

xeCJK 宏包

ctex.org

2012/11/23 v3.1.0*

目录

1	简介	1	5.3	功能开关	14
2	基本用法	2	5.4	字符类别设定	15
3	用户手册	2	5.5	字符类别处理	17
3.1	宏包选项	2	5.6	字符输出规则	21
3.2	字体设置与选择	4	5.7	段末孤字处理	27
3.2.1	X _g TeX 的字体名查找	7	5.8	增加 CJK 子分区	29
3.3	CJK 分区字体设置	8	5.9	标点处理	31
3.4	设置 CJK 字符范围	8	5.10	后备字体	41
3.5	标点符号的处理	9	5.11	CJK 字体族声明方式	43
3.5.1	设置特定标点符号的宽度和间距	9	5.12	字体切换	50
3.5.2	定义标点符号处理格式	9	5.13	数学字体设置	55
4	已知问题和兼容性	11	5.14	抄录环境中的间距调整	56
5	xeCJK 代码实现	11	5.15	xeCJK 其它选项	58
5.1	运行环境检查	11	5.16	xeCJK 初始化设置	59
5.2	内部工具	12	5.17	兼容性修补	60
			5.18	xeCJK.cfg	64
			版本历史	65	
			代码索引	65	

1 简介

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

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

xeCJK 使用了 X_gTeX 的一些最新特性,需要 X_gTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 依赖 L^AT_EX3 项目的宏包套件 l3kernel 和 l3packages。xeCJK 还需要通过 fontspec 宏包来调用系统字体。将在 3.1 节介绍的功能选项 CJKnumber 需要 CJK 宏包套件中的 CJKnumb 的支持; indentfirst 选项需要 indentfirst 宏包。此外还需要 xpatch 和 etoolbox 宏包。xeCJK 会自动根据需要载入这些宏包。

xeCJK 的原始作者是孙文昌,2009 年 5 月起宏包被收入 ctex-kit 项目进行维护,目前主要维护者是刘海洋¹和李清²。

*ctex-kit rev368.

¹leoliu.pku@gmail.com

²sobenlee@gmail.com

2 基本用法

与其他 \LaTeX 宏包一样,引入 `xeCJK` 宏包只要在导言区使用

```
\usepackage{xeCJK}
```

在引入 `xeCJK` 宏包之后,只要设置 `CJK` 文字的字体,就可以在文档中使用中日韩文字了。
可以在各种文档类中使用 `xeCJK` 宏包,最简单的示例是:

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

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

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

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

`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。在本文档所在的文件夹的 `example` 目录下面也有一些例子可以参考。

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。`xeCJK` 内部调用 `fontspec` 宏包,可以在调用 `xeCJK` 的时候,使用它的宏包选项。`xeCJK` 会将 `fontspec` 的选项传递给它。

<code>\xeCJKsetup</code>	<code>\xeCJKsetup { \langle key_1 \rangle = \langle var_1 \rangle, \langle key_2 \rangle = \langle var_2 \rangle, \dots }</code>
--------------------------	--

其中 $\langle key_1 \rangle$, $\langle key_2 \rangle$ 是设置选项,而 $\langle var_1 \rangle$, $\langle var_2 \rangle$ 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>LocalConfig \star</code>	<code>LocalConfig = { \langle true false name \rangle }</code>
---	---

是否使用本地配置文件 `xeCJK- $\langle name \rangle$.cfg`。 $\langle name \rangle$ 可以是不包含空格的任意使文件名合法的字符串。如果设置为 `true`,则使用的是 `xeCJK.cfg`; 设置为 `false` 则不载入配置文件。可以把将要在下文介绍到的对 `xeCJK` 的一些设置(例如设置常用 `CJK` 字体、修改字符范围和定义新的标点输出格式等)保存到文件 `xeCJK- $\langle name \rangle$.cfg`。然后把这个文件放在本地的 TDS 目录下的适当位置。使用 \TeX Live 的用户,可以新建下列目录,然后再把 `xeCJK- $\langle name \rangle$.cfg` 放在里面:

```
texlive/texmf-local/tex/xelatex/xecjk
```

