCommand \xeCJKsetup { m }
1946 {
1947   \keys_set:nn { \xeCJK / options } {#1}
1948   \tex_ignorespaces:D
1949 }
(End definition for \xeCJKsetup. This function is documented on page 3.)
```

`\xeCJKsetemboldenfactor`

```
\xeCJKsetslantfactor 1950 \NewDocumentCommand \xeCJKsetEmboldenFactor { m }
1951 { \xeCJKsetup { EmboldenFactor = {#1} } }
1952 \NewDocumentCommand \xeCJKsetSlantFactor { m }
1953 { \xeCJKsetup { SlantFactor = {#1} } }
(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor.)
```

`\punctstyle`

```
\xeCJKplainchr 1954 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
1955 \cs_new_nopar:Npn \xeCJKplainchr { \punctstyle { plain } }
(End definition for \punctstyle and \xeCJKplainchr.)
```

`\CJKsetecglue`

```
1956 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
1957 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
(End definition for \CJKsetecglue.)
```

5.12 兼容性修补

`\/` 修复倾斜校正。

```
1958 \cs_new_eq:NN \xeCJK_italiccorr_aux \/
1959 \bool_new:N \g_xeCJK_italiccorr_bool
1960 \cs_set_protected_nopar:Npn \/
1961 {
1962   \scan_stop:
1963   \int_compare:nNt \tex_lastkern:D = \c_three
1964   {
1965     \tex_unkern:D \tex_unkern:D
1966     \bool_gset_true:N \g_xeCJK_italiccorr_bool
1967   }
1968   \xeCJK_italiccorr_aux
1969 }
1970 \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 1971 \cs_new_nopar:Nn \xeCJK_compatibility_hook:
\xeCJK_Others_class 1972 {
1973   \int_compare:nNf
1974   { \c_three + \seq_length:N \g_xeCJK_class_seq } = \xe@alloc@intercharclass
1975   {
1976     \int_set_eq:NN \l_tmpa_int \c_three
1977     \int_until_do:nNnn \l_tmpa_int = \xe@alloc@intercharclass
```

```

1978     {
1979         \int_incr:N \l_tmpa_int
1980         \seq_if_in:NVF \g_xeCJK_class_seq \l_tmpa_int
1981         { \xeCJK_set_others_toks:n \l_tmpa_int }
1982     }
1983 }
1984 }
1985 \cs_new_nopar:Nn \xeCJK_set_others_toks:n
1986 {
1987     \int_set:Nn \xeCJK_Others_class {#1}
1988     \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
1989     {
1990         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
1991         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
1992         \exp_args:Nnnx \xeCJK_app_inter_class_toks:nnn {##1} { Others }
1993         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
1994         \exp_args:Nnnx \xeCJK_pre_inter_class_toks:nnn { Others } {##1}
1995         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
1996         \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
1997         {
1998             \xeCJK_copy_inter_class_toks:nnnn
1999             { Others } { Boundary } { Default } { Boundary }
2000         }
2001         \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2002         {
2003             \xeCJK_copy_inter_class_toks:nnnn
2004             { Boundary } { Others } { Boundary } { Default }
2005         }
2006     }
2007 }
2008 \int_new:N \xeCJK_Others_class
2009 \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。

```

2010 \cs_new_nopar:Nn \xeCJK_patch:Nnn
2011 { \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。
2012 \AtBeginDocument
2013 {
2014     \tl_map_inline:nn
2015     {
2016         \textellipsis \textemdash \textperiodcentered \textcentereddot
2017         \textquotelleft \textquoteright \textquotedblleft \textquotedblright
2018     }
2019     { \xeCJK_patch:Nnn #1 { \group_begin: \makexeCJKinactive } { \group_end: } }
2020     \xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2021     \cs_set_eq:NN \xeCJK_aux_r \r
2022     \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \xeCJK_aux_r {#1} } }
2023     \@ifpackageloaded { pifont }

```

```

2024 {
2025   \RenewDocumentCommand \Pifont { m }
2026   {
2027     \fontencoding {U} \fontfamily {#1} \fontseries {m} \fontshape {n}
2028     \selectfont \makexeCJKinactive
2029   }
2030 } { }
2031 }

```

禁止在 xeCJK 宏包后再载入 CJK 宏包。

```

2032 \tl_set:cn { ver@CJK.sty } { 2050/01/01 }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

2033 \cs_if_free:NT \CJK@ifundefined
2034 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2035 \NewDocumentCommand \xeCJKcaption { o m }
2036 {
2037   \IfValueT {#1} { \XeTeXdefaultencoding "#1" }
2038   \cs_set_nopar:Nx \xeCJK_reset_at_catcode:
2039   {
2040     \exp_not:n { \char_set_catcode:nn { '\@ } }
2041     { \char_value_catcode:n { '\@ } }
2042   }
2043   \char_set_catcode_letter:N \@
2044   \file_input:n { #2.cpx }
2045   \xeCJK_reset_at_catcode:
2046   \XeTeXdefaultencoding "UTF-8"
2047 }

```

(End definition for \xeCJKcaption.)

