

xeCJK 宏包

ctex.org*

2012/05/21 v3.0.6

目录

1	简介	2
2	基本用法	2
3	用户手册	3
3.1	宏包选项	3
3.2	字体设置与选择	5
3.2.1	X _Y TeX 的字体名查找	8
3.3	CJK 分区字体设置	9
3.4	设置 CJK 字符范围	10
3.5	设置标点宽度和间距	10
4	已知问题和兼容性	11
5	xeCJK 代码实现	11
5.1	字符类别设定	12
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	33
5.6	后备字体	41
5.7	CJK 字体族声明方式	42
5.8	字体切换	51
5.9	处理等宽字体和抄录环境	56
5.10	xeCJK 其它选项	58
5.11	xeCJK 初始化设置	59
5.12	兼容性修补	60
6	例子	63
6.1	xeCJK-example-autofake.tex	63
6.2	xeCJK-example-fallback.tex	64
6.3	xeCJK-example-subCJKblock.tex	65
6.4	xeCJK-example-CJKecglue.tex	67
6.5	xeCJK-example-checksingle.tex	67

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

1 简介

`xeCJK` 是一个 $\text{Xe}_{\text{L}}\text{TeX}$ 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 $\text{Xe}_{\text{L}}\text{TeX}$ 的一些最新特性,需要 $\text{Xe}_{\text{L}}\text{TeX}$ 0.9995.0 [2009/06/29] 以后的版本。`xeCJK` 的依赖 LaTeX 3 项目的 `expl3`, `xparse` 和 `l3keys2e` 宏包,并依赖 `fontspec`¹ 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-kit`² 项目进行维护,目前主要维护者是刘海洋³ 和李清⁴。

2 基本用法

与其他 LaTeX 宏包一样,引入 `xeCJK` 宏包只要在导言区使用

```
\usepackage{xeCJK}
```

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

可以在各种文档类中使用 `xeCJK` 宏包,最简单的示例是:

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

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

上述示例设置了中文字体 `SimSun`(宋体)。运行此示例要求系统安装了设置的字体,源文件用 UTF-8 编码保存,使用 $\text{Xe}_{\text{L}}\text{TeX}$ 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>} <p>设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距,默认值是一个空格。使用这个选项设置的 glue 最好也要用一定的弹性。请注意,这里设置的 glue 只影响 xeCJK 根据需要自动添加的空白,源文件中直接输入的 CJK 文字与西文之间的空格不受影响(直接输出)。有时候 xeCJK 可能不能正确地调整间距,需要手动加空格。</p>
<hr/> xCJKe glue <hr/>	xCJKe glue = {<true false glue>} <p>缺省状态下,xeCJK 不对源文件中直接输入的 CJK 文字与西文之间的空格进行调整,如果需要调整,请使用这个选项。如果使用这个选项,将使用 CJKe glue 替换源文件中直接输入的 CJK 文字与西文之间的空格。</p>
<hr/> CheckSingle <hr/>	CheckSingle = <true false> <p>是否避免单个 CJK 文字单独占一个段落的最后一行。</p>
<hr/> InlineEnv InlineEnv+ InlineEnv- <hr/>	InlineEnv = {env1, env2, env3, ...} <p>在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 L^AT_EX 环境的开始 \begin{...} 和结束 \end{...} 视为断行的地方,如果有某些特殊的 L^AT_EX 环境没有造成断行,可以使用这个选项来声明它,以便 CheckSingle 能正确识别。</p>
<hr/> AutoFallBack <hr/>	AutoFallBack = <true false> <p>当文档中有个别生僻字时,可以使用这个选项,自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。</p>
<hr/> AutoFakeBold ☆ <hr/>	AutoFakeBold = {<true false 数字>} <p>全局设定当没有声明对应的粗体时,是否使用伪粗体;当输入的是数字时,将使用伪粗体,并将使用输入的数字作为伪粗体的默认粗细程度。</p>
<hr/> AutoFakeSlant ☆ <hr/>	AutoFakeSlant = {<true false 数字>} <p>全局设定当没有声明对应的斜体时,是否使用伪斜体;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。</p>
<hr/> EmboldenFactor ☆ <hr/>	EmboldenFactor = {<数字 4>} <p>设置伪粗体的默认粗细程度。</p>
<hr/> SlantFactor ☆ <hr/>	SlantFactor = {<数字 0.167>} <p>设置伪斜体的粗细程度,范围是 -0.999 ~ 0.999。</p>
<hr/> PunctStyle <hr/>	PunctStyle = {<quanjiao banjiao kaiming hangmoban CCT plain>} <p>设置标点处理格式,有效值分别为</p> <p>quanjiao 全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;</p> <p>banjiao 半角式: 所有标点占半个汉字宽度;</p> <p>kaiming 开明式: 句末点号用全角,其他半角;</p> <p>hangmoban 行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</p> <p>CCT CCT 格式;</p> <p>plain 原样(不调整标点间距)。</p>

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

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

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

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

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

AllowBreakBetweenPuncts		AllowBreakBetweenPuncts = {⟨true false⟩}
-------------------------	--	--

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

3.2 字体设置与选择

\setCJKmainfont	★	\setCJKmainfont [⟨font features⟩] {⟨font name⟩}
-----------------	---	---

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

AutoFakeBold		AutoFakeBold = {⟨true false 数字⟩}
AutoFakeSlant		AutoFakeSlant = {⟨true false 数字⟩}

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

\setCJKsansfont	★	\setCJKsansfont [⟨font features⟩] {⟨font name⟩}
-----------------	---	---

设置正文无衬线族的 CJK 字体,影响 `\sffamily` 和 `\textsf` 的字体。

\setCJKmonofont	★	\setCJKmonofont [⟨font features⟩] {⟨font name⟩}
-----------------	---	---

设置正文等宽族的 CJK 字体,影响 `\ttfamily` 和 `\texttt` 的字体。为了有利于等宽字体的代码对齐等情形,xeCJK 在 ⟨font features⟩ 里增加了 `Mono` 这个选项。

<hr/> Mono <hr/>	<p>Mono = $\langle Scale \rangle Exspace$</p> <p>有效值仅为 Scale 或 Exspace。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/> \CJKfixedspacing \CJKflexiblespacing <hr/>	<p>\CJKfixedspacing</p> <p>\CJKflexiblespacing</p> <p>\CJKfixedspacing 命令设置固定字距和不压缩标点格式,符合上面 Mono=Exspace 选项的意义。\\CJKflexiblespacing 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 \\CJKfixedspacing 的效果。</p>
<hr/> \setCJKmonoscale \setCJKmonoexspace <hr/>	<p>\setCJKmonoscale</p> <p>\setCJKmonoexspace</p> <p>\setCJKmonoscale 和 \setCJKmonoexspace 命令分别计算并设置当前字号下西文等宽字体与 CJK 字体的放缩比例和字距,以符合上面所说的 Mono=Scale 与 Mono=Exspace 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/> \setCJKfamilyfont ☆ <hr/>	<p>\setCJKfamilyfont $\langle family \rangle$ [$\langle font features \rangle$] $\langle font name \rangle$</p> <p>声明新的 CJK 字体族 $\langle family \rangle$ 并指定字体。</p>
<hr/> \CJKfamily <hr/>	<p>\CJKfamily $\langle family \rangle$</p> <p>用于在文档中切换 CJK 字体族,$\langle family \rangle$ 必须预先声明。</p>
<hr/> \newCJKfontfamily ☆ <hr/>	<p>\newCJKfontfamily [$\langle family \rangle$] \font-switch [$\langle font features \rangle$] $\langle font name \rangle$</p> <p>声明新的 CJK 字体族 $\langle family \rangle$ 并指定字体,并定义 \font-switch,在文档中可以使用它来切换 CJK 字体族。可以不必指定 $\langle family \rangle$,这时候 $\langle family \rangle$ 将等于 $\langle font-switch \rangle$。事实上,\\newCJKfontfamily 是 \\setCJKfamilyfont 和 \\CJKfamily 的合并。例如</p> <p style="padding-left: 40px;">\newCJKfontfamily[song]\songti{SimSun}</p> <p>等价于</p> <p style="padding-left: 40px;">\setCJKfamilyfont{song}{SimSun}</p> <p style="padding-left: 40px;">\newcommand\songti{\CJKfamily{song}}</p>
<hr/> \CJKfontspec <hr/>	<p>\CJKfontspec [$\langle font features \rangle$] $\langle font name \rangle$</p> <p>在文档中随机定义新的 CJK 字体族,并马上使用它。</p>
<hr/> \defaultCJKfontfeatures ☆ <hr/>	<p>\defaultCJKfontfeatures $\langle font features \rangle$</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;">\defaultCJKfontfeatures{Scale=0.962216}</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。

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

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

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

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

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

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

例如, 可以使用

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

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

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

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

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

在文档中取消对 CJK 分区的声明。

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

在文档中恢复对 CJK 分区的声明。

3.4 设置 CJK 字符范围

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

`<class range>` 的格式和 3.3 节的 `<block range>` 相同。`<class>` 的有效值见源代码(第 5.1 节)。xeCJK 已经支持 Unicode 中所有 CJK 文字和标点。一般来说, 不要轻易改变字符类别。

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

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

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

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

3.5 设置标点宽度和间距

这里的设置对除了 plain 以外的所有标点处理格式都有效。

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

`<标点列表>` 可以是单个标点, 也可以是多个标点。例如,

```
\xeCJKsetwidth{。?}{0.7em}
```

将设置句号和问号所占的宽度为 0.7em。

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

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

```
\xeCJKsetkern{:}{“}{0.3em}
```

4 已知问题和兼容性

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

可以在 xeCJK 之后载入 CJKfntef 宏包,以实现汉字加点等。

xeCJK 依赖 Xe_{La}TeX 的 \XeTeXinterchartoks 机制,与使用相同机制的宏包(例如 polyglossia)会存在大小不一的冲突。xeCJK 虽然为此作了一些处理,但与它们共同使用时应该小心。

5 xeCJK 代码实现

```
1 <{*package>
2 \ProvidesExplPackage{\ExplFileName}{\ExplFileDate}{3.0.6}{\ExplFileDescription}

\XeCJKmsgnew:nn
\XeCJKerror:n
\XeCJKerror:nx
\XeCJKwarning:nx
\XeCJKinfo:nx
\XeCJKinfo:nxx
3 \cs_new_nopar:Npn \XeCJKmsgnew:nn { \msg_new:nnn { xeCJK } }
4 \cs_new_nopar:Npn \XeCJKerror:n { \msg_error:nn { xeCJK } }
5 \cs_new_nopar:Npn \XeCJKerror:nx { \msg_error:nnx { xeCJK } }
6 \cs_new_nopar:Npn \XeCJKwarning:n { \msg_warning:nn { xeCJK } }
7 \cs_new_nopar:Npn \XeCJKwarning:nx { \msg_warning:nnx { xeCJK } }
8 \cs_new_nopar:Npn \XeCJKwarning:nxx { \msg_warning:nnxx { xeCJK } }
9 \cs_new_nopar:Npn \XeCJKinfo:nx { \msg_info:nnx { xeCJK } }
10 \cs_new_nopar:Npn \XeCJKinfo:nxx { \msg_info:nnxx { xeCJK } }
(End definition for \XeCJKmsgnew:nn and others.)
XeCJK 必须使用 XeLaTeX 引擎的支持。
11 \XeCJKmsgnew:nn { Require-XeTeX }
12 {
13   The~XeCJK~package~requires~XeTeX~to~function.\\
14   You~must~change~your~typesetting~engine~to~"xelatex" \\
15   instead~of~"plain"~"latex"~or~"pdflatex"~or~"lualatex".
16 }
17 \xetex_if_engine:F { \msg_critical:nn { xeCJK } { Require-XeTeX } }
18 \XeCJKmsgnew:nn { l3-too-old }
19 {
20   Support~package~'expl3'~too~old. \\
21   Please~update~an~up~to~date~version~of~the~bundles\\
22   'l3kernel'~and~'l3packages'\\
23   using~your~TeX~package~manager~or~from~CTAN.
24 }
25 \ifpackageafter { expl3 } { 2012/02/19 } { } { \XeCJKerror:n { l3-too-old } }
26 \RequirePackage{xparse}
27 \RequirePackage{l3keys2e}

\XeTeXglyphbouds 可以得到一个字符的左右边距,用于标点压缩。
28 \XeCJKmsgnew:nn { XeTeX-too-old }
29 {
30   \token_to_str:N \XeTeXglyphbouds\c_space_tl is~not~defined.\\
31   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
32 }
```

```

33 \cs_if_free:NT \XeTeXglyphbounds
34 {
35   \xeCJK_error:n { XeTeX-too-old }
36   \AtBeginDocument { \punctstyle { plain } }
37 }

```

抑制 BOM。

```

38 \char_set_catcode_ignore:n { "FEFF }

```

`\xeCJKactive` `\xeCJK` 功能开关。

```

39 \keys_define:nn { \xeCJK / options }
40 {
41   \xeCJKactive .choice:,
42   \xeCJKactive / true .code:n = { \makexeCJKactive },
43   \xeCJKactive / false .code:n = { \makexeCJKinactive },
44   \xeCJKactive .default:n = { true },
45 }

```

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

`\makexeCJKactive`

`\makexeCJKinactive`

```

46 \NewDocumentCommand \makexeCJKactive { } { \XeTeXinterchartokenstate = \c_one }
47 \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 \XeTeX 中预定义的类别, `\xeCJK` 新增加了 `HalfLeft`、`HalfRight`、`NormalSpace`。

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

`\xeCJK_Default_class` 这五类是 \XeTeX 预定义的类别。

```

\xeCJK_CJK_class      48 \int_const:Nn \xeCJK_Default_class { 0 }
\xeCJK_FullLeft_class 49 \int_const:Nn \xeCJK_CJK_class { 1 }
\xeCJK_FullRight_class 50 \int_const:Nn \xeCJK_FullLeft_class { 2 }
\xeCJK_Boundary_class 51 \int_const:Nn \xeCJK_FullRight_class { 3 }
                    52 \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。

```

\xg_xeCJK_class_seq 53 \cs_new_nopar:Nn \xeCJK_new_class:n
54 {
55   \exp_args:Nc \newXeTeXintercharclass { xeCJK_#1_class }
56   \clist_gclear_new:c { g_xeCJK_#1_range_clist }
57   \seq_gput_right:Nv \g_xeCJK_class_seq { xeCJK_#1_class }
58 }
59 \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。

```

\xg_xeCJK_base_class_clist
\xg_xeCJK_CJK_class_clist 60 \xeCJK_new_class:n { HalfLeft }
\xg_xeCJK_CJK_class_prop 61 \xeCJK_new_class:n { HalfRight }
62 \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 63 \clist_set:Nn \g_xeCJK_base_class_clist
\g_xeCJK_CJK_class_prop 64 {
65   Default, CJK, FullLeft, FullRight, Boundary,
66   HalfLeft, HalfRight, NormalSpace
67 }
68 \clist_map_inline:Nn \g_xeCJK_base_class_clist
69 { \clist_gclear_new:c { g_xeCJK_#1_range_clist } }
70 \clist_new:N \g_xeCJK_CJK_class_clist
71 \prop_new:N \g_xeCJK_CJK_class_prop
72 \cs_new_nopar:Nn \xeCJK_save_CJK_class:n
73 {
74   \clist_gput_right:Nx \g_xeCJK_CJK_class_clist {#1}
75   \prop_gput:Nvx \g_xeCJK_CJK_class_prop { xeCJK_#1_class } {#1}
76 }
77 \cs_generate_variant:Nn \prop_gput:Nnn { Nvx }
78 \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。

```

79 \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 80 \prg_set_conditional:Nnn \tl_if_blank:x { p , T , F , TF }
81 {
82   \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
83     \prg_return_true:
84   \else:
85     \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
86       \prg_return_true: \else: \prg_return_false: \fi:
87     \fi:
88 }
(End definition for \tl_if_blank:x.)

```

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

`\xeCJKsetcharclass_aux:Nn`

```

89 \NewDocumentCommand \xeCJKDeclareCharClass { > { \TrimSpaces } m m }
90 {
91   \clist_gput_right:cx { g_xeCJK_#1_range_clist } {#2}
92   \clist_map_inline:xn {#2}
93   {
94     \str_if_eq:xxF {##1} { -> }
95     {
96       \xeCJKsetcharclass_aux:Nn \xeCJKsetcharclass {##1}
97       { \xeCJK_class_num:n {#1} }
98     }
99   }
100 }
101 \NewDocumentCommand \xeCJKsetcharclass_aux:Nn
102 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
103 \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。

```

104 \NewDocumentCommand \xeCJKsetcharclass { m m m }
105 {
106   \xeCJK_check_num_range:nnNN {#1} {#2} \l_tmpa_int \l_tmpb_int
107   \int_set:Nn \l_tmpc_int {#3}
108   \prop_if_in:NVTF \g_xeCJK_CJK_class_prop \l_tmpc_int
109   { \cs_set_eq:NN \xeCJK_set_catcode:n \char_set_catcode_other:n }
110   { \cs_set_eq:NN \xeCJK_set_catcode:n \use_none:n }
111   \loop
112     \xeCJK_set_catcode:n \l_tmpa_int
113     \XeTeXcharclass \l_tmpa_int = \l_tmpc_int
114     \if_int_compare:w \l_tmpa_int < \l_tmpb_int
115       \int_incr:N \l_tmpa_int
116   \repeat
117 }
(End definition for \xeCJKsetcharclass.)

```

`\xeCJK_check_num_range:nnNN`

```

118 \cs_new_nopar:Nn \xeCJK_check_num_range:nnNN
119 {
120   \bool_if:nTF { \tl_if_blank_p:x {#1} || \tl_if_blank_p:x {#2} }
121   {
122     \int_set:Nn #3 { \tl_if_blank:xTF {#1} {#2} {#1} }
123     \int_set_eq:NN #3 #4
124   }
125   {
126     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
127     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
128   }
129 }
(End definition for \xeCJK_check_num_range:nnNN.)

```

\xeCJKResetPunctClass 用于重置标点符号所属的字符类。

```
130 \NewDocumentCommand \xeCJKResetPunctClass { }
131 {
132   \xeCJKDeclareCharClass { HalfLeft }
133     { "28 , "2D , "5B , "60 , "7B }
134   \xeCJKDeclareCharClass { HalfRight }
135     { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , }
136   \xeCJKDeclareCharClass { FullLeft }
137     {
138       "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
139       "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
140       "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
141       "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6 ,
142     }
143   \xeCJKDeclareCharClass { FullRight }
144     {
145       "00B7 , "2019 , "201D , "2014 , "2025 , "2026 , "2030 , "2500 , "3001 ,
146       "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 , "3015 ,
147       "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 , "3047 ,
148       "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C , "309D ,
149       "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
150       "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE , "FE50 ,
151       "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C , "FE5E ,
152       "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B , "FF1F ,
153       "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 , "FF69 ,
154       "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E , "FF9F ,
155     }
156 }
```

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

设置CJK文字的范围。

```
157 \xeCJKDeclareCharClass { CJK }
158 {
159   • Hangul Jamo (谚文字母)
160     "1100 -> "11FF ,
161   • CJK Radicals Supplement (中日韩部首补充)
162     "2E80 -> "2EFF ,
163   • Kangxi Radicals (康熙字典部首)
164     "2F00 -> "2FDF ,
165   • Ideographic Description Characters (表意文字描述符)
166     "2FF0 -> "2FFF ,
167   • CJK Symbols and Punctuation (中日韩符号和标点)
168     "3000 -> "303F ,
169   • Hiragana (日文平假名)
170     "3040 -> "309F ,
```

- Katakana (日文片假名)
165 "30A0 -> "30FF ,
- Bopomofo (注音字母)
166 "3100 -> "312F ,
- Hangul Compatibility Jamo (谚文兼容字母)
167 "3130 -> "318F ,
- Kanbun (象形字注释标志)
168 "3190 -> "319F ,
- Bopomofo Extended (注音字母扩展)
169 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)
170 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)
171 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
172 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
173 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
174 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
175 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
176 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
177 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
178 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
179 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
180 "AC00 -> "D7AF ,

- Kana Supplement (日文假名增补)

181 "B000 -> "B0FF ,

- Hangul Jamo Extended-B (谚文扩展 B)

182 "D7B0 -> "D7FF ,

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

183 "F900 -> "FAFF ,

- CJK Compatibility Forms (中日韩兼容形式)

184 "FE30 -> "FE4F ,

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

185 "FF00 -> "FFEF ,

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

186 "20000 -> "2A6DF ,

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

187 "2A700 -> "2B73F ,

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

188 "2B740 -> "2B81F ,

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

189 "2F800 -> "2FA1F ,

190 }

重置标点符号的字符类。

191 \xeCJKResetPunctClass

\normalspacedchars 声明前后不加间距的字符。

192 \NewDocumentCommand \normalspacedchars { m }

193 {

194 \tl_map_inline:nn {#1}

195 { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }

196 }

197 \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

198 \cs_new_nopar:Nn \xeCJK_inter_class_toks:nnn

199 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = {#3} }

200 \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。

```
201 \cs_new_nopar:Nn \xeCJK_clear_inter_class_toks:nn
202 {
203   \XeTeXinterchartoks
204   \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} = { \c_empty_tl }
205 }
(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。

```
206 \cs_new_nopar:Nn \xeCJK_pre_inter_class_toks:nnn
207 {
208   \use:x
209   {
210     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
211     {
212       \exp_not:n {#3} \tex_the:D \XeTeXinterchartoks
213       \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2}
214     }
215   }
216 }
(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 217 \cs_new_nopar:Nn \xeCJK_app_inter_class_toks:nnn
218 {
219   \use:x
220   {
221     \XeTeXinterchartoks \xeCJK_class_num:n {#1} \xeCJK_class_num:n {#2} =
222     {
223       \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1}
224       \xeCJK_class_num:n {#2} \exp_not:n {#3}
225     }
226   }
227 }
228 \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。

```
229 \cs_new_nopar:Nn \xeCJK_get_inter_class_toks:nn
230 { \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。

```
231 \cs_new_nopar:Nn \xeCJK_copy_inter_class_toks:nnnn
232 {
233   \use:x
234   {
235     \xeCJK_inter_class_toks:nnn {#1} {#2}
236     { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
237   }
238 }
(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。

```

239 \cs_new_nopar:Nn \xeCJK_clear_CJK_toks:
240 {
241   \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
242     { \xeCJK_clear_inter_class_toks:nn { Boundary } {##1} }
243 }
(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	✓	✓	✓	✓	✓		✓	

```

244 \clist_map_inline:nn { Default, HalfLeft, HalfRight, NormalSpace }
245 {
246   \clist_map_inline:nn { CJK, FullLeft, FullRight }
247     {
248       \xeCJK_inter_class_toks:nnn {#1} {##1}
249       {
250         \c_group_begin_token
251         \xeCJK_select_font:
252         \xeCJK_clear_inter_class_toks:nn {#1} {##1}
253         \xeCJK_clear_CJK_toks:
254       }
255       \xeCJK_inter_class_toks:nnn {##1} {#1} { \c_group_end_token }
256     }
257   \xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
258   \clist_map_inline:nn { FullLeft, FullRight }
259     { \xeCJK_app_inter_class_toks:nnc {#1} {##1} { xeCJK_CJK_and_##1:N } }
260 }
261 \clist_map_inline:nn { Default, HalfLeft, NormalSpace }
262 {
263   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
264   {
265     \bool_if:nTF
266     {
267       \l_xeCJK_xecglue_bool &&
268       \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
269       \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl
270     }
271     {
272       \tex_unskip:D
273       \bool_if:nTF
274       {

```

```

275         \int_compare_p:nNn \tex_lastkern:D = \c_one ||
276         \int_compare_p:nNn \tex_lastkern:D = \c_two
277     }
278     { \CJKecglue } { \c_space_token }
279 }
280 {
281     \int_compare:nNnTF \tex_lastkern:D = \c_one { \CJKecglue }
282     { \int_compare:nNnT \tex_lastkern:D = \c_two \xeCJK_space_or_xecglue: }
283 }
284 }
285 \str_if_eq:xxF {#1} { NormalSpace }
286 { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
287 }
288 \clist_map_inline:nn { Default, HalfRight, NormalSpace }
289 {
290     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
291     {
292         \peek_catcode:NTF \c_space_token
293         { { \xeCJK_default_space_kern: } } { { \xeCJK_default_kern: } }
294     }
295     \str_if_eq:xxF {#1} { NormalSpace }
296     { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
297 }
298 \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }

```

\xeCJK_Boundary_and_CJK:

```

299 \cs_new_nopar:Nn \xeCJK_Boundary_and_CJK:
300 {
301     {
302         \bool_if:nTF
303         {
304             \int_compare_p:nNn \tex_lastkern:D = \c_one ||
305             \int_compare_p:nNn \tex_lastkern:D = \c_two
306         }
307         { \CJKglue }
308         {
309             \bool_if:nTF
310             {
311                 \int_compare_p:nNn \tex_lastkern:D = \c_three ||
312                 \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
313                 ( \int_compare_p:nNn \etex_lastnodetype:D = \c_twelve &&
314                   \g_xeCJK_itcorr_bool )
315             }
316             { \CJKecglue }
317             {
318                 \bool_if:nT
319                 {
320                     \l_xeCJK_xecglue_bool &&
321                     \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
322                     ( \skip_if_eq_p:nn \tex_lastskip:D \l_xeCJK_space_glue_tl ||
323                       \int_compare_p:nNn \tex_lastskip:D = \l_xeCJK_ecglue_dim )
324                 }

```

```

325         {
326             \skip_set_eq:NN \l_tmpa_skip \tex_lastskip:D
327             \tex_unskip:D
328             \bool_if:nTF
329             {
330                 \int_compare_p:nNn \tex_lastkern:D = \c_four      ||
331                 \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
332                 \int_compare_p:nNn \tex_lastkern:D = \c_three     ||
333                 \g_xeCJK_itcorr_bool
334             }
335             { \CJKecglue }
336             {
337                 \bool_if:nT
338                 {
339                     ! ( \int_compare_p:nNn \tex_lastkern:D = \c_one ) ||
340                     ! ( \int_compare_p:nNn \tex_lastkern:D = \c_two ) ||
341                     \l_xeCJK_space_bool
342                 }
343                 { \skip_horizontal:N \l_tmpa_skip }
344             }
345         }
346     }
347 }
348 \bool_gset_false:N \g_xeCJK_itcorr_bool
349 }
350 \c_group_begin_token
351 \xeCJK_clear_CJK_toks:
352 \xeCJK_select_font:
353 \CJKsymbol
354 }
355 \tl_set:Nn \l_xeCJK_space_glue_tl
356 {
357     \tex_fontdimen:D \c_two \tex_font:D
358     \@plus \tex_fontdimen:D \c_three \tex_font:D
359     \@minus \tex_fontdimen:D \c_four \tex_font:D
360 }
361 (End definition for \xeCJK_Boundary_and_CJK:.)
362 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
363 {
364     \c_group_end_token
365     { \xeCJK_CJK_kern: }
366     \xeCJK_ignorespaces:
367 }
368 \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }
369 \xeCJK_CJK_and_CJK:N
370 \cs_new_nopar:Nn \xeCJK_CJK_and_CJK:N { \CJKglue \CJKsymbol {#1} }
371 (End definition for \xeCJK_CJK_and_CJK:N.)
372 \clist_map_inline:nn { FullLeft, FullRight }
373 {
374     \xeCJK_inter_class_toks:nnn { Boundary } {#1}

```

```

372     {
373         \c_group_begin_token
374         \xeCJK_clear_CJK_toks:
375         \xeCJK_select_font:
376         \use:c { xeCJK_CJK_and_#1:N }
377     }
378     \xeCJK_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
379     \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \c_group_end_token \tex_ignorespaces:D }
380 }
381 \clist_map_inline:Nn \g_xeCJK_base_class_clist
382 {
383     \bool_if:nF
384     { \str_if_eq_p:xx {#1} { FullLeft } || \str_if_eq_p:xx {#1} { FullRight } }
385     {
386         \xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \nobreak }
387         \xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \xeCJK_after_FullRight: }
388     }
389 }
390 \clist_map_inline:nn { CJK, FullLeft, FullRight }
391 {
392     \clist_map_inline:nn { FullLeft, FullRight }
393     { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
394 }

```

`\xeCJK_punct_rule:n` 用于抹去标点符号的左/右空白。

```

395 \cs_new_nopar:Nn \xeCJK_punct_rule:n
396 {
397     \tex_vrule:D
398     \@width \dim_eval:c { \l_xeCJK_punct_coord_tl/rule/#1 }
399     \@depth \c_zero_dim
400     \@height \c_zero_dim \scan_stop:
401 }

```

(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。

```

402 \cs_new_nopar:Nn \xeCJK_punct_glue:n
403 {
404     \skip_horizontal:n
405     {
406         \dim_eval:c { \l_xeCJK_punct_coord_tl/glue/#1 }
407         \@plus \dim_eval:n { \tl_use:c { \l_xeCJK_punct_coord_tl/glue/#1 } /4 }
408         \@minus \dim_eval:n { \tl_use:c { \l_xeCJK_punct_coord_tl/glue/#1 } /2 }
409     }
410 }

```

(End definition for \xeCJK_punct_glue:n.)

`\xeCJK_punct_kern:n` 相邻两个标点之间的间距。

```

411 \cs_new_nopar:Nn \xeCJK_punct_kern:n
412 { \tex_kern:D \dim_eval:c { \l_xeCJK_punct_coord_tl/kern/#1 } }

```

(End definition for \xeCJK_punct_kern:n.)

```

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

\XeCJK_before_FullLeft:N
414 \cs_new_nopar:Nn \XeCJK_before_FullLeft:N
415 {
416   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
417   \XeCJK_punct_rule:n { 1/#1 }
418   \CJKpunctsymbol {#1}
419 }
(End definition for \XeCJK_before_FullLeft:N.)

\XeCJK_after_FullRight:
420 \cs_new_nopar:Nn \XeCJK_after_FullRight:
421 {
422   \XeCJK_get_punct_bounds:nV { r } \l_xeCJK_lastpunct_tl
423   \XeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
424   \XeCJK_punct_glue:n { r/\l_xeCJK_lastpunct_tl }
425 }
(End definition for \XeCJK_after_FullRight:.)

\XeCJK_CJK_and_FullLeft:N
426 \cs_new_nopar:Nn \XeCJK_CJK_and_FullLeft:N
427 {
428   \XeCJK_get_punct_bounds:nN { l } {#1}
429   \int_compare:nNnF \etex_lastnodetype:D = \c_one
430     { \XeCJK_punct_glue:n { 1/#1 } }
431   \XeCJK_before_FullLeft:N {#1}
432   \XeCJK_punct_if_middle:NT {#1}
433   {
434     \XeCJK_punct_rule:n { m/#1 }
435     \XeCJK_punct_glue:n { 1/#1 }
436   }
437 }
(End definition for \XeCJK_CJK_and_FullLeft:N.)

\XeCJK_CJK_and_FullRight:N
438 \cs_new_nopar:Nn \XeCJK_CJK_and_FullRight:N
439 {
440   \XeCJK_get_punct_bounds:nN { r } {#1}
441   \XeCJK_punct_if_long:NTF {#1} \CJKglue
442   {
443     \nobreak
444     \XeCJK_punct_if_middle:NT {#1}
445     {
446       \XeCJK_punct_glue:n { r/#1 }
447       \XeCJK_punct_rule:n { m/#1 }
448     }
449   }
450   \tl_gset:Nx \l_xeCJK_lastpunct_tl {#1}
451   \CJKpunctsymbol {#1}
452 }

```

(End definition for \xeCJK_CJK_and_FullRight:N.)

\xeCJK_FullLeft_and_FullLeft:N

```

453 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullLeft:N
454 {
455   \nobreak
456   \xeCJK_get_punct_bounds:nN { l } {#1}
457   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
458   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
459   \xeCJK_before_FullLeft:N {#1}
460 }

```

(End definition for \xeCJK_FullLeft_and_FullLeft:N.)

\xeCJK_FullLeft_and_FullRight:N

```

461 \cs_new_nopar:Nn \xeCJK_FullLeft_and_FullRight:N
462 {
463   \nobreak
464   \xeCJK_get_punct_bounds:nN { r } {#1}
465   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
466   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
467   \nobreak
468   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
469   \CJKpunctsymbol {#1}
470 }

```

(End definition for \xeCJK_FullLeft_and_FullRight:N.)

\xeCJK_FullRight_and_FullLeft:N

```

471 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullLeft:N
472 {
473   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
474   \xeCJK_get_punct_bounds:nN { l } {#1}
475   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
476   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
477   \xeCJK_punct_nobreak:
478   \xeCJK_before_FullLeft:N {#1}
479 }

```

(End definition for \xeCJK_FullRight_and_FullLeft:N.)