\xeCJK_ULprepunctchar:n

\xeCJK_ULpostpunctchar:n

\xeCJK_ULroutines:

```

2048 \cs_new_protected_nopar:Nn \xeCJK_ULprepunctchar:n
2049 {
2050   { \makexeCJKinactive \CJKpunctsymbol {#1} \nobreak }
2051   \tex_ignorespaces:D
2052 }
2053 \cs_new_protected_nopar:Nn \xeCJK_ULpostpunctchar:n
2054 {
2055   { \makexeCJKinactive \CJKpunctsymbol {#1} }
2056   \xeCJK_ignorespaces:
2057 }
2058 \cs_new_protected_nopar:Nn \xeCJK_ULroutines:
2059 {
2060   \xeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKeclue \CJKsymbol }
2061   \xeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2062   \xeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKeclue \CJKsymbol }
2063   \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }
2064   \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2065   {
2066     \xeCJK_inter_class_toks:nnn {##1} { FullLeft } { \xeCJK_ULprepunctchar:n }
2067     \xeCJK_inter_class_toks:nnn {##1} { FullRight } { \xeCJK_ULpostpunctchar:n }
2068   }
2069   \xeCJK_UL_subclass_patch_tl

```

```

2070 }
(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。
2071 \AtBeginDocument
2072 {
2073   \cs_if_exist:NT \UL@hook
2074   {
2075     \addto@hook \UL@hook
2076     {
2077       \cs_set_eq:NN \xeCJK_UL_CJKsymbol \CJKsymbol
2078       \cs_set_eq:NN \xeCJK_UL_CJKpunctsymbol \CJKpunctsymbol
2079       \cs_set_nopar:Npn \CJKsymbol #1
2080       {
2081         { \xeCJK_select_font: \xeCJK_UL_CJKsymbol {#1} }
2082         \xeCJK_CJK_kern: \xeCJK_ignorespaces:
2083       }
2084       \cs_set_nopar:Npn \CJKpunctsymbol #1
2085       { { \xeCJK_select_font: \xeCJK_UL_CJKpunctsymbol {#1} } }
2086       \xeCJK_ULroutines:
2087     }
2088   }
2089   \cs_if_exist:NT \XeTeX@CJKfntef@hook
2090   {
2091     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2092     { \xeCJK_select_font: \makeXeCJKinactive }
2093   }
2094 }
使用 CJKnumb 宏包。
2095 \bool_if:NT \g_xeCJK_number_bool
2096 {
2097   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2098   \cs_set_nopar:Npn \CJKaddEncHook #1#2 { \cs_set_nopar:cpn { xeCJK_enc_#1 } {#2} }
2099   \cs_set_nopar:Npn \Unicode #1#2
2100   { \tex_char:D \int_eval:n { #1 * \c_two_hundred_fifty_six + #2 } }
2101   \RequirePackage { CJKnumb }
2102   \use:c { xeCJK_enc_ \CJK@UnicodeEnc }
2103   \tl_set:Nn \CJK@tenthousand { 万 }
2104   \tl_set:Nn \CJK@hundredmillion { 亿 }
2105 }
2106 </package>

```

6 例子

6.1 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]{simsum.ttc}
7   \setCJKmonofont[ItalicFont=simkai.ttf]{simsum.ttc}
8   \begin{document}
9   \centering
10  \begin{tabular}{lllll}
11  \hline
12  {\bf rm} & md & up & \verb|\rmfamily\mdseries\upshape| & &
13             & & & {\rmfamily\mdseries\upshape English 中文字体} & \\\
14            & md & it & \verb|\rmfamily\mdseries\itshape| & &
15             & & & {\rmfamily\mdseries\itshape English 中文字体} & \\\
16            & md & sl & \verb|\rmfamily\mdseries\slshape| & &
17             & & & {\rmfamily\mdseries\slshape English 中文字体} & \\\cline{2-5}
18            & bf & up & \verb|\rmfamily\bfseries\upshape| & &
19             & & & {\rmfamily\bfseries\upshape English 中文字体} & \\\
20            & bf & it & \verb|\rmfamily\bfseries\itshape| & &
21             & & & {\rmfamily\bfseries\itshape English 中文字体} & \\\
22            & bf & sl & \verb|\rmfamily\bfseries\slshape| & &
23             & & & {\rmfamily\bfseries\slshape English 中文字体} & \\\hline
24  {\bf sf} & md & up & \verb|\sffamily\mdseries\upshape| & &
25             & & & {\sffamily\mdseries\upshape English 中文字体} & \\\
26            & md & it & \verb|\sffamily\mdseries\itshape| & &
27             & & & {\sffamily\mdseries\itshape English 中文字体} & \\\
28            & md & sl & \verb|\sffamily\mdseries\slshape| & &
29             & & & {\sffamily\mdseries\slshape English 中文字体} & \\\cline{2-5}
30            & bf & up & \verb|\sffamily\bfseries\upshape| & &
31             & & & {\sffamily\bfseries\upshape English 中文字体} & \\\
32            & bf & it & \verb|\sffamily\bfseries\itshape| & &
33             & & & {\sffamily\bfseries\itshape English 中文字体} & \\\
34            & bf & sl & \verb|\sffamily\bfseries\slshape| & &
35             & & & {\sffamily\bfseries\slshape English 中文字体} & \\\hline
36  {\bf tt} & md & up & \verb|\ttfamily\mdseries\upshape| & &
37             & & & {\ttfamily\mdseries\upshape English 中文字体} & \\\
38            & md & it & \verb|\ttfamily\mdseries\itshape| & &
39             & & & {\ttfamily\mdseries\itshape English 中文字体} & \\\
40            & md & sl & \verb|\ttfamily\mdseries\slshape| & &
41             & & & {\ttfamily\mdseries\slshape English 中文字体} & \\\cline{2-5}
42            & bf & up & \verb|\ttfamily\bfseries\upshape| & &
43             & & & {\ttfamily\bfseries\upshape English 中文字体} & \\\
44            & bf & it & \verb|\ttfamily\bfseries\itshape| & &
45             & & & {\ttfamily\bfseries\itshape English 中文字体} & \\\
46            & bf & sl & \verb|\ttfamily\bfseries\slshape| & &
47             & & & {\ttfamily\bfseries\slshape English 中文字体} & \\\hline
48  \end{tabular}
49  \end{document}
50  \

```

6.2 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 漢字源流考
62
63 \textbf{漢字源流考}
64
65 \textsl{漢字源流考}
66
67 \CJKunderwave{漢字源流考}
68 \begin{table}[ht]
69 \caption{生僻字测试}
70 \medskip\centering
71 \begin{tabular}{*4{<{\ttfamily U+}l}}
72 𠂇 & 3400 & 𠂈 & 3401 & 𠂉 & 3402 & 𠂊 & 3403 & \\\
73 𠂋 & 3404 & 𠂌 & 3405 & 𠂍 & 3406 & 𠂎 & 3407 & \\\
74 𠂏 & 3408 & 𠂐 & 3409 & 𠂑 & 340A & 𠂒 & 340B & \\\
75 𠂓 & 340C & 𠂔 & 340D & 𠂕 & 340E & 𠂖 & 341F & \\\
76 𠂗 & 3410 & 𠂘 & 3411 & 𠂙 & 3412 & 𠂚 & 3413 & \\\
77 𠂛 & 3414 & 𠂜 & 3415 & 𠂝 & 3416 & 𠂞 & 3417 & \\\
78 𠂟 & 3418 & 𠂠 & 3419 & 𠂡 & 341A & 𠂢 & 341B & \\\
79 𠂣 & 341C & 𠂤 & 341D & 𠂥 & 341E & 𠂦 & 341F & \\\[1ex]
80 𠂧 & 20000 & 𠂨 & 20001 & 𠂩 & 20002 & 𠂪 & 20003 & \\\
81 𠂫 & 20004 & 𠂬 & 20005 & 𠂭 & 20006 & 𠂮 & 20007 & \\\
82 𠂯 & 20008 & 𠂰 & 20009 & 𠂱 & 2000A & 𠂲 & 2000B & \\\
83 𠂳 & 2000C & 𠂴 & 2000D & 𠂵 & 2000E & 𠂶 & 2000F & \\\
84 𠂷 & 20010 & 𠂸 & 20011 & 𠂹 & 20012 & 𠂺 & 20013 & \\\
85 𠂻 & 20014 & 𠂼 & 20015 & 𠂽 & 20016 & 𠂾 & 20017 & \\\
86 𠂿 & 20018 & 𠃀 & 20019 & 𠃁 & 2001A & 𠃂 & 2001B & \\\
87 𠃃 & 2001C & 𠃄 & 2001D & 𠃅 & 2001E & 𠃆 & 2001F & \\\
88 \end{tabular}
89 \end{table}
90 \end{document}
91 </ex-fallback>