\xeCJK_FullRight_and_FullRight:N

```

480 \cs_new_nopar:Nn \xeCJK_FullRight_and_FullRight:N
481 {
482   \xeCJK_punct_rule:n { r/\l_xeCJK_lastpunct_tl }
483   \xeCJK_get_punct_bounds:nN { r } {#1}
484   \xeCJK_get_kern:VN \l_xeCJK_lastpunct_tl {#1}
485   \xeCJK_punct_kern:n { \l_xeCJK_lastpunct_tl - #1 }
486   \nobreak
487   \tl_set:Nx \l_xeCJK_lastpunct_tl {#1}
488   \CJKpunctsymbol {#1}
489 }

```

(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: 490 \cs_new_nopar:Nn \xeCJK_CJK_kern: { \tex_kern:D -1 sp \tex_kern:D 1 sp }
\xeCJK_default_kern: 491 \cs_new_nopar:Nn \xeCJK_CJK_space_kern: { \tex_kern:D -2 sp \tex_kern:D 2 sp }
\xeCJK_default_space_kern: 492 \cs_new_nopar:Nn \xeCJK_default_kern: { \tex_kern:D -3 sp \tex_kern:D 3 sp }
493 \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。

```

494 \keys_define:nn { xCJK / 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 495 \keys_define:nn { xCJK / options }
496 {
497   CJKecglue .code:n =
498   {
499     \cs_set:Npx \CJKecglue {#1}
500     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
501   },
502   xCJKecglue .choice:,
503   xCJKecglue / true .code:n =
504   {
505     \bool_set_true:N \l_xeCJK_xecglue_bool
506     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
507   },
508   xCJKecglue / false .code:n =
509   {
510     \bool_set_false:N \l_xeCJK_xecglue_bool
511     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_token
512   },
513   xCJKecglue / unknown .code:n =
514   {
515     \bool_set_true:N \l_xeCJK_xecglue_bool
516     \cs_set:Npx \CJKecglue {#1}
517     \settowidth \l_xeCJK_ecglue_dim \CJKecglue
518     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
519   },
520   xCJKecglue .default:n = { true } ,
521 }
522 \dim_new:N \l_xeCJK_ecglue_dim
523 \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。

```

524 \keys_define:nn { xCJK / options }
525 {
526   CJKspace .choice:,
527   CJKspace / true .code:n = { \CJKspace },
528   CJKspace / false .code:n = { \CJKnospace },
529   CJKspace .default:n = { true },
530   space .meta:n = { CJKspace = true },
531   nospace .meta:n = { CJKspace = false },
532 }

```

(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 533 \NewDocumentCommand \CJKspace { }
534 {
535   \bool_set_true:N \l_xeCJK_space_bool
536   \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_peek_math:
537 }
538 \NewDocumentCommand \CJKnospace { }
539 {
540   \bool_set_false:N \l_xeCJK_space_bool
541   \cs_set_eq:NN \xeCJK_ignorespaces: \xeCJK_ignore_spaces:
542 }
```

(End definition for \CJKspace and \CJKnospace.)

\xeCJK_ignore_spaces:

```
\xeCJK_peek_math: 543 \cs_new_nopar:Nn \xeCJK_ignore_spaces:
544 {
545   \peek_catcode:NTF \c_space_token
546   {
547     \int_compare:nNt \tex_lastkern:D = \c_one
548     { \tex_unkern:D \tex_unkern:D { \xeCJK_CJK_space_kern: } }
549     \peek_catcode_ignore_spaces:NTF \c_math_toggle_token
550     { \xeCJK_space_or_xecglue: }
551     {
552       \bool_if:nT
553       {
554         \xeCJK_if_outer_macro_p:c { l_peek_token } ||
555         \token_if_cs_p:c { l_peek_token }
556       }
557       { \xeCJK_space_or_xecglue: }
558     }
559   }
560   {
561     \xeCJK_if_outer_macro:cF { l_peek_token }
562     {
563       \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
564       { \cs_if_exist:cF { l_peek_token } \exp_not:N }
565     }
566   }
567 }
568 \cs_new_nopar:Nn \xeCJK_peek_math:
569 {
570   \peek_catcode:NTF \c_space_token
571   {
572     \int_compare:nNt \tex_lastkern:D = \c_one
573     { \tex_unkern:D \tex_unkern:D { \xeCJK_CJK_space_kern: } }
574     \peek_catcode_ignore_spaces:NF \c_space_token { \xeCJK_space_or_xecglue: }
575   }
576   {
577     \xeCJK_if_outer_macro:cF { l_peek_token }
578     {
```

```

579         \token_if_math_toggle:cTF { l_peek_token } \CJKecglue
580         { \cs_if_exist:cF { l_peek_token } \exp_not:N }
581     }
582 }
583 }

```

(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 584 \keys_define:nn { xeCJK / options }
585 {
586     CheckSingle .choice:,
587     CheckSingle / true .code:n = { \xeCJKenablechecksingle },
588     CheckSingle / false .code:n = { \xeCJKdisablechecksingle },
589     CheckSingle .default:n = { true },
590     CJKchecksingle .meta:n = { CheckSingle = true },
591     CheckSingleglue .tl_set_x:N = { \CJKchecksingleglue },
592 }

```

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

\xeCJKenablechecksingle 孤字处理功能开关。

```

\xeCJKdisablechecksingle 593 \bool_new:N \l_xeCJK_checksingle_bool
594 \NewDocumentCommand \xeCJKenablechecksingle { }
595 {
596     \bool_if:NF \l_xeCJK_checksingle_bool
597     {
598         \bool_set_true:N \l_xeCJK_checksingle_bool
599         \cs_set_eq:NN \xeCJK_checksingle_save_CJKsymbol:N \xeCJK_CJK_and_CJK:N
600         \cs_set_eq:NN \xeCJK_checksingle_save_CJKglue \CJKglue
601         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle:N
602         \cs_set_nopar:Npn \CJKglue { \CJKchecksingleglue }
603     }
604 }
605 \NewDocumentCommand \xeCJKdisablechecksingle { }
606 {
607     \bool_if:NT \l_xeCJK_checksingle_bool
608     {
609         \bool_set_false:N \l_xeCJK_checksingle_bool
610         \cs_set_eq:NN \CJKglue \xeCJK_checksingle_save_CJKglue
611         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_checksingle_save_CJKsymbol:N
612     }
613 }

```

(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle.)

```

\xeCJK_checksingle:N
\xeCJK_checksingle:NN 614 \cs_new_nopar:Nn \xeCJK_checksingle:N
\xeCJK_checksingle:NNN 615 {
616     \peek_catcode:NTF \c_catcode_other_token
617     { \xeCJK_checksingle:NN {#1} }
618     {
619         \bool_if:nTF

```

```

620     {
621         ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } )           &&
622         \tl_if_empty_p:x { \token_get_arg_spec:c { l_peek_token } } &&
623         \exp_args:NNc \exp_args:No \tl_if_single_token_p:n { l_peek_token } &&
624         \exp_args:NNc \exp_after:wN \token_if_other_p:N { l_peek_token }
625     }
626     { \xeCJK_checksingl:NN {#1} }
627     { \xeCJK_checksingl_save_CJKsymbol:N {#1} }
628 }
629 }
630 \cs_new_nopar:Nn \xeCJK_checksingl:NN
631 {
632     \peek_catcode:NTF \c_catcode_other_token
633     { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
634     {
635         \bool_if:nTF
636         {
637             ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
638             ( \token_if_space_p:c { l_peek_token } ||
639               \token_if_math_toggle_p:c { l_peek_token } ||
640               \token_if_cs_p:c { l_peek_token } )
641         }
642         {
643             \bool_if:nTF { \token_if_space_p:c { l_peek_token } }
644             {
645                 \peek_catcode_ignore_spaces:NTF \c_catcode_other_token
646                 { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
647                 {
648                     \bool_if:nTF
649                     {
650                         ! ( \xeCJK_if_outer_macro_p:c { l_peek_token } ) &&
651                         ( \token_if_math_toggle_p:c { l_peek_token } ||
652                           \token_if_cs_p:c { l_peek_token } )
653                     }
654                     { \xeCJK_checksingl:NNN {#1} {#2} \c_space_token }
655                     { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 \c_space_token }
656                 }
657             }
658             { \xeCJK_checksingl:NNN {#1} {#2} { } }
659         }
660         { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2 }
661     }
662 }
663 \cs_new_nopar:Nn \xeCJK_checksingl:NNN
664 {
665     \bool_if:nTF
666     {
667         \token_if_eq_meaning_p:NN \l_peek_token \par ||
668         \token_if_eq_meaning_p:NN \l_peek_token \[ % \
669     }
670     { \CJKsymbol {#1} \nobreak #2 }
671     { \xeCJK_checksingl_env:NNN {#1} {#2} {#3} }

```

672 }
 (End definition for \xeCJK_checksingl:N, \xeCJK_checksingl:NN, and \xeCJK_checksingl:NNN.)

\xeCJK_checksingl_env:NNN
 \xeCJK_checksingl_env:NNNNn 673 \cs_new_nopar:Nn \xeCJK_checksingl_env:NNN
 674 {
 675 \peek_catcode_remove:NTF \c_math_toggle_token
 676 {
 677 \peek_catcode:NTF \c_math_toggle_token
 678 { \CJKsymbol {#1} \nobreak #2 \c_math_toggle_token }
 679 { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 \c_math_toggle_token }
 680 }
 681 {
 682 \peek_meaning_remove:NTF \begin
 683 { \xeCJK_checksingl_env:NNNNn {#1} {#2} {#3} \begin }
 684 {
 685 \peek_meaning_remove:NTF \end
 686 { \xeCJK_checksingl_env:NNNNn {#1} {#2} {#3} \end }
 687 { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 }
 688 }
 689 }
 690 }
 691 \cs_new_nopar:Nn \xeCJK_checksingl_env:NNNNn
 692 {
 693 \clist_if_in:NnTF \g_xeCJK_inline_env_clist {#5}
 694 { \xeCJK_checksingl_save_CJKsymbol:N {#1} #2#3 }
 695 { \CJKsymbol {#1} \nobreak #2 }
 696 \scan_stop: #4 {#5}
 697 }
 (End definition for \xeCJK_checksingl_env:NNN and \xeCJK_checksingl_env:NNNNn.)

InlineEnv
 InlineEnv+ 698 \keys_define:nn { xeCJK / options }
 InlineEnv- 699 {
 700 InlineEnv .clist_gset:N = \g_xeCJK_inline_env_clist ,
 701 InlineEnv+ .code:n =
 702 {
 703 \clist_gput_right:Nx \g_xeCJK_inline_env_clist {#1}
 704 \clist_gremove_duplicates:N \g_xeCJK_inline_env_clist
 705 },
 706 InlineEnv- .code:n =
 707 {
 708 \clist_map_inline:xn {#1}
 709 { \clist_gremove_all:Nn \g_xeCJK_inline_env_clist {##1} }
 710 },
 711 }

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

712 \cs_generate_variant:Nn \token_if_cs_p:N { c }
 713 \cs_generate_variant:Nn \token_if_space_p:N { c }
 714 \cs_generate_variant:Nn \token_get_arg_spec:N { c }
 715 \cs_generate_variant:Nn \token_if_math_toggle_p:N { c }
 716 \cs_generate_variant:Nn \token_if_math_toggle:NTF { c }

\xeCJK_if_outer_macro:p:c 判断是否是 \outer 宏,输入的是控制序列的名字。

```
\xeCJK_if_outer_macro:cTF
717 \group_begin:
718 \char_set_lccode:nn { ``\; } { ``\: }
719 \char_set_lccode:nn { ``\Z } { ``\t }
720 \char_set_lccode:nn { ``\Y } { ``\c }
721 \char_set_lccode:nn { ``\T } { ``\T }
722 \char_set_lccode:nn { ``\F } { ``\F }
723 \tl_map_function:nN { O U Z E R M A Y R ; } \char_set_catcode_other:N
724 \tl_to_lowercase:n
725 {
726   \group_end:
727   \prg_new_conditional:Nnn \xeCJK_if_outer_macro:c { p , T , F , TF }
728   {
729     \exp_last_unbraced:Nf \xeCJK_if_outer_macro_aux:w
730     \cs_meaning:c {#1} OUZER ~ MAYRO ; \q_stop
731   }
732   \cs_new_nopar:Npn \xeCJK_if_outer_macro_aux:w #1 OUZER ~ MAYRO ; #2 \q_stop
733   {
734     \prg_case_str:xxn { #1 ~ }
735     {
736       { \token_to_str:N \ } { \prg_return_true: }
737       { \token_to_str:N \long \token_to_str:N \ } { \prg_return_true: }
738       { \token_to_str:N \protected \token_to_str:N \ } { \prg_return_true: }
739       { \token_to_str:N \protected \token_to_str:N \long \token_to_str:N \ }
740       { \prg_return_true: }
741     }
742     { \prg_return_false: }
743   }
744 }
(End definition for \xeCJK_if_outer_macro:c)
```

\xeCJK_if_eq_meaning:p:cN 对 l3token 打的补丁,使 \peek_catcode_ignore_spaces:NTF 等可用于 \outer 宏。

```
\xeCJK_if_eq_meaning:cNTF
745 \tl_replace_once:Nnn \peek_ignore_spaces_execute_branches:
746 { \token_if_eq_meaning:NNTF \l_peek_token }
747 { \xeCJK_if_eq_meaning:cNTF { l_peek_token } }
748 \prg_new_conditional:Nnn \xeCJK_if_eq_meaning:cN { p , T , F , TF }
749 {
750   \exp_args:Nc \if_meaning:w {#1} #2
751   \prg_return_true: \else: \prg_return_false: \fi:
752 }
(End definition for \xeCJK_if_eq_meaning:cN.)
```

5.4 增加 CJK 子分区

```
\xeCJK_UL_subclass_patch_tl
\g_xeCJK_CJK_subclass_clist
753 \tl_new:N \xeCJK_UL_subclass_patch_tl
754 \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:
755 \cs_new_nopar:Nn \xeCJK_erase_CJKsymbol:
```

```

756 {
757     \cs_gset_eq:NN \CJKsymbol_Block \CJKsymbol
758     \cs_gset_eq:NN \CJKsymbol \prg_do_nothing:
759 }
760 \cs_new_nopar:Nn \xeCJK_restore_CJKsymbol:
761 {
762     \cs_gset_eq:NN \CJKsymbol \CJKsymbol_Block
763     \CJKsymbol
764 }
(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。

```

765 \NewDocumentCommand \xeCJKDeclareSubCJKBlock { m m }
766 { \xeCJKDeclareSubCharClass { CJK } {#1} {#2} }
767 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock. This function is documented on page 9.)

```

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

```

768 \bool_new:N \l_xeCJK_sub_cancel_bool
769 \NewDocumentCommand \xeCJKCancelSubCJKBlock { m }
770 {
771     \bool_if:NF \l_xeCJK_sub_cancel_bool
772     {
773         \bool_set_true:N \l_xeCJK_sub_cancel_bool
774         \xeCJK_sub_restore_or_cancel:n {#1}
775     }
776 }
777 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { m }
778 {
779     \bool_if:NT \l_xeCJK_sub_cancel_bool
780     {
781         \bool_set_false:N \l_xeCJK_sub_cancel_bool
782         \xeCJK_sub_restore_or_cancel:n {#1}
783     }
784 }
785 \cs_new_nopar:Nn \xeCJK_sub_restore_or_cancel:n
786 {
787     \clist_map_inline:xn {#1}
788     {
789         \cs_if_free:cTF { xeCJK_ CJK/##1 _class }
790         { \xeCJK_error:nx { SubBlock-undefined } {##1} }
791         {
792             \xeCJKDeclareCharClass
793             { CJK \bool_if:NF \l_xeCJK_sub_cancel_bool { /##1 } }
794             { \clist_use:c { g_xeCJK_CJK/##1_range_clist } }
795         }
796     }
797 }
798 \xeCJK_msg_new:nn { SubBlock-undefined }
799 {
800     The~CJK~sub~block~'#1'~is~undefined.\\
801     Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \c_space_tl

```

802 to~declare~it.

803 }

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

\xeCJKDeclareSubCharClass

```
804 \NewDocumentCommand \xeCJKDeclareSubCharClass
805 { > { \TrimSpaces } m > { \TrimSpaces } m m }
806 {
807   \cs_if_free:cT { xeCJK_ #1/#2 _class }
808   {
809     \xeCJK_new_class:n { #1/#2 }
810     \xeCJK_set_sub_class_toks:nn {#1} {#2}
811     \xeCJK_new_sub_key:n {#2}
812     \xeCJK_UL_subclass_patch:nn {#1} {#2}
813   }
814   \xeCJKDeclareCharClass { #1/#2 } {#3}
815 }
```

816 \@onlypreamble \xeCJKDeclareSubCharClass

(End definition for \xeCJKDeclareSubCharClass.)