```

6.3 example-SubCJKBlock.tex

```

92 <*ex-block>
93 \documentclass{article}
94 \usepackage{xeCJK}
95 \usepackage{array}
96 \xeCJKDeclareSubCJKBlock{Ext-A} { "3400 -> "4DBF }
97 \xeCJKDeclareSubCJKBlock{Ext-B} { "20000 -> "2A6DF }
98 \xeCJKDeclareSubCJKBlock{Kana} { "3040 -> "309F, "30A0 -> "30FF, "31F0 -> "31FF, }
99 \xeCJKDeclareSubCJKBlock{Hangul}{ "1100 -> "11FF, "3130 -> "318F, "A960 -> "A97F, "AC00 -> "D7AF }
100 \setCJKmainfont[Ext-A=SimHei,Ext-B=SimSun-ExtB]{SimSun}
101 \setCJKmainfont[Kana]{Meiryo}
102 \setCJKmainfont[Hangul]{Malgun Gothic}
103 \parindent=2em
104 \begin{document}

```

`\long\def\showtext{%`
 中日韩越统一表意文字（英语：CJKV Unified Ideographs），旧称中日韩统一表意文字（英语：CJK Unified Ideographs），也称统一汉字（英语：Unihan），目的是要把分别来自中文、日文、韩文、越文、壮文中，对于相同起源、本义相同、形状一样或稍异的表意文字（主要为汉字，但也有仿汉字如：方块壮字、日文汉字（かんじ / kanji）、韩文汉字（한자 / 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）。

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

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

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

`\CJKnospace}`
`\showtext`
`\bigskip`
`\xeCJKCancelSubCJKBlock{Kana,Hangul}`
`\showtext`
`\bigskip`
`\xeCJKRestoreSubCJKBlock{Hangul}`
`\showtext`
`\begin{table}[ht]`
`\caption{生僻字测试}`
`\medskip\centering`
`\begin{tabular}{*4{|c>{\ttfamily U+}}l}}`
 𪛗 & 3400 & 𪛘 & 3401 & 𪛚 & 3402 & 𪛜 & 3403 & \\
 𪛞 & 3404 & 𪛠 & 3405 & 𪛢 & 3406 & 𪛤 & 3407 & \\
 𪛦 & 3408 & 𪛨 & 3409 & 𪛪 & 340A & 𪛬 & 340B & \\

```

132 𠂇 & 340C & 𠂈 & 340D & 𠂉 & 340E & 𠂊 & 341F \\
133 𠂋 & 3410 & 𠂌 & 3411 & 𠂍 & 3412 & 𠂎 & 3413 \\
134 𠂏 & 3414 & 𠂐 & 3415 & 𠂑 & 3416 & 𠂒 & 3417 \\
135 𠂓 & 3418 & 𠂔 & 3419 & 𠂕 & 341A & 𠂖 & 341B \\
136 𠂗 & 341C & 𠂘 & 341D & 𠂙 & 341E & 𠂚 & 341F \\[1ex]
137 𠂛 & 20000 & 𠂜 & 20001 & 𠂝 & 20002 & 𠂞 & 20003 \\
138 𠂟 & 20004 & 𠂠 & 20005 & 𠂡 & 20006 & 𠂢 & 20007 \\
139 𠂣 & 20008 & 𠂤 & 20009 & 𠂥 & 2000A & 𠂦 & 2000B \\
140 𠂧 & 2000C & 𠂨 & 2000D & 𠂩 & 2000E & 𠂪 & 2000F \\
141 𠂫 & 20010 & 𠂬 & 20011 & 𠂭 & 20012 & 𠂮 & 20013 \\
142 𠂯 & 20014 & 𠂰 & 20015 & 𠂱 & 20016 & 𠂲 & 20017 \\
143 𠂳 & 20018 & 𠂴 & 20019 & 𠂵 & 2001A & 𠂶 & 2001B \\
144 𠂷 & 2001C & 𠂸 & 2001D & 𠂹 & 2001E & 𠂺 & 2001F \\
145 \end{tabular}
146 \end{table}
147 \end{document}
148 </ex-block>

```