\xeCJK_set_sub_class_toks:nn

```
817 \cs_new_nopar:Nn \xeCJK_set_sub_class_toks:nn
818 {
819   \clist_map_inline:Nn \g_xeCJK_base_class_clist
820   {
821     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
822     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
823     \str_if_eq:xxTF {##1} { CJK }
824     {
825       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
826       { \xeCJK_switch_font:nn {#1} {#2} }
827     }
828     {
829       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
830       { \xeCJK_erase_CJKsymbol: }
831       \xeCJK_app_inter_class_toks:nnn {##1} { #1/#2 }
832       { \xeCJK_switch_font:nn {#1} {#2} \xeCJK_restore_CJKsymbol: }
833     }
834   }
835   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
836   \clist_if_empty:NF \g_xeCJK_CJK_subclass_clist
837   {
838     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
839     {
840       \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
841       \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
842       \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
843       { \xeCJK_switch_font:nn {#2} {##1} }
844       \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
845       { \xeCJK_switch_font:nn {##1} {#2} }
846     }
847   }
```



```

847     }
848     \clist_gput_right:Nn \g_xeCJK_CJK_subclass_clist {#2}
849     \xeCJK_save_CJK_class:n { #1/#2 }
850     \clist_map_inline:nn { CJK, FullLeft, FullRight }
851     {
852         \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
853         { \xeCJK_switch_font:nn {#2} {#1} }
854     }
855 }
(End definition for \xeCJK_set_sub_class_toks:nn.)

```

\xeCJK_UL_subclass_patch:nn

```

856 \cs_new_nopar:Nn \xeCJK_UL_subclass_patch:nn
857 {
858     \tl_put_right:Nn \xeCJK_UL_subclass_patch_tl
859     {
860         \clist_map_inline:nn { Default, HalfLeft, HalfRight }
861         {
862             \str_if_eq:xxTF {##1} { HalfLeft }
863             { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKecglue \CJKsymbol } }
864             { \xeCJK_inter_class_toks:nnn {#1} { #1/#2 } { \CJKsymbol } }
865             \xeCJK_inter_class_toks:nnn { Boundary } { #1/#2 }
866             {
867                 \xeCJK_erase_CJKsymbol:
868                 \xeCJK_Boundary_and_CJK:
869                 \xeCJK_switch_font:nn {#1} {#2}
870                 \xeCJK_restore_CJKsymbol:
871             }
872         }
873     }
874 }
(End definition for \xeCJK_UL_subclass_patch:nn.)

```

5.5 标点处理

\g_xeCJK_punctstyle_clist 标点处理方式。

```

875 \clist_new:N \g_xeCJK_punctstyle_clist
876 \clist_set:Nn \g_xeCJK_punctstyle_clist
877 { CCT , halfwidth , fullwidth , marginkerning , mixedwidth , plain }
878 \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
879 { \tl_const:cn { c_xeCJK_ps_#1_tl } {#1} }
(End definition for \g_xeCJK_punctstyle_clist.)

```

\keys_choices_generate:V

.generate_choices:V

```

880 \cs_generate_variant:Nn \keys_choices_generate:n { V }
881 \cs_new_protected:cpn { \c_keys_props_root_tl .generate_choices:V } #1
882 { \keys_choices_generate:V {#1} }
(End definition for \keys_choices_generate:V and .generate_choices:V.)

```

PunctStyle 相关选项声明。

```

AllowBreakBetweenPuncts 883 \keys_define:nn { xeCJK / options }
KaiMingPunct 884 {
  LongPunct 885 AllowBreakBetweenPuncts .choice:,
  MiddlePunct 886 AllowBreakBetweenPuncts / true .code:n = { \xeCJKallowbreakbetweenpuncts },
  PunctWidth 887 AllowBreakBetweenPuncts / false .code:n = { \xeCJKnobreakbetweenpuncts },
  888 AllowBreakBetweenPuncts .default:n = { true },
  889 KaiMingPunct .code:n = { \xeCJK_set_special_punct:nn { mixedwidth } {#1} },
  890 KaiMingPunct+ .code:n = { \xeCJK_add_special_punct:nn { mixedwidth } {#1} },
  891 KaiMingPunct- .code:n = { \xeCJK_sub_special_punct:nn { mixedwidth } {#1} },
  892 LongPunct .code:n = { \xeCJK_set_special_punct:nn { long } {#1} },
  893 LongPunct+ .code:n = { \xeCJK_add_special_punct:nn { long } {#1} },
  894 LongPunct- .code:n = { \xeCJK_sub_special_punct:nn { long } {#1} },
  895 MiddlePunct+ .code:n = { \xeCJK_add_special_punct:nn { middle } {#1} },
  896 MiddlePunct .code:n = { \xeCJK_set_special_punct:nn { middle } {#1} },
  897 MiddlePunct- .code:n = { \xeCJK_sub_special_punct:nn { middle } {#1} },
  898 PunctWidth .tl_gset:N = \g_xeCJK_punct_width_tl ,
  899 PunctStyle .choice_code:n =
  900 {
  901 \tl_set:Nx \l_xeCJK_punctstyle_tl { \l_keys_choice_tl }
  902 \tl_if_eq:NNT \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
  903 { \xeCJKallowbreakbetweenpuncts }
  904 },
  905 PunctStyle .generate_choices:V = \g_xeCJK_punctstyle_clist,
  906 PunctStyle / banjiao .meta:n = { PunctStyle = halfwidth },
  907 PunctStyle / quanjiao .meta:n = { PunctStyle = fullwidth },
  908 PunctStyle / kaiming .meta:n = { PunctStyle = mixedwidth },
  909 PunctStyle / hangmobanjiao .meta:n = { PunctStyle = marginkerning },
  910 PunctStyle / unknown .code:n =
  911 { \xeCJK_error:nx { PunctStyle-undefined } \l_keys_value_tl },
  912 }
  913 \xeCJK_msg_new:nn { PunctStyle-undefined }
  914 {
  915 Punctstyle~"#1"~is~not~available. \\\
  916 You~can~only~use~one~of\\\
  917 "\g_xeCJK_punctstyle_clist".\\
  918 }

```

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

```

\g_xeCJK_allowbreakbetweenpuncts
\g_xeCJK_nobreakbetweenpuncts 919 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
  920 { \cs_set_eq:NN \xeCJK_punct_nobreak: \xeCJK_zero_glue: }
  921 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
  922 { \cs_set_eq:NN \xeCJK_punct_nobreak: \nobreak }
  923 \cs_new_protected_nopar:Nn \xeCJK_zero_glue: { \skip_horizontal:N \c_zero_skip }
  (End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts.)

```

```

\g_xeCJK_special_ps_clist
\g_xeCJK_set_special_punct:nn 924 \clist_set:Nn \g_xeCJK_special_ps_clist { mixedwidth , long , middle }
\g_xeCJK_add_special_punct:nn 925 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
\g_xeCJK_sub_special_punct:nn 926 {
  927 \tl_new:c { l_xeCJK_#1_punct_tl }

```

```

928 \prop_new:c { l_xeCJK_#1_punct_prop }
929 }
930 \cs_new_nopar:Nn \xeCJK_set_special_punct:nn
931 {
932 \tl_set:cx { l_xeCJK_#1_punct_tl } {#2}
933 \prop_clear:c { l_xeCJK_#1_punct_prop }
934 \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
935 { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
936 }
937 \cs_new_nopar:Nn \xeCJK_add_special_punct:nn
938 {
939 \tl_put_right:cx { l_xeCJK_#1_punct_tl } {#2}
940 \tl_map_inline:cn { l_xeCJK_#1_punct_tl }
941 { \prop_put:cnn { l_xeCJK_#1_punct_prop } {##1} { } }
942 }
943 \cs_new_nopar:Nn \xeCJK_sub_special_punct:nn
944 {
945 \tl_map_inline:xn {#2}
946 {
947 \tl_remove_all:cn { l_xeCJK_#1_punct_tl } {##1}
948 \prop_del:cn { l_xeCJK_#1_punct_prop } {##1}
949 }
950 }
951 \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 952 \cs_generate_variant:Nn \dim_set:Nn { Nc }
\dim_set_min:Nc 953 \cs_generate_variant:Nn \dim_add:Nn { Nc }
954 \cs_generate_variant:Nn \dim_set_min:Nn { Nc }
(End definition for \dim_set:Nc, \dim_add:Nc, and \dim_set_min:Nc.)

```

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

```

955 \tl_set:Nn \l_xeCJK_punct_coors_tl { \l_xeCJK_font_coors_tl/\l_xeCJK_punctstyle_tl }
(End definition for \l_xeCJK_punct_coors_tl.)

```

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

```

956 \cs_new_nopar:Nn \xeCJK_get_punct_bounds:nN
957 {
958 \tl_if_exist:cF { \l_xeCJK_punct_coors_tl/rule/#1/#2 }
959 {
960 \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
961 {
962 \clist_map_inline:nn { glue , rule , bound }
963 { \tl_gset:cV { \l_xeCJK_punct_coors_tl/##1/#1/#2 } \c_zero_dim }
964 \tl_gset:cV { \l_xeCJK_punct_coors_tl/rule/m/#2 } \c_zero_dim
965 }
966 {
967 \tl_if_exist:cF { \l_xeCJK_punct_coors_tl/bound/#1/#2 }
968 { { \xeCJK_select_font: \xeCJK_get_punct_dimen:N {#2} } }

```

```

969 \xeCJK_punct_if_long:NTF {#2}
970 {
971   \dim_zero:N \l_tmpa_dim
972   \dim_zero:N \l_tmpb_dim
973   \dim_zero:N \l_tmpc_dim
974 }
975 {
976   \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coor_tl/bound/#1/#2 }
977   \dim_set:Nc \l_tmpb_dim
978     { \l_xeCJK_punct_coor_tl/bound/\str_if_eq:xxTF {#1} 1r1/#2 }
979   \dim_set:Nn \l_tmpc_dim
980     {
981       \cs_if_exist_use:cF { g_xeCJK_punct_width/#2 }
982       {
983         \tl_if_blank:xTF \g_xeCJK_punct_width_tl
984           { \c_zero_dim }
985           { \g_xeCJK_punct_width_tl }
986       }
987     }
988 \xeCJK_punct_if_middle:NTF {#2}
989 {
990   \dim_set:Nn \l_tmpc_dim
991     {
992       \dim_compare:nNnTF \l_tmpc_dim > \c_zero_dim
993         { \l_tmpc_dim }
994         {
995           \prg_case_tl:Nnn \l_xeCJK_punctstyle_tl
996             {
997               \c_xeCJK_ps_halfwidth_tl { .5 em }
998               \c_xeCJK_ps_mixedwidth_tl { .5 em }
999               \c_xeCJK_ps_CCT_tl { .7 em }
1000               \c_xeCJK_ps_fullwidth_tl { 1 em }
1001             }
1002             { 1 em }
1003         }
1004       - \tl_use:c { \l_xeCJK_punct_coor_tl/dimen/#2 }
1005     }
1006   \dim_set:Nn \l_tmpc_dim { .5 \l_tmpc_dim }
1007 }
1008 {
1009   \dim_compare:nNnTF \l_tmpc_dim > \c_zero_dim
1010     {
1011       \dim_set:Nn \l_tmpc_dim
1012         {
1013           \l_tmpc_dim - \l_tmpb_dim
1014           - \tl_use:c { \l_xeCJK_punct_coor_tl/dimen/#2 }
1015         }
1016     }
1017   {
1018     \dim_set_eq:NN \l_tmpc_dim \l_tmpa_dim
1019     \dim_set_min:Nn \l_tmpc_dim \l_tmpb_dim
1020     \prg_case_tl:Nnn \l_xeCJK_punctstyle_tl

```

```

1021         {
1022             \c_xeCJK_ps_halfwidth_tl { \prg_do_nothing: }
1023             \c_xeCJK_ps_mixedwidth_tl
1024             {
1025                 \xeCJK_punct_if_mixedwidth:NT {#2}
1026                 {
1027                     \dim_add:Nn \l_tmpc_dim { .5 \l_tmpa_dim }
1028                     \dim_set_min:Nn \l_tmpc_dim \l_tmpa_dim
1029                 }
1030             }
1031             \c_xeCJK_ps_CCT_tl
1032             {
1033                 \xeCJK_punct_if_mixedwidth:NTF {#2}
1034                 { \dim_add:Nn \l_tmpc_dim { .5 \l_tmpa_dim } }
1035                 { \dim_add:Nn \l_tmpc_dim { .3 \l_tmpa_dim } }
1036                 \dim_set_min:Nn \l_tmpc_dim \l_tmpa_dim
1037             }
1038         }
1039         { \dim_set_eq:NN \l_tmpc_dim \l_tmpa_dim }
1040     }
1041 }
1042 \dim_set_max:Nn \l_tmpc_dim \c_zero_dim
1043 }
1044 \tl_gset:cx { \l_xeCJK_punct_coor_tl/rule/#1/#2 }
1045 { - \dim_use:N \l_tmpa_dim }
1046 \tl_gset:cx { \l_xeCJK_punct_coor_tl/rule/m/#2 }
1047 { - \dim_use:N \l_tmpb_dim }
1048 \tl_gset:cV { \l_xeCJK_punct_coor_tl/glue/#1/#2 } \l_tmpc_dim
1049 }
1050 }
1051 }
1052 \cs_generate_variant:Nn \xeCJK_get_punct_bounds:nN { nV }
(End definition for \xeCJK_get_punct_bounds:nN.)

```

\xeCJKsetwidth

```

1053 \NewDocumentCommand \xeCJKsetwidth { m m }
1054 { \tl_map_inline:xn {#1} { \tl_gset:cn { g_xeCJK_punct_width/##1 } {#2} } }
(End definition for \xeCJKsetwidth. This function is documented on page 10.)

```

```

\xeCJK_punct_if_right_p:N
\xeCJK_punct_if_both_left_right_p:NN 1055 \prg_new_conditional:Nnn \xeCJK_punct_if_right:N { p , T , F , TF }
\xeCJK_punct_if_long_p:N 1056 {
\xeCJK_punct_if_middle_p:N 1057 \tl_if_exist:cTF { \l_xeCJK_punct_coor_tl/glue/r/#1 }
\xeCJK_punct_if_mixedwidth_p:N 1058 \prg_return_true: \prg_return_false:
\xeCJK_punct_if_right:NTF 1059 }
\xeCJK_punct_if_both_left_right:NNTF 1060 \prg_new_conditional:Nnn \xeCJK_punct_if_both_left_right:NN { p , T , F , TF }
\xeCJK_punct_if_long:NTF 1061 {
\xeCJK_punct_if_middle:NTF 1062 \bool_if:nTF
\xeCJK_punct_if_mixedwidth:NTF 1063 {
1064 ( \tl_if_exist_p:c { \l_xeCJK_punct_coor_tl/glue/r/#1 } &&
1065 \tl_if_exist_p:c { \l_xeCJK_punct_coor_tl/glue/r/#2 } ) ||
1066 ( \tl_if_exist_p:c { \l_xeCJK_punct_coor_tl/glue/l/#1 } &&

```

```

1067         \tl_if_exist_p:c { \l_xeCJK_punct_coord_tl/glue/1/#2 } )
1068     }
1069     \prg_return_true: \prg_return_false:
1070 }
1071 \cs_generate_variant:Nn \prg_new_conditional:Nnn { c }
1072 \clist_map_inline:Nn \g_xeCJK_special_ps_clist
1073 {
1074     \prg_new_conditional:cnn { xeCJK_punct_if_#1:N } { p , T , F , TF }
1075     {
1076         \prop_if_in:cnTF { l_xeCJK_#1_punct_prop } {##1}
1077         \prg_return_true: \prg_return_false:
1078     }
1079 }

```

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

\xeCJK_get_kern:NN 标点挤压。行末半角方式,相邻标点拥挤;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。

```

1080 \cs_new_nopar:Nn \xeCJK_get_kern:NN
1081 {
1082     \tl_if_exist:cF { \l_xeCJK_punct_coord_tl/kern/#1-#2 }
1083     {
1084         \dim_zero:N \l_tmpa_dim
1085         \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_plain_tl
1086         {
1087             \tl_if_exist:cTF { g_xeCJK_punct/kern/#1-#2 }
1088             { \dim_set:Nc \l_tmpa_dim { g_xeCJK_punct/kern/#1-#2 } }
1089             { \xeCJK_calc_kern:NN {#1} {#2} }
1090         }
1091         \tl_gset:cV { \l_xeCJK_punct_coord_tl/kern/#1-#2 } \l_tmpa_dim
1092     }
1093 }
1094 \cs_new_nopar:Nn \xeCJK_calc_kern:NN
1095 {
1096     \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_marginkerning_tl
1097     {
1098         \xeCJK_punct_if_right:NT {#1}
1099         { \dim_add:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/r/#1 } }
1100         \xeCJK_punct_if_right:NF {#2}
1101         { \dim_add:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#2 } }
1102     }
1103     {
1104         \tl_if_eq:NNTF \l_xeCJK_punctstyle_tl \c_xeCJK_ps_fullwidth_tl
1105         { \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1.5 em } {#1} {#2} }
1106         {
1107             \xeCJK_punct_if_both_left_right:NNTF {#1} {#2}
1108             {
1109                 \xeCJK_punct_if_mixedwidth:NT {#1}
1110                 {
1111                     \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1 em } {#1} {#2}
1112                     \dim_compare:nNtT \l_tmpa_dim < { .1 em }
1113                     { \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coord_tl/bound/l/#1 } }
1114                 }

```

```

1115     }
1116     {
1117         \bool_if:nTF
1118         {
1119             \xeCJK_punct_if_mixedwidth_p:N {#1} &&
1120             ! ( \tl_if_eq_p:NN \l_xeCJK_punctstyle_tl \c_xeCJK_ps_halfwidth_tl )
1121         }
1122         {
1123             \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coors_tl/glue/r/#1 }
1124             \dim_set:Nn \l_tmpa_dim { .7 \l_tmpa_dim }
1125         }
1126         {
1127             \xeCJK_calc_kern:NnNN \l_tmpa_dim { 1 em } {#1} {#2}
1128             \dim_compare:nNnT \l_tmpa_dim < { .1 em }
1129             {
1130                 \dim_set:Nc \l_tmpa_dim { \l_xeCJK_punct_coors_tl/bound/l/#1 }
1131                 \dim_set_min:Nc \l_tmpa_dim { \l_xeCJK_punct_coors_tl/bound/r/#1 }
1132                 \dim_set:Nc \l_tmpb_dim { \l_xeCJK_punct_coors_tl/bound/l/#2 }
1133                 \dim_set_min:Nc \l_tmpb_dim { \l_xeCJK_punct_coors_tl/bound/r/#2 }
1134                 \dim_set_max:Nn \l_tmpa_dim \l_tmpb_dim
1135             }
1136         }
1137     }
1138 }
1139 \bool_if:nT
1140 { \xeCJK_punct_if_long_p:N {#1} || \xeCJK_punct_if_long_p:N {#2} }
1141 { \dim_set_max:Nn \l_tmpa_dim { .1 em } }
1142 }
1143 \dim_set_max:Nn \l_tmpa_dim \c_zero_dim
1144 }
1145 \cs_generate_variant:Nn \xeCJK_get_kern:NN { V }
1146 (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。

```

1146 \cs_new_nopar:Nn \xeCJK_calc_kern:NnNN
1147 {
1148     \dim_set:Nn #1
1149     {
1150         #2
1151         - \tl_use:c { \l_xeCJK_punct_coors_tl/
1152             \xeCJK_punct_if_right:NTF {#3} { bound } { glue } /l/#3 }
1153         - \tl_use:c { \l_xeCJK_punct_coors_tl/
1154             \xeCJK_punct_if_right:NTF {#4} { glue } { bound } /r/#4 }
1155         - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/#3 }
1156         - \tl_use:c { \l_xeCJK_punct_coors_tl/dimen/#4 }
1157         \xeCJK_punct_if_both_left_right:NNT {#3} {#4}
1158         { + \tl_use:c { \l_xeCJK_punct_coors_tl/rule/m/#4 } }
1159     }
1160 }
1161 (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。

```

1161 \cs_new_nopar:Nn \xeCJK_get_punct_dimen:N
1162 {
1163   \xeCJK_get_glyph_bounds:nNN { left } {#1} \l_tmpa_dim
1164   \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/l/#1 } \l_tmpa_dim
1165   \dim_set_eq:NN \l_tmpb_dim \l_tmpa_dim
1166   \xeCJK_get_glyph_bounds:nNN { right } {#1} \l_tmpa_dim
1167   \tl_gset:cV { \l_xeCJK_punct_coord_tl/bound/r/#1 } \l_tmpa_dim
1168   \dim_add:Nn \l_tmpb_dim \l_tmpa_dim
1169   \tl_gset:cx { \l_xeCJK_punct_coord_tl/dimen/#1 }
1170   { \dim_eval:n { \tex_fontcharwd:D \tex_font:D `#1 - \l_tmpb_dim } }
1171   \bool_if:nT
1172   {
1173     \xeCJK_punct_if_long_p:N {#1} &&
1174     ! ( \str_if_eq_p:xx {#1} { \cdots } || \str_if_eq_p:xx {#1} { \dots } )
1175   }
1176   {
1177     \clist_map_inline:Nn \g_xeCJK_punctstyle_clist
1178     {
1179       \str_if_eq:xxF \c_xeCJK_ps_plain_tl {##1}
1180       {
1181         \tl_gset:cx { \l_xeCJK_font_coord_tl/##1/kern/#1-#1 }
1182         { - \dim_use:N \l_tmpb_dim }
1183       }
1184     }
1185   }
1186 }

```

(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。

```

1187 \cs_new_nopar:Nn \xeCJK_get_glyph_bounds:nNN
1188 {
1189   \dim_set:Nn #3
1190   {
1191     \XeTeXglyphbounds
1192     \prg_case_str:xxn {#1}
1193     {
1194       { left } { \c_one } { top } { \c_two }
1195       { right } { \c_three } { bottom } { \c_four }
1196     }
1197     { \c_three }
1198     \int_eval:n { \XeTeXcharglyph `#2 }
1199   }
1200 }

```

(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。

```

1201 \NewDocumentCommand \xeCJKsetkern { m m m }
1202 { \tl_gset:cn { g_xeCJK_punct/kern/#1-#2 } {#3} }

```

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

5.6 后备字体

AutoFallback 后备字体的宏包选项声明。

```

1203 \keys_define:nn { xeCJK / options }
1204 {
1205   AutoFallback .choice:,
1206   AutoFallback / true .code:n = { \xeCJKenablefallback },
1207   AutoFallback / false .code:n = { \xeCJKdisablefallback },
1208   AutoFallback .default:n = { true },
1209   fallback .meta:n = { AutoFallback = true },
1210 }

```

(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

```

1211 \bool_new:N \l_xeCJK_fallback_bool
1212 \NewDocumentCommand \xeCJKenablefallback { }
1213 {
1214   \bool_if:NF \l_xeCJK_fallback_bool
1215   {
1216     \bool_set_true:N \l_xeCJK_fallback_bool
1217     \cs_set_eq:NN \xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1218     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_testsymbol:N
1219   }
1220 }
1221 \NewDocumentCommand \xeCJKdisablefallback { }
1222 {
1223   \bool_if:NT \l_xeCJK_fallback_bool
1224   {
1225     \bool_set_false:N \l_xeCJK_fallback_bool
1226     \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_save_CJKsymbol:N
1227   }
1228 }

```

(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。

```

1229 \cs_new_nopar:Nn \xeCJK_fallback_testsymbol:N
1230 {
1231   \font_glyph_if_exist:NnTF \tex_font:D {`#1}
1232   { \xeCJK_fallback_save_CJKsymbol:N {#1} }
1233   {
1234     \xeCJK_family_if_exist:nTF { \xeCJK@family/FallBack }
1235     { {
1236       \tl_put_right:Nn \xeCJK@family { /FallBack }
1237       \xeCJK_select_font:
1238       \xeCJK_fallback_testsymbol:N {#1}
1239     } }
1240     {
1241       \xeCJK_warning:nx { fallback } {#1}
1242       \xeCJK_fallback_save_CJKsymbol:N {#1}
1243     }
1244   }
1245 }

```

```

1246 \xeCJK_msg_new:nn { fallback }
1247 {
1248   CJKfamily~'\xeCJK@family'~
1249   ( \prop_get:NV \g_xeCJK_family_prop \xeCJK@family )~
1250   does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {\`#1}).\\
1251 }
(End definition for \xeCJK_fallback_testsymbol:N.)

```

\setCJKfallbackfamilyfont

```

\setCJK_set_family_fallback:nnn 1252 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1253 {
1254   \exp_args:Nx \tl_if_in:nnTF {#3} { , }
1255   { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} }
1256   { \xeCJK_set_family:nnn { #1/FallBack } {#2} {#3} }
1257 }
1258 \cs_new_nopar:Nn \xeCJK_set_family_fallback:nnn
1259 {
1260   \group_begin:
1261   \tl_set:Nx \l_xeCJK_family_fb_tl {#1}
1262   \prop_get:NVNF \g_xeCJK_family_prop \l_xeCJK_family_fb_tl \l_xeCJK_fontname_tl
1263   { \tl_clear:N \l_xeCJK_fontname_tl }
1264   \clist_map_inline:xn {#3}
1265   {
1266     \tl_put_right:Nn \l_xeCJK_family_fb_tl { /FallBack }
1267     \xeCJK_get_sub_features:nn \l_xeCJK_family_fb_tl {##1}
1268     \clist_put_left:cx { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist } {#2}
1269     \xeCJK_set_family:ncc \l_xeCJK_family_fb_tl
1270     { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontfeat_clist }
1271     { \l_xeCJK_ \l_xeCJK_family_fb_tl _fontname_tl }
1272   }
1273   \group_end:
1274 }

```

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

5.7 CJK 字体族声明方式

```

1275 \bool_new:N \g_xeCJK_AutoFakeBold_bool
1276 \bool_new:N \g_xeCJK_AutoFakeSlant_bool
1277 \fp_new:N \g_xeCJK_EmboldenFactor_fp
1278 \fp_new:N \g_xeCJK_SlantFactor_fp

```

AutoFakeBold 伪粗体和伪斜体的宏包选项声明。

```

AutoFakeSlant 1279 \keys_define:nn { xeCJK / options }
EmboldenFactor 1280 {
SlantFactor 1281   AutoFakeBold .choice:,
1282   AutoFakeBold / true .code:n =
1283   { \bool_set_true:N \g_xeCJK_AutoFakeBold_bool },
1284   AutoFakeBold / false .code:n =
1285   { \bool_set_false:N \g_xeCJK_AutoFakeBold_bool },
1286   AutoFakeBold / unknown .code:n =
1287   {
1288     \bool_set_true:N \g_xeCJK_AutoFakeBold_bool

```

```

1289     \fp_set:Nn \g_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1290   },
1291   AutoFakeBold .default:n = { true },
1292   AutoFakeSlant .choice:,
1293   AutoFakeSlant / true .code:n =
1294     { \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool },
1295   AutoFakeSlant / false .code:n =
1296     { \bool_set_false:N \g_xeCJK_AutoFakeSlant_bool },
1297   AutoFakeSlant / unknown .code:n =
1298     {
1299       \bool_set_true:N \g_xeCJK_AutoFakeSlant_bool
1300       \fp_set:Nn \g_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1301     },
1302   AutoFakeSlant .default:n = { true },
1303   EmboldenFactor .fp_set:N = \g_xeCJK_EmboldenFactor_fp,
1304   SlantFactor .fp_set:N = \g_xeCJK_SlantFactor_fp,
1305   BoldFont .meta:n = { AutoFakeBold = true },
1306   boldfont .meta:n = { AutoFakeBold = true },
1307   SlantFont .meta:n = { AutoFakeSlant = true },
1308   slantfont .meta:n = { AutoFakeSlant = true },
1309 }

```

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

\keys_set_known:nxN

```

1310 \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 1311 \clist_new:N \g_xeCJK_sub_key_clist
1312 \cs_new_nopar:Nn \xeCJK_new_sub_key:n
1313 {
1314   \clist_gput_right:Nx \g_xeCJK_sub_key_clist {#1}
1315   \keys_define:nn { xeCJK / features }
1316   {
1317     #1 .code:n =
1318     {
1319       \tl_if_blank:xTF {##1}
1320       {
1321         \bool_set_false:c { l_xeCJK_copy_#1_bool }
1322         \bool_set_false:c { l_xeCJK_add_#1_bool }
1323         \tl_put_right:Nx \l_xeCJK_family_tl { /#1 }
1324       }
1325       {
1326         \clist_put_right:Nx \l_xeCJK_sub_key_clist {#1}
1327         \str_if_eq:xxTF {##1} *
1328         {
1329           \bool_set_true:c { l_xeCJK_copy_#1_bool }
1330           \bool_set_false:c { l_xeCJK_add_#1_bool }
1331         }
1332         {
1333           \bool_set_false:c { l_xeCJK_copy_#1_bool }
1334           \bool_set_true:c { l_xeCJK_add_#1_bool }

```

```

1335         \xeCJK_get_sub_features:nn {#1} {##1}
1336     }
1337 }
1338 },
1339 #1 .default:n = \c_empty_tl,
1340 }
1341 }
1342 \cs_generate_variant:Nn \xeCJK_new_sub_key:n { x }
1343 \cs_new_nopar:Nn \xeCJK_get_sub_features:nn
1344 {
1345     \tl_set:Nx \l_tmpa_tl {#2}
1346     \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1347     \xeCJK_gobble_brace:N \l_tmpa_tl
1348     \exp_args:No \tl_if_head_eq_charcode:nNTF \l_tmpa_tl [ % ]
1349     {
1350         \exp_after:wN \xeCJK_get_sub_features_aux:n \l_tmpa_tl \c_empty_tl
1351         \tl_if_blank:xT \l_xeCJK_sub_fontname_tl
1352         {
1353             \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_tmpa_tl
1354             \clist_clear:N \l_xeCJK_sub_fontfeat_clist
1355         }
1356     }
1357     { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_tmpa_tl }
1358     \tl_if_blank:xTF \l_xeCJK_sub_fontname_tl
1359     { \tl_set:Nx \l_xeCJK_sub_fontname_tl \l_xeCJK_fontname_tl }
1360     { \tl_replace_all:Nnx \l_xeCJK_sub_fontname_tl * \l_xeCJK_fontname_tl }
1361     \clist_set:cx { \l_xeCJK_#1_fontfeat_clist } \l_xeCJK_sub_fontfeat_clist
1362     \tl_set:cx { \l_xeCJK_#1_fontname_tl } \l_xeCJK_sub_fontname_tl
1363 }
(End definition for \xeCJK_new_sub_key:n and \g_xeCJK_sub_key_clist.)

\xeCJK_gobble_brace:N
\xeCJK_get_sub_features:nn 1364 \cs_new_nopar:Nn \xeCJK_gobble_brace:N
\xeCJK_get_sub_features_aux:n 1365 {
1366     \exp_args:No \tl_if_head_group:nT #1
1367     {
1368         \exp_last_unbraced:NNo \tl_set:Nn #1 #1
1369         \xeCJK_gobble_brace:N #1
1370     }
1371 }
1372 \cs_new_nopar:Npn \xeCJK_get_sub_features_aux:n [#1] #2
1373 {
1374     \clist_set:Nx \l_xeCJK_sub_fontfeat_clist {#1}
1375     \tl_set:Nx \l_xeCJK_sub_fontname_tl {#2}
1376 }
(End definition for \xeCJK_gobble_brace:N, \xeCJK_get_sub_features:nn, and \xeCJK_get_sub_features_aux:n.)

Fallback
1377 \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 1378 \cs_set_protected:cpn { \c_keys_props_root_tl .clist_set_x:N } #1
1379 { \keys_variable_set:NnN #1 { clist } x }
1380 \cs_set_protected:cpn { \c_keys_props_root_tl .clist_set_x:c } #1
1381 { \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 1382 \cs_new_nopar:Nn \xeCJK_map_features_id:n
    ItalicFont 1383 { \prop_get:Nn \g_xeCJK_features_id_prop { #1 } }
    BoldItalicFont 1384 \prop_new:N \g_xeCJK_features_id_prop
    SlantedFont 1385 \prop_put:Nnn \g_xeCJK_features_id_prop { bf } { Bold }
    BoldSlantedFont 1386 \prop_put:Nnn \g_xeCJK_features_id_prop { it } { Italic }
    BoldFeatures 1387 \prop_put:Nnn \g_xeCJK_features_id_prop { bfit } { BoldItalic }
    ItalicFeatures 1388 \prop_put:Nnn \g_xeCJK_features_id_prop { sl } { Slanted }
    BoldItalicFeatures 1389 \prop_put:Nnn \g_xeCJK_features_id_prop { bfs1 } { BoldSlanted }
    SlantedFeatures 1390 \prop_map_inline:Nn \g_xeCJK_features_id_prop
    BoldSlantedFeatures 1391 {
1392     \keys_define:nn { xeCJK / features }
1393     {
1394         #2Font .tl_set_x:c = { l_xeCJK_fontname_#1_tl },
1395         #2Features .clist_set_x:c = { l_xeCJK_fontfeat_#1_clist } ,
1396     }
1397 }
(End definition for \g_xeCJK_features_id_prop and others.)

AutoFakeBold
AutoFakeSlant 1398 \keys_define:nn { xeCJK / features }
Mono 1399 {
1400     AutoFakeBold .choice:,
1401     AutoFakeBold / false .code:n =
1402     { \bool_set_false:N \l_xeCJK_AutoFakeBold_bool },
1403     AutoFakeBold / unknown .code:n =
1404     {
1405         \bool_set_true:N \l_xeCJK_AutoFakeBold_bool
1406         \fp_set:Nn \l_xeCJK_EmboldenFactor_fp { \l_keys_value_tl }
1407     },
1408     AutoFakeBold .default:n = { \g_xeCJK_EmboldenFactor_fp },
1409     AutoFakeSlant .choice:,
1410     AutoFakeSlant / false .code:n =
1411     { \bool_set_false:N \l_xeCJK_AutoFakeSlant_bool },
1412     AutoFakeSlant / unknown .code:n =
1413     {
1414         \bool_set_true:N \l_xeCJK_AutoFakeSlant_bool
1415         \fp_set:Nn \l_xeCJK_SlantFactor_fp { \l_keys_value_tl }
1416     },
1417     AutoFakeSlant .default:n = { \g_xeCJK_SlantFactor_fp },
1418     Mono .choice:,
1419     Mono / Exspace .code:n =
1420     {
1421         \xeCJK_set_monoexspace:
1422         \tl_clear:N \l_xeCJK_monoscale_tl

```

```

1423     },
1424     Mono / Scale .code:n =
1425     {
1426         \xeCJK_set_monoscale:
1427         \tl_set:Nx \l_xeCJK_monoscale_tl { Scale = { \fp_use:N \g_xeCJK_monoscale_fp } }
1428     }
1429 }

```

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

\xeCJK_set_init:

```

1430 \cs_new_nopar:Nn \xeCJK_set_init:
1431 {
1432     \tl_clear:N \l_xeCJK_monoscale_tl
1433     \int_gincr:N \g_xeCJK_fontspec_int
1434     \prop_clear:N \l_xeCJK_add_font_prop
1435     \prop_map_inline:Nn \g_xeCJK_features_id_prop
1436     {
1437         \tl_clear:c { l_xeCJK_fontname_##1_tl }
1438         \clist_clear:c { l_xeCJK_fontfeat_##1_clist }
1439     }
1440     \clist_clear:N \l_xeCJK_pass_features_clist
1441     \clist_clear:N \l_xeCJK_sub_key_clist
1442     \clist_map_inline:Nn \g_xeCJK_sub_key_clist
1443     {
1444         \bool_set_false:c { l_xeCJK_copy_##1_bool }
1445         \bool_set_false:c { l_xeCJK_add_##1_bool }
1446     }
1447     \bool_set_eq:NN \l_xeCJK_AutoFakeBold_bool \g_xeCJK_AutoFakeBold_bool
1448     \bool_set_eq:NN \l_xeCJK_AutoFakeSlant_bool \g_xeCJK_AutoFakeSlant_bool
1449     \fp_set_eq:NN \l_xeCJK_EmboldenFactor_fp \g_xeCJK_EmboldenFactor_fp
1450     \fp_set_eq:NN \l_xeCJK_SlantFactor_fp \g_xeCJK_SlantFactor_fp
1451 }

```

(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。

```

1452 \cs_new_protected_nopar:Nn \xeCJK_set_family:nnn
1453 {
1454     \group_begin:
1455     \xeCJK_set_init:
1456     \tl_set:Nx \l_xeCJK_family_tl {#1}
1457     \clist_set:Nx \l_xeCJK_fontoptions_clist {#2}
1458     \tl_set:Nx \l_xeCJK_fontname_tl {#3}
1459     \keys_set_known:nxN { xeCJK / features }
1460     { \g_xeCJK_default_features_clist, #2 } \l_xeCJK_pass_features_clist
1461     \xeCJK_parse_features:
1462     \xeCJK_pass_features:
1463     \xeCJK_check_family:V \l_xeCJK_family_tl
1464     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1465     {
1466         \exp_not:n { \fontspec_set_family:Nnn \l_xeCJK_fontspec_family_tl }
1467         { \l_xeCJK_pass_features_clist } { \l_xeCJK_fontname_tl }
1468         \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop

```

```

1469         { \l_xeCJK_family_tl } \exp_not:N \l_xeCJK_fontspec_family_tl
1470     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_family_tl }
1471     \exp_not:n
1472     { {
1473         \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1474         \exp_not:N \fontfamily { \l_xeCJK_fontspec_family_tl }
1475         \exp_not:N \selectfont
1476     } }
1477 }
1478 \xeCJK_save_family_info:
1479 \xeCJK_add_sub_block:
1480 \group_end:
1481 }
1482 \cs_generate_variant:Nn \xeCJK_set_family:nnn { ncc }
    (End definition for \xeCJK_set_family:nnn.)

```

\xeCJK_check_family:n

```

1483 \cs_new_nopar:Nn \xeCJK_check_family:n
1484 {
1485     \prop_get:NnNT \g_xeCJK_family_prop {#1} \l_tmpa_tl
1486     {
1487         \prop_gpop:NnNT \g_xeCJK_family_name_prop {#1} \l_tmpa_tl { }
1488         \xeCJK_warning:nxx { CJKfamily-redef } {#1} \l_tmpa_tl
1489     }
1490 }
1491 \cs_generate_variant:Nn \xeCJK_check_family:n { V }
1492 \xeCJK_msg_new:nn { CJKfamily-redef } { Redefining~CJKfamily~`#1'~(#2). }
    (End definition for \xeCJK_check_family:n.)

```

\xeCJK_add_font:nn

\xeCJK_add_font_if_new:nn

```

1493 \cs_new_nopar:Nn \xeCJK_add_font:nn
1494 {
1495     \prop_put:Nxx \l_xeCJK_add_font_prop
1496     { \xeCJK_map_features_id:n {#1} Font } {#2}
1497 }
1498 \cs_new_nopar:Nn \xeCJK_add_font_if_new:nn
1499 {
1500     \prop_put_if_new:Nxx \l_xeCJK_add_font_prop
1501     { \xeCJK_map_features_id:n {#1} Font } {#2}
1502 }
1503 \cs_generate_variant:Nn \xeCJK_add_font:nn { nx, nc, nV, nv }
1504 \cs_generate_variant:Nn \xeCJK_add_font_if_new:nn { nx, nc, nV, nv }
    (End definition for \xeCJK_add_font:nn and \xeCJK_add_font_if_new:nn.)

1505 \cs_generate_variant:Nn \prop_put:Nnn { Nxx }
1506 \cs_generate_variant:Nn \prop_get:Nn { NV }
1507 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
1508 \cs_generate_variant:Nn \prop_if_in:NnF { Nx }
1509 \cs_generate_variant:Nn \prop_put_if_new:Nnn { Nxx }

```

\xeCJK_add_fake_bold:n

\xeCJK_add_fake_slant:n

```

1510 \cs_new_nopar:Nn \xeCJK_add_fake_bold:n
1511 {

```

```

1512     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1513     { FakeBold = { \fp_use:N \l_xeCJK_EmboldenFactor_fp } }
1514 }
1515 \cs_new_nopar:Nn \xeCJK_add_fake_slant:n
1516 {
1517     \clist_put_left:cx { l_xeCJK_fontfeat_#1_clist }
1518     { FakeSlant = { \fp_use:N \l_xeCJK_SlantFactor_fp } }
1519 }
(End definition for \xeCJK_add_fake_bold:n and \xeCJK_add_fake_slant:n.)

```

\xeCJK_parse_features:

```

\xeCJK_parse_features:n 1520 \cs_new_nopar:Nn \xeCJK_parse_features:
1521 { \prop_map_inline:Nn \g_xeCJK_features_id_prop { \xeCJK_parse_features:n {##1} } }
1522 \cs_new_nopar:Nn \xeCJK_parse_features:n
1523 {
1524     \xeCJK_if_font_select:nTF {#1}
1525     {
1526         \xeCJK_add_font:nv {#1} { l_xeCJK_fontname_#1_tl }
1527         \xeCJK_if_it_or_sl:nTF {#1}
1528         {
1529             \xeCJK_if_font_select:nF {bf#1}
1530             { \xeCJK_add_font_if_new:nv {bf#1} { l_xeCJK_fontname_#1_tl } }
1531         }
1532         {
1533             \str_if_eq:xxT {#1} {bf}
1534             {
1535                 \clist_map_inline:nn { it , sl }
1536                 {
1537                     \xeCJK_if_font_select:nF {bf##1}
1538                     { \xeCJK_add_font:nv {bf##1} \l_xeCJK_fontname_bf_tl }
1539                 }
1540             }
1541         }
1542     }
1543     { \xeCJK_set_fake:n {#1} }
1544 }
(End definition for \xeCJK_parse_features: and \xeCJK_parse_features:n.)

```

\xeCJK_if_font_select_p:n

```

\xeCJK_if_it_or_sl_p:n 1545 \prg_new_conditional:Nnn \xeCJK_if_font_select:n { p, T, F, TF }
\xeCJK_if_font_select:nTF 1546 {
\xeCJK_if_it_or_sl:nTF 1547     \exp_args:Nc
1548     \tl_if_blank:xTF { l_xeCJK_fontname_#1_tl } \prg_return_false: \prg_return_true:
1549 }
1550 \prg_new_conditional:Nnn \xeCJK_if_it_or_sl:n { p, T, F, TF }
1551 {
1552     \bool_if:nTF { \str_if_eq_p:xx { it } {#1} || \str_if_eq_p:xx { sl } {#1} }
1553     \prg_return_true: \prg_return_false:
1554 }
(End definition for \xeCJK_if_font_select:n and \xeCJK_if_it_or_sl:n.)

```


\xeCJK_set_fake:n

```
1555 \cs_new_nopar:Nn \xeCJK_set_fake:n
1556 {
1557   \str_if_eq:xxTF {#1} { bf }
1558   { \bool_if:NT \l_xeCJK_AutoFakeBold_bool { \xeCJK_add_fake_bold:n {#1} } }
1559   {
1560     \bool_if:NTF \l_xeCJK_AutoFakeSlant_bool
1561     {
1562       \bool_if:nT
1563       { \xeCJK_if_it_or_sl_p:n {#1} ||
1564         ( \str_if_eq_p:xx {#1} {bf} ) && ! ( \xeCJK_if_font_select_p:n {it} ) ) ||
1565         ( \str_if_eq_p:xx {#1} {bfs} ) && ! ( \xeCJK_if_font_select_p:n {sl} ) )
1566       }
1567       { \xeCJK_add_fake_slant:n {#1} }
1568     }
1569     { \xeCJK_if_it_or_sl:nT {#1} { \xeCJK_map_it_sl:n {#1} } }
1570     \bool_if:nT
1571     { \l_xeCJK_AutoFakeBold_bool
1572       && ! ( \xeCJK_if_it_or_sl_p:n {#1} )
1573       && ! ( \xeCJK_if_font_select_p:n {bf} )
1574     }
1575     { \xeCJK_add_fake_bold:n {#1} }
1576   }
1577   \xeCJK_add_font_if_new:nn {#1} *
1578 }
```

(End definition for \xeCJK_set_fake:n.)

\xeCJK_map_it_sl:n

```
1579 \cs_new_nopar:Nn \xeCJK_map_it_sl:n
1580 {
1581   \xeCJK_if_map_font_select:nT {#1}
1582   {
1583     \xeCJK_add_font:nx {#1} { \xeCJK_get_map_font:n {#1} }
1584     \xeCJK_if_font_select:nF {bf#1}
1585     { \xeCJK_add_font_if_new:nx {bf#1} { \xeCJK_get_map_font:n {#1} } }
1586   }
1587 }
```

(End definition for \xeCJK_map_it_sl:n.)

\xeCJK_get_map_font:n

```
1588 \cs_new_nopar:Nn \xeCJK_get_map_font:n
1589 { \tl_use:c { l_xeCJK_fontname_\str_if_eq:xxTF {#1} {it} {sl} {it} _tl } }
1590
```

(End definition for \xeCJK_get_map_font:n.)

\xeCJK_if_map_font_select_p:n

```
\xeCJK_if_map_font_select:nTF 1590 \prg_new_conditional:Nnn \xeCJK_if_map_font_select:n { p, T, F, TF }
1591 {
1592   \tl_if_blank:xxTF { \xeCJK_get_map_font:n {#1} }
1593   \prg_return_false: \prg_return_true:
1594 }
```

(End definition for \xeCJK_if_map_font_select:n.)

\xeCJK_pass_features:

```
1595 \cs_new_nopar:Nn \xeCJK_pass_features:
1596 {
1597   \prop_map_inline:Nn \g_xeCJK_features_id_prop
1598   {
1599     \clist_if_empty:cF { l_xeCJK_fontfeat_##1_clist }
1600     {
1601       \clist_put_right:Nx \l_xeCJK_pass_features_clist
1602       { ##2Features = { \clist_use:c { l_xeCJK_fontfeat_##1_clist } } }
1603     }
1604   }
1605   \prop_map_inline:Nn \l_xeCJK_add_font_prop
1606   { \clist_put_right:Nx \l_xeCJK_pass_features_clist { ##1 = { ##2 } } }
1607   \tl_if_blank:xF \l_xeCJK_monoscale_tl
1608   { \clist_put_right:Nx \l_xeCJK_pass_features_clist \l_xeCJK_monoscale_tl }
1609 }
(End definition for \xeCJK_pass_features:.)
```

\g_xeCJK_family_prop

```
\g_xeCJK_family_name_prop 1610 \prop_new:N \g_xeCJK_family_prop
\g_xeCJK_family_options_prop 1611 \prop_new:N \g_xeCJK_family_name_prop
1612 \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:

```
1613 \cs_new_nopar:Nn \xeCJK_save_family_info:
1614 {
1615   \group_begin:
1616   \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1617   {
1618     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist {##1}
1619     \clist_remove_all:Nn \l_xeCJK_fontoptions_clist { ##1 = }
1620   }
1621   \prop_gput:NVV \g_xeCJK_family_prop \l_xeCJK_family_tl \l_xeCJK_fontname_tl
1622   \prop_gput:NVV \g_xeCJK_family_options_prop \l_xeCJK_family_tl \l_xeCJK_fontoptions_clist
1623   \group_end:
1624 }
(End definition for \xeCJK_save_family_info:.)
```

\xeCJK_add_sub_block:

```
1625 \cs_new_nopar:Nn \xeCJK_add_sub_block:
1626 {
1627   \clist_remove_duplicates:N \l_xeCJK_sub_key_clist
1628   \clist_map_inline:Nn \l_xeCJK_sub_key_clist
1629   {
1630     \tl_set:Nx \l_xeCJK_sub_family_tl { \l_xeCJK_family_tl/##1 }
1631     \bool_if:cT { l_xeCJK_copy_##1_bool }
1632     {
1633       \xeCJK_check_family:V \l_xeCJK_sub_family_tl
1634       \prop_get:NVNT \g_xeCJK_family_prop \l_xeCJK_family_tl \l_tmpa_tl
1635       { \prop_gput:NVV \g_xeCJK_family_prop \l_xeCJK_sub_family_tl \l_tmpa_tl }
1636       \prop_get:NVNT \g_xeCJK_family_options_prop \l_xeCJK_family_tl \l_tmpa_clist
```

```

1637     {
1638         \clist_remove_all:Nn \l_tmpa_clist { ##1 = * }
1639     \prop_gput:NVV \g_xeCJK_family_options_prop \l_xeCJK_sub_family_tl \l_tmpa_clist
1640     }
1641     \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_sub_family_tl }
1642     {
1643         \xeCJK_family_if_exist:nT { \l_xeCJK_family_tl }
1644         {
1645             \prop_get:NnNT \exp_not:N \g_xeCJK_family_name_prop
1646             { \l_xeCJK_family_tl } \exp_not:N \l_tmpa_tl
1647             {
1648                 \prop_gput:NnV \exp_not:N \g_xeCJK_family_name_prop
1649                 { \l_xeCJK_sub_family_tl } \exp_not:N \l_tmpa_tl
1650             \cs_gset_protected_nopar:cpx { xeCJK/family/\l_xeCJK_sub_family_tl }
1651             \exp_not:n
1652             { {
1653                 \exp_not:N \fontencoding { \g_fontspec_encoding_tl }
1654                 \exp_not:N \fontfamily { \l_tmpa_tl }
1655                 \exp_not:N \selectfont
1656             } }
1657         }
1658     }
1659 }
1660 }
1661 \bool_if:cT { l_xeCJK_add_##1_bool }
1662 {
1663     \xeCJK_set_family:ncc \l_xeCJK_sub_family_tl
1664     { l_xeCJK_##1_fontfeat_clist } { l_xeCJK_##1_fontname_tl }
1665 }
1666 }
1667 }

```

(End definition for \xeCJK_add_sub_block:.)

\xeCJK_copy_family:nn

```

1668 \cs_new_nopar:Nn \xeCJK_copy_family:nn
1669 {
1670     \xeCJK_family_if_exist:nT {#2}
1671     {
1672         \tl_map_inline:nn
1673         { \g_xeCJK_family_prop \g_xeCJK_family_options_prop \g_xeCJK_family_name_prop }
1674         { \prop_get:NnNT ##1 {#2} \l_tmpa_tl { \prop_gput:NnV ##1 {#1} \l_tmpa_tl } }
1675         \cs_gset_eq:cc { xeCJK/family/#1 } { xeCJK/family/#2 }
1676     }
1677 }
1678 \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_coord_tl 缓存当前字体的原始格式,以加速编译。

```

\xeCJK_select_font: 1679 \tl_set:Nn \l_xeCJK_font_coord_tl { xeCJK/\xeCJK@family/\f@series/\f@shape/\f@size }
\xeCJK_switch_font:nn 1680 \cs_new_nopar:Nn \xeCJK_select_font:

```

```

1681 {
1682   \cs_if_exist_use:cF \l_xeCJK_font_coor_tl
1683   {
1684     \tl_set:Nx \l_xeCJK_current_coor_tl \l_xeCJK_font_coor_tl
1685     \cs_if_exist_use:cT { xeCJK/family/\xeCJK@family }
1686     { \exp_last_unbraced:NNV \cs_gset_eq:cN \l_xeCJK_current_coor_tl \font@name }
1687   }
1688 }
1689 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
1690 \cs_new_nopar:Nn \xeCJK_switch_font:nn
1691 {
1692   \str_if_eq:xxF {#1} {#2}
1693   {
1694     \xeCJK_info:nxx { CJK-block } {#1} {#2}
1695     \tl_remove_all:Nn \xeCJK@family { /#1 }
1696     \str_if_eq:xxF {#2} { CJK }
1697     {
1698       \tl_set:Nx \l_tmpa_tl \xeCJK@family
1699       \tl_put_right:Nx \xeCJK@family { /#2 }
1700       \xeCJK_family_if_exist:nF \xeCJK@family
1701       {
1702         \xeCJK_family_if_exist:nTF { \CJKfamilydefault/#2 }
1703         { \xeCJK_copy_family:xx \xeCJK@family { \CJKfamilydefault/#2 } }
1704         { \xeCJK_copy_family:xx \xeCJK@family \l_tmpa_tl }
1705       }
1706     }
1707   }
1708   \xeCJK_select_font:
1709 }
1710 \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

```

1711 \prg_new_protected_conditional:Nnn \xeCJK_family_if_exist:n { T, F, TF }
1712 {
1713   \cs_if_free:cTF { xeCJK/family/#1 } \prg_return_false:
1714   {
1715     \prop_if_in:NxF \g_xeCJK_family_name_prop {#1} { \use:c { xeCJK/family/#1 } }
1716     \prg_return_true:
1717   }
1718 }
(End definition for \xeCJK_family_if_exist:nTF.)

```

\CJKfamily 用于切换 CJK 字体族。

```

1719 \NewDocumentCommand \CJKfamily { m }
1720 {
1721   \xeCJK_family_if_exist:nTF {#1}
1722   { \tl_set:Nx \xeCJK@family {#1} }
1723   {
1724     \cs_if_free:cT { xeCJK_warned_aux_#1 }
1725     {
1726       \xeCJK_warning:nx { CJKfamily-Unknown } {#1}

```

```

1727         \tl_new:c { xeCJK_warned_aux_#1 }
1728     }
1729 }
1730 \tex_ignorespaces:D
1731 }
1732 \xeCJK_msg_new:nn { CJKfamily-Unknown }
1733 {
1734     Unknown~CJK~family~'#1'~is~ignored.\\\
1735     Try~to~use~
1736     \prg_case_str:xxn {#1}
1737     {
1738         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1739         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1740         \CJKttdefault { \token_to_str:N \setCJKmonofont }
1741     }
1742     { \token_to_str:N \setCJKfamilyfont {#1} }
1743     [...]{}~to~define~it.
1744 }

```

(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 1745 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
\CJKfontspec 1746 { \xeCJK_set_family:nnn {#1} {#2} {#3} }
1747 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
1748 {
1749     \IfNoValueTF {#1}
1750     { \tl_set:Nx \l_xeCJK_family_tl { \cs_to_str:N #2 } }
1751     { \tl_set:Nx \l_xeCJK_family_tl {#1} }
1752     \xeCJK_set_family:nnn \l_xeCJK_family_tl {#3} {#4}
1753     \cs_new_protected_nopar:Npx #2 { \exp_not:N \CJKfamily { \l_xeCJK_family_tl } }
1754 }
1755 \int_new:N \g_xeCJK_fontspec_int
1756 \int_gdecr:N \g_xeCJK_fontspec_int
1757 \NewDocumentCommand \CJKfontspec { O{} m }
1758 {
1759     \tl_set:Nx \l_xeCJK_CJKfontspec_id_tl { CJKfontspec/#1/#2/id }
1760     \cs_if_free:cT \l_xeCJK_CJKfontspec_id_tl
1761     {
1762         \int_gincr:N \g_xeCJK_fontspec_int
1763         \tl_gset:Nx \g_xeCJK_family_spec_tl
1764         { CJKfontspec ( \int_use:N \g_xeCJK_fontspec_int ) }
1765         \xeCJK_set_family:nnn \g_xeCJK_family_spec_tl {#1} {#2}
1766         \tl_gset:cx \l_xeCJK_CJKfontspec_id_tl \g_xeCJK_family_spec_tl
1767     }
1768     \exp_args:Nv \CJKfamily \l_xeCJK_CJKfontspec_id_tl
1769     \tex_ignorespaces:D
1770 }

```

(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 1771 \clist_new:N \g_xeCJK_default_features_clist

```

```

1772 \NewDocumentCommand \defaultCJKfontfeatures { m }
1773 { \clist_gset:Nn \g_xeCJK_default_features_clist {#1} }
1774 \@onlypreamble \defaultCJKfontfeatures
1775 \NewDocumentCommand \addCJKfontfeatures { m }
1776 {
1777   \prop_if_in:NVTF \g_xeCJK_family_prop \xeCJK@family
1778   {
1779     \group_begin:
1780     \clist_set:Nx \l_tmpa_clist {#1}
1781     \prop_get:NVN \g_xeCJK_family_prop \xeCJK@family \l_tmpa_tl
1782     \clist_map_inline:Nn \g_xeCJK_CJK_subclass_clist
1783     {
1784       \clist_if_in:NnT \l_tmpa_clist {##1}
1785       {
1786         \clist_remove_all:Nn \l_tmpa_clist {##1}
1787         \prop_get:NxNF \g_xeCJK_family_prop { \xeCJK@family/##1 } \l_tmpb_tl
1788         {
1789           \prop_get:NxNF \g_xeCJK_family_prop
1790             { \CJKfamilydefault/##1 } \l_tmpb_tl
1791           { \tl_set:Nx \l_tmpb_tl \l_tmpa_tl }
1792         }
1793         \clist_set:Nx \l_tmpa_clist
1794           { ##1 = { [ \l_tmpa_clist ] { \l_tmpb_tl } } }
1795         \clist_map_break:
1796       }
1797     }
1798     \prop_get:NVNF \g_xeCJK_family_options_prop \xeCJK@family \l_tmpb_clist
1799     { \clist_clear:N \l_tmpb_clist }
1800     \clist_clear:N \l_tmpc_clist
1801     \clist_map_inline:Nn \l_tmpb_clist
1802     {
1803       \clist_if_in:NnF \l_tmpa_clist {##1}
1804       { \clist_put_right:Nn \l_tmpc_clist {##1} }
1805     }
1806     \clist_put_left:NV \l_tmpa_clist \l_tmpc_clist
1807     \use:x
1808     { \group_end: \exp_not:N \CJKfontspec [ \l_tmpa_clist ] { \l_tmpa_tl } }
1809   }
1810   { \xeCJK_warning:n { addCJKfontfeature-ignored } }
1811   \tex_ignorespaces:D
1812 }
1813 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
1814 \xeCJK_msg_new:nn { addCJKfontfeature-ignored }
1815 {
1816   \tl_to_str:N \addCJKfontfeature (s)~ignored.\!\!\!
1817   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
1818 }

```

(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 1819 \NewDocumentCommand \setCJKmainfont { O{ } m }
1820 { \xeCJK_set_family:nnn \CJKrmdefault {#1} {#2} }

```

```

1821 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1822 \NewDocumentCommand \setCJKsansfont { O{ } m }
1823 { \xeCJK_set_family:nnn \CJKsfdefault {#1} {#2} }
(End definition for \setCJKmainfont and \setCJKsansfont. These functions are documented on page 5.)

```

```

\CJKrmdefault
\CJKsfdefault 1824 \cs_if_free:NT \CJKrmdefault { \tl_set:Nn \CJKrmdefault { rm } }
\CJKttdefault 1825 \cs_if_free:NT \CJKsfdefault { \tl_set:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 1826 \cs_if_free:NT \CJKttdefault { \tl_set:Nn \CJKttdefault { tt } }
1827 \cs_if_free:NT \CJKfamilydefault { \tl_set:Nn \CJKfamilydefault { \CJKrmdefault } }
1828 \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。

```

1829 \AtEndOfPackage
1830 {
1831   \tex_protected:D \tl_put_right:Nn \normalfont { \CJKfamily \CJKfamilydefault }
1832   \tex_protected:D \tl_put_right:Nn \rmfamily { \CJKfamily \CJKrmdefault }
1833   \tex_protected:D \tl_put_right:Nn \sffamily { \CJKfamily \CJKsfdefault }
1834   \tex_protected:D \tl_put_right:Nn \ttfamily { \CJKfamily \CJKttdefault }
1835   \cs_set_eq:NN \reset@font \normalfont
1836 }

```

`\setCJKmathfont` 设置 CJK 数学字体。

```

1837 \NewDocumentCommand \setCJKmathfont { O{ } m }
1838 { \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。

```

1839 \tl_const:Nn \c_xeCJK_math_tl { CJKmath }
1840 \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`

```

1841 \NewDocumentCommand \xeCJKsetmathcode { m m m m }
1842 {
1843   \xeCJK_check_num_range:nnNN {#1} {#2} \l_tmpa_int \l_tmpb_int
1844   \int_set:Nn \l_tmpc_int { \xeCJK_math_type:n {#3} }
1845   \loop
1846     \XeTeXmathcode \l_tmpa_int = \l_tmpc_int \use:c { sym #4 } \l_tmpa_int
1847     \if_int_compare:w \l_tmpa_int < \l_tmpb_int
1848       \int_incr:N \l_tmpc_int
1849   \repeat
1850 }
1851 \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。

```

1852 \tl_put_left:Nn \document
1853 {
1854   \group_end:
1855   \normalfont

```

```

1856 \prop_if_empty:NTF \g_xeCJK_family_prop
1857 { \xeCJK_warning:n { no-CJKfamily } }
1858 {
1859   \bool_if:NT \g_xeCJK_math_bool
1860   {
1861     \xeCJK_family_if_exist:nTF \c_xeCJK_math_tl
1862     { \tl_set:Nx \l_tmpa_tl \c_xeCJK_math_tl }
1863     {
1864       \xeCJK_family_if_exist:nTF \CJKfamilydefault
1865       { \tl_set:Nx \l_tmpa_tl \CJKfamilydefault }
1866       { \tl_set_eq:NN \l_tmpa_tl \q_no_value }
1867     }
1868     \prop_get:NVNT \g_xeCJK_family_name_prop \l_tmpa_tl \l_xeCJK_math_family_tl
1869     {
1870       \DeclareSymbolFont \c_xeCJK_math_tl \g_fontspec_encoding_tl
1871       \l_xeCJK_math_family_tl \mddefault \shapedefault
1872       \SetSymbolFont \c_xeCJK_math_tl { bold } \g_fontspec_encoding_tl
1873       \l_xeCJK_math_family_tl \bfdefault \shapedefault
1874       \clist_map_inline:nn { CJK, FullLeft, FullRight }
1875       {
1876         \clist_map_inline:cn { g_xeCJK_#1_range_clist }
1877         {
1878           \xeCJK_setcharclass_aux:Nn \xeCJKsetmathcode {##1}
1879           \mathalpha \c_xeCJK_math_tl
1880         }
1881       }
1882     }
1883   }
1884 }
1885 \group_begin:
1886 }
1887 \xeCJK_msg_new:nn { no-CJKfamily }
1888 {
1889   It~seems~that~you~have~not~declare~a~CJKfamily.\\
1890   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
1891   \token_to_str:N~\xeCJKmainfont[...]{...}\\\\
1892   in~the~preamble~to~declare~the~main~CJKfamily.\\
1893 }

```

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。

```

1894 \@onlypreamble \setCJKmainfont
1895 \@onlypreamble \setCJKmathfont
1896 \@onlypreamble \setCJKsansfont
1897 \@onlypreamble \setCJKmonofont
1898 \@onlypreamble \setCJKromanfont

```

5.9 处理等宽字体和抄录环境

```

\g_xeCJK_monoscale_fp
\g_xeCJK_exspace_dim

```

```

1899 \fp_new:N \g_xeCJK_monoscale_fp
1900 \dim_new:N \g_xeCJK_exspace_dim
(End definition for \g_xeCJK_monoscale_fp and \g_xeCJK_exspace_dim.)

```


\setCJKmonoscale

```

1901 \NewDocumentCommand \setCJKmonoscale { }
1902 {
1903   \CJKflexiblespacing
1904   \xeCJK_set_monoscale:
1905   \addCJKfontfeatures { Scale = \fp_use:N \g_xeCJK_monoscale_fp }
1906 }
1907 \cs_new_nopar:Nn \xeCJK_set_monoscale:
1908 {
1909   \group_begin:
1910   \fontfamily \ttdefault \selectfont
1911   \fp_gset_from_dim:Nn \g_xeCJK_monoscale_fp
1912   { \c_two \tex_fontdimen:D \c_two \tex_font:D }
1913   \fp_gdiv:Nn \g_xeCJK_monoscale_fp \f@size
1914   \group_end:
1915 }

```

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

\setCJKmonoexspace

```

1916 \NewDocumentCommand \setCJKmonoexspace { } { \xeCJK_set_monoexspace: }
1917 \cs_new_nopar:Nn \xeCJK_set_monoexspace:
1918 {
1919   \group_begin:
1920   \fontfamily \ttdefault \selectfont
1921   \dim_gset:Nn \g_xeCJK_exspace_dim
1922   { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size \p@ }
1923   \group_end:
1924 }

```

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

\CJKfixedspacing

\CJKflexiblespacing

```

1925 \NewDocumentCommand \CJKfixedspacing { }
1926 {
1927   \bool_if:NF \l_xeCJK_fixed_spacing_bool
1928   {
1929     \bool_set_true:N \l_xeCJK_fixed_spacing_bool
1930     \tl_set_eq:NN \l_xeCJK_flexible_punctstyle_tl \l_xeCJK_punctstyle_tl
1931     \bool_set_eq:NN \l_xeCJK_fixed_xecglue_bool \l_xeCJK_xecglue_bool
1932     \cs_set_eq:NN \xeCJK_flexible_ecglue: \CJKecglue
1933     \cs_set_eq:NN \xeCJK_flexible_cjkglue: \CJKglue
1934     \xeCJK_set_monoexspace:
1935     \xeCJKsetup
1936     {
1937       PunctStyle = plain ,
1938       CJKglue    = { \skip_horizontal:N \g_xeCJK_exspace_dim } ,
1939       CJKecglue  = { \skip_horizontal:n { .5\g_xeCJK_exspace_dim } } ,
1940       xCJKecglue = false ,
1941     }
1942   }
1943 }
1944 \NewDocumentCommand \CJKflexiblespacing { }
1945 {

```

```

1946 \bool_if:NT \l_xeCJK_fixed_spacing_bool
1947 {
1948   \bool_set_false:N \l_xeCJK_fixed_spacing_bool
1949   \exp_args:Nx \xeCJKsetup
1950   {
1951     PunctStyle = { \l_xeCJK_flexible_punctstyle_tl } ,
1952     CJKglue    = { \xeCJK_flexible_cjkglue: } ,
1953     CJKecglue  = { \xeCJK_flexible_ecglue: } ,
1954     xCJKecglue = { \bool_if:NTF \l_xeCJK_fixed_xecglue_bool { true } { false } } ,
1955   }
1956 }
1957 }
1958 \bool_new:N \l_xeCJK_fixed_spacing_bool
1959 \AtBeginDocument { \tl_put_right:Nn \verbatim@font \CJKfixedspacing }

```

(End definition for \CJKfixedspacing and \CJKflexiblespacing. These functions are documented on page 6.)

\setCJKmonofont 设置文档的 CJK 等宽字体族。

```

1960 \NewDocumentCommand \setCJKmonofont { s t+ O{} m }
1961 {
1962   \IfBooleanTF {#1}
1963   { \xeCJK_set_family:nnn \CJKttdefault { Mono = Scale, #3 } {#4} }
1964   {
1965     \IfBooleanTF {#2}
1966     { \xeCJK_set_family:nnn \CJKttdefault { Mono = Exspace, #3 } {#4} }
1967     { \xeCJK_set_family:nnn \CJKttdefault {#3} {#4} }
1968   }
1969 }

```

(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。

```

1970 \bool_new:N \g_xeCJK_indent_bool
1971 \keys_define:nn { xeCJK / options }
1972 {
1973   CJKnumber      .bool_set:N = \g_xeCJK_number_bool ,
1974   indentfirst    .bool_set:N = \g_xeCJK_indent_bool ,
1975   normalindentfirst .meta:n = { indentfirst = false } ,
1976   quiet .code:n =
1977   {
1978     \msg_redirect_module:nnn { xeCJK } { warning } { info }
1979     \msg_redirect_module:nnn { xeCJK } { info } { none }
1980     \PassOptionsToPackage { quiet } { fontspec }
1981   },
1982   silent .code:n =
1983   {
1984     \msg_redirect_module:nnn { xeCJK } { warning } { none }
1985     \msg_redirect_module:nnn { xeCJK } { info } { none }
1986     \PassOptionsToPackage { silent } { fontspec }
1987   },
1988   unknown .code:n =

```

```

1989     {
1990         \@ifpackageloaded { fontspec }
1991         { \xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
1992         { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
1993     },
1994 }
1995 \xeCJK_msg_new:nn { key-unknown }
1996 {
1997     Sorry,~but~\l_keys_module_tl\c_space_tl does~not~have~a~key~called~'#1'.\\
1998     The~key~'#1'~is~being~ignored.
1999 }

```

(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。

```

2000 \keys_set:nn { xeCJK / options }
2001 {
2002     CJKglue = { \skip_horizontal:n { \c_zero_skip \@plus .08\baselineskip } } ,
2003     CJKecglue      = \c_space_token ,
2004     CheckSingleglue = \CJKglue ,
2005     xCJKecglue      = false ,
2006     CheckSingle     = false ,
2007     CJKspace        = false ,
2008     CJKmath         = false ,
2009     xeCJKactive      = true ,
2010     indentfirst     = true ,
2011     EmboldenFactor  = 4 ,
2012     SlantFactor     = .167 ,
2013     PunctStyle      = quanjiao ,
2014     KaiMingPunct    = { . 。 ? ! } ,
2015     LongPunct       = { — … — } ,
2016     MiddlePunct     = { · · · } ,
2017     AllowBreakBetweenPuncts = false ,
2018 }

```

```

\CJKsymbol
\CJKpunctsymbol
2019 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2020 \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。

```

2021 \ProcessKeysOptions { xeCJK / options }
2022 \RequirePackage { fontspec } [ 2011/09/13 ]
2023 \bool_if:NT \g_xeCJK_indent_bool
2024 {
2025     \cs_set_eq:NN \@afterindentfalse \@afterindenttrue
2026     \@afterindenttrue
2027 }

```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```

2028 \NewDocumentCommand \xeCJKsetup { m }

```

```

2029 {
2030   \keys_set:nn { xeCJK / options } {#1}
2031   \tex_ignorespaces:D
2032 }
(End definition for \xeCJKsetup. This function is documented on page 3.)

```

```

\xeCJKsetemboldenfactor
\xeCJKsetslantfactor
2033 \NewDocumentCommand \xeCJKsetEmboldenFactor { m }
2034 { \xeCJKsetup { EmboldenFactor = {#1} } }
2035 \NewDocumentCommand \xeCJKsetSlantFactor { m }
2036 { \xeCJKsetup { SlantFactor = {#1} } }
(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor.)

```

```

\punctstyle
\xeCJKplainchr
2037 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2038 \cs_new_nopar:Npn \xeCJKplainchr { \punctstyle { plain } }
(End definition for \punctstyle and \xeCJKplainchr.)

```

```

\CJKsetecglue
2039 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2040 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
(End definition for \CJKsetecglue.)

```

5.12 兼容性修补

∕ 修复倾斜校正。

```

2041 \cs_new_eq:NN \xeCJK_itorr_aux ∕
2042 \bool_new:N \g_xeCJK_itorr_bool
2043 \cs_set_protected_nopar:Npn ∕
2044 {
2045   \scan_stop:
2046   \int_compare:nNt \tex_lastkern:D = \c_three
2047   {
2048     \tex_unkern:D \tex_unkern:D
2049     \bool_gset_true:N \g_xeCJK_itorr_bool
2050   }
2051   \xeCJK_itorr_aux
2052 }
2053 \cs_set_eq:NN @@italiccorr ∕
(End definition for ∕.)

```

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

```

\xeCJK_set_others_toks:n
\xeCJK_Others_class
2054 \cs_new_nopar:Nn \xeCJK_compatibility_hook:
2055 {
2056   \int_compare:nNf
2057   { \c_three + \seq_length:N \g_xeCJK_class_seq } = \xe@alloc@intercharclass
2058   {
2059     \int_set_eq:NN \l_tmpa_int \c_three
2060     \int_until_do:nNnn \l_tmpa_int = \xe@alloc@intercharclass
2061     {
2062       \int_incr:N \l_tmpa_int

```

```

2063         \seq_if_in:NVF \g_xeCJK_class_seq \l_tmpa_int
2064         { \xeCJK_set_others_toks:n \l_tmpa_int }
2065     }
2066 }
2067 }
2068 \cs_new_nopar:Nn \xeCJK_set_others_toks:n
2069 {
2070     \int_set:Nn \xeCJK_Others_class {#1}
2071     \clist_map_inline:Nn \g_xeCJK_CJK_class_clist
2072     {
2073         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2074         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2075         \exp_args:Nnnx \xeCJK_app_inter_class_toks:nnn {##1} { Others }
2076         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
2077         \exp_args:Nnnx \xeCJK_pre_inter_class_toks:nnn { Others } {##1}
2078         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
2079         \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2080         {
2081             \xeCJK_copy_inter_class_toks:nnnn
2082             { Others } { Boundary } { Default } { Boundary }
2083         }
2084         \tl_if_empty:xT { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2085         {
2086             \xeCJK_copy_inter_class_toks:nnnn
2087             { Boundary } { Others } { Boundary } { Default }
2088         }
2089     }
2090 }
2091 \int_new:N \xeCJK_Others_class
2092 \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。

```

2093 \cs_new_nopar:Nn \xeCJK_patch:Nnn
2094 { \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。
2095 \AtBeginDocument
2096 {
2097     \tl_map_inline:nn
2098     {
2099         \textellipsis \textemdash \textperiodcentered \textcentereddot
2100         \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2101     }
2102     { \xeCJK_patch:Nnn #1 { \group_begin: \makexeCJKinactive } { \group_end: } }
2103     \xeCJK_patch:Nnn \tipaencoding { \makexeCJKinactive } { }
2104     \cs_set_eq:NN \xeCJK_aux_r \r
2105     \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \xeCJK_aux_r {#1} } }
2106     \@ifpackageloaded { pifont }
2107     {
2108         \RenewDocumentCommand \Pifont { m }

```

```

2109      {
2110      \fontencoding {U} \fontfamily {#1} \fontseries {m} \fontshape {n}
2111      \selectfont \makexeCJKinactive
2112      }
2113    } { }
2114  }

```

禁止在 xeCJK 宏包后再载入 CJK 宏包。

```

2115 \tl_set:cn { ver@CJK.sty } { 2050/01/01 }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

2116 \cs_if_free:NT \CJK@ifundefined
2117 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2118 \NewDocumentCommand \xeCJKcaption { o m }
2119 {
2120   \IfValueT {#1} { \XeTeXdefaultencoding "#1" }
2121   \cs_set_nopar:Nx \xeCJK_reset_at_catcode:
2122   {
2123     \exp_not:n { \char_set_catcode:nn { \@ } }
2124     { \char_value_catcode:n { \@ } }
2125   }
2126   \char_set_catcode_letter:N \@
2127   \file_input:n { #2.cpx }
2128   \xeCJK_reset_at_catcode:
2129   \XeTeXdefaultencoding "UTF-8"
2130 }

```

(End definition for \xeCJKcaption.)

\xeCJK_ULprepunctchar:n

\xeCJK_ULpostpunctchar:n

\xeCJK_ULroutines:

```

2131 \cs_new_protected_nopar:Nn \xeCJK_ULprepunctchar:n
2132 {
2133   { \makexeCJKinactive \CJKpunctsymbol {#1} \nobreak }
2134   \tex_ignorespaces:D
2135 }
2136 \cs_new_protected_nopar:Nn \xeCJK_ULpostpunctchar:n
2137 {
2138   { \makexeCJKinactive \CJKpunctsymbol {#1} }
2139   \xeCJK_ignorespaces:
2140 }
2141 \cs_new_protected_nopar:Nn \xeCJK_ULroutines:
2142 {
2143   \xeCJK_inter_class_toks:nnn { Default } { CJK } { \CJKecglue \CJKsymbol }
2144   \xeCJK_inter_class_toks:nnn { HalfLeft } { CJK } { \CJKsymbol }
2145   \xeCJK_inter_class_toks:nnn { HalfRight } { CJK } { \CJKecglue \CJKsymbol }
2146   \xeCJK_inter_class_toks:nnn { Boundary } { CJK } { \xeCJK_Boundary_and_CJK: }
2147   \clist_map_inline:nn { Default, HalfLeft, HalfRight, Boundary }
2148   {
2149     \xeCJK_inter_class_toks:nnn {##1} { FullLeft } { \xeCJK_ULprepunctchar:n }
2150     \xeCJK_inter_class_toks:nnn {##1} { FullRight } { \xeCJK_ULpostpunctchar:n }
2151   }
2152   \xeCJK_UL_subclass_patch_tl
2153 }

```

(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。

```

2154 \AtBeginDocument
2155 {
2156   \cs_if_exist:NT \UL@hook
2157   {
2158     \addto@hook \UL@hook
2159     {
2160       \cs_set_eq:NN \xeCJK_UL_CJKsymbol \CJKsymbol
2161       \cs_set_eq:NN \xeCJK_UL_CJKpunctsymbol \CJKpunctsymbol
2162       \cs_set_nopar:Npn \CJKsymbol #1
2163       {
2164         { \xeCJK_select_font: \xeCJK_UL_CJKsymbol {#1} }
2165         \xeCJK_CJK_kern: \xeCJK_ignorespaces:
2166       }
2167       \cs_set_nopar:Npn \CJKpunctsymbol #1
2168       { { \xeCJK_select_font: \xeCJK_UL_CJKpunctsymbol {#1} } }
2169       \xeCJK_ULroutines:
2170     }
2171   }
2172   \cs_if_exist:NT \XeTeX@CJKfntef@hook
2173   {
2174     \cs_set_nopar:Npn \XeTeX@CJKfntef@hook
2175     { \xeCJK_select_font: \makexeCJKinactive }
2176   }
2177 }

```

使用 CJKnumb 宏包。

```

2178 \bool_if:NT \g_xeCJK_number_bool
2179 {
2180   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2181   \cs_set_nopar:Npn \CJKaddEncHook #1#2 { \cs_set_nopar:cpn { xeCJK_enc_#1 } {#2} }
2182   \cs_set_nopar:Npn \Unicode #1#2
2183   { \tex_char:D \int_eval:n { #1 * \c_two_hundred_fifty_six + #2 } }
2184   \RequirePackage { CJKnumb }
2185   \use:c { xeCJK_enc_ \CJK@UnicodeEnc }
2186   \tl_set:Nn \CJK@tenthousand { 万 }
2187   \tl_set:Nn \CJK@hundredmillion { 亿 }
2188 }
2189 \</package>

```

6 例子

6.1 xeCJK-example-autofake.tex

```

1 \<*ex-autofake>
2 \documentclass{article}
3 \usepackage[AutoFakeBold,AutoFakeSlant]{xeCJK}
4 \setCJKmainfont[BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsun.ttc}
5 \setCJKsansfont[AutoFakeSlant=false,
6   BoldFont=simhei.ttf, ItalicFont=simkai.ttf]{simsun.ttc}

```

```

7 \setCJKmonofont[ItalicFont=simkai.ttf]{simsun.ttc}
8 \begin{document}
9 \centering
10 \begin{tabular}{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 \end{ex-autofake}

```

6.2 xeCJK-example-fallback.tex

```

51 \documentclass{article}
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 北 & 3400 & 西 & 3401 & 𠂇 & 3402 & 个 & 3403 & \\
73 𠂇 & 3404 & 乂 & 3405 & 𠂇 & 3406 & 𠂇 & 3407 & \\
74 𠂇 & 3408 & 𠂇 & 3409 & 𠂇 & 340A & 𠂇 & 340B & \\
75 𠂇 & 340C & 𠂇 & 340D & 𠂇 & 340E & 𠂇 & 341F & \\
76 𠂇 & 3410 & 𠂇 & 3411 & 𠂇 & 3412 & 𠂇 & 3413 & \\
77 𠂇 & 3414 & 𠂇 & 3415 & 𠂇 & 3416 & 𠂇 & 3417 & \\
78 𠂇 & 3418 & 𠂇 & 3419 & 𠂇 & 341A & 𠂇 & 341B & \\
79 𠂇 & 341C & 𠂇 & 341D & 𠂇 & 341E & 𠂇 & 341F & \\
80 𠂇 & 20000 & 𠂇 & 20001 & 𠂇 & 20002 & 𠂇 & 20003 & \\
81 𠂇 & 20004 & 𠂇 & 20005 & 𠂇 & 20006 & 𠂇 & 20007 & \\
82 𠂇 & 20008 & 𠂇 & 20009 & 𠂇 & 2000A & 𠂇 & 2000B & \\
83 𠂇 & 2000C & 𠂇 & 2000D & 𠂇 & 2000E & 𠂇 & 2000F & \\
84 𠂇 & 20010 & 𠂇 & 20011 & 𠂇 & 20012 & 𠂇 & 20013 & \\
85 𠂇 & 20014 & 𠂇 & 20015 & 𠂇 & 20016 & 𠂇 & 20017 & \\
86 𠂇 & 20018 & 𠂇 & 20019 & 𠂇 & 2001A & 𠂇 & 2001B & \\
87 𠂇 & 2001C & 𠂇 & 2001D & 𠂇 & 2001E & 𠂇 & 2001F & \\
88 \end{tabular}
89 \end{table}
90 \end{document}
91 </ex-fallback>

```

6.3 xeCJK-example-subCJKblock.tex

```

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

```

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

107

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

109

110 \CJKspace

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

112

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

114 \CJKnospace}

115 \showtext

116

117 \bigskip

118 \xeCJKCancelSubCJKBlock{Kana,Hangul}

119 \showtext

120

121 \bigskip

122 \xeCJKRestoreSubCJKBlock{Hangul}

123 \showtext

124

125 \begin{table}[ht]

126 \caption{生僻字测试}

127 \medskip\centering

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

129 𪛗 & 3400 & 𪛘 & 3401 & 𪛚 & 3402 & 𪛜 & 3403 & \\\

130 𪛞 & 3404 & 𪛠 & 3405 & 𪛢 & 3406 & 𪛤 & 3407 & \\\

131 𪛦 & 3408 & 𪛨 & 3409 & 𪛪 & 340A & 𪛬 & 340B & \\\

132 𪛮 & 340C & 𪛰 & 340D & 𪛲 & 340E & 𪛴 & 341F & \\\

```

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

```

6.4 xeCJK-example-CJKecglue.tex

```

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

```

6.5 xeCJK-example-checksingle.tex

```

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

```

```

180 \end{equation}
181 一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十。}
182 \begin{document}
183 \hsize=30em
184 \parindent=0pt
185 \showtext
186
187 \hrulefill\bigskip
188
189 \xeCJKsetup{CheckSingle}
190 \showtext
191 \end{document}
192 </ex-single>

```