最后还需要在命令行下执行 `mktxlsr`,刷新文件名数据库以便 \TeX 系统能够找到它。

<hr/> CJKnumber ★ <hr/>	<p><code>CJKnumber = $\langle true false \rangle$</code></p> <p>为了与 <code>CJKnumb</code> 宏包兼容, <code>xeCJK</code> 作了一些特别处理。因此, 如果需要使用到 <code>CJKnumb</code>, 请通过这个选项调用它。</p>
<hr/> indentfirst ★ <hr/>	<p><code>indentfirst = $\langle true false \rangle$</code></p> <p>是否使用 <code>indentfirst</code> 宏包, 使得跟在章节标题后面的第一段首行也缩进。</p> <p>请注意, <code>xeCJK</code> 宏包中只有上述 <code>LocalConfig</code>、<code>CJKnumber</code> 和 <code>indentfirst</code> 这三个选项需要在调用 <code>xeCJK</code> 时设置, 而不能通过 <code>\xeCJKsetup</code> 来设置。</p>
<hr/> xeCJKactive <hr/>	<p><code>xeCJKactive = $\langle true false \rangle$</code></p> <p>打开/关闭对中文的特殊处理。事实上, 这个选项会打开/关闭 \XeTeX 的整个字符类机制, 依赖这个机制的宏包都会受到影响。</p>
<hr/> CJKspace <hr/>	<p><code>CJKspace = $\langle true false \rangle$</code></p> <p>缺省状态下, <code>xeCJK</code> 会忽略 <code>CJK</code> 文字之间的空格, 使用这一选项来保留它们之间的空格。</p>
<hr/> CJKmath ★ <hr/>	<p><code>CJKmath = $\langle true false \rangle$</code></p> <p>是否支持在数学环境中直接输入 <code>CJK</code> 字符。使用这个选项后, 可以直接在数学环境中输出 <code>CJK</code> 字符。</p>
<hr/> CJKglue <hr/>	<p><code>CJKglue = {\hspace{0pt plus .08\baselineskip}}</code></p> <p>设置 <code>CJK</code> 文字之间插入的 <code>glue</code>, 上边是 <code>xeCJK</code> 的默认值。一般来说, 除非有特殊需要(例如, 改变文字间距等), 否则不需要设置这个选项, 使用默认值即可。如果要设置这个选项, 为了行末的对齐, 设置的 <code>glue</code> 最好有一定的弹性。</p>
<hr/> CJKeckglue <hr/>	<p><code>CJKeckglue = {\glue}</code></p> <p>设置 <code>CJK</code> 文字与西文、<code>CJK</code> 文字与行内数学公式之间的间距, 默认值是一个空格。使用这个选项设置的 <code>glue</code> 最好也要用一定的弹性。请注意, 这里设置的 <code>glue</code> 只影响 <code>xeCJK</code> 根据需要自动添加的空白, 源文件中直接输入的 <code>CJK</code> 文字与西文之间的空格不受影响(直接输出)。有时候 <code>xeCJK</code> 可能不能正确地调整间距, 需要手动加空格。</p>
<hr/> xCJKeckglue <hr/>	<p><code>xCJKeckglue = {\langle true false glue \rangle}</code></p> <p>缺省状态下, <code>xeCJK</code> 不对源文件中直接输入的 <code>CJK</code> 文字与西文之间的空格进行调整, 如果需要调整, 请使用这个选项。如果使用这个选项, 将使用 <code>CJKeckglue</code> 替换源文件中直接输入的 <code>CJK</code> 文字与西文之间的空格。</p>
<hr/> CheckSingle <hr/>	<p><code>CheckSingle = $\langle true false \rangle$</code></p> <p>是否避免单个 <code>CJK</code> 文字单独占一个段落的最后一行。</p>
<hr/> InlineEnv InlineEnv+ InlineEnv- <hr/>	<p><code>InlineEnv = {env1, env2, env3, ...}</code></p> <p>在使用 <code>CheckSingle</code> 选项的时候, <code>xeCJK</code> 会将 <code>CJK</code> 文字后接着的 \LaTeX 环境的开始 <code>\begin{...}</code> 和结束 <code>\end{...}</code> 视为断行的地方, 如果有某些特殊的 \LaTeX 环境没有造成断行, 可以使用这个选项来声明它, 以便 <code>CheckSingle</code> 能正确识别。</p>
<hr/> AutoFallBack <hr/>	<p><code>AutoFallBack = $\langle true false \rangle$</code></p> <p>当文档中有个别生僻字时, 可以使用这个选项, 自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。</p>
<hr/> AutoFakeBold ★ <hr/>	<p><code>AutoFakeBold = {\langle true false 数字 \rangle}</code></p> <p>全局设定当没有声明对应的粗体时, 是否使用伪粗体; 当输入的是数字时, 将使用伪粗体, 并将使用输入的数字作为伪粗体的默认粗细程度。</p>

AutoFakeSlant ☆	AutoFakeSlant = {⟨true false 数字⟩}
	全局设定当没有声明对应的斜体时,是否使用伪斜体;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。
EmboldenFactor ☆	EmboldenFactor = {⟨数字 4⟩}
	设置伪粗体的默认粗细程度。
SlantFactor ☆	SlantFactor = {⟨数字 0.167⟩}
	设置伪斜体的粗细程度,范围是 $-0.999 \sim 0.999$ 。
PunctStyle	PunctStyle = {⟨ <i>quanjiao</i> <i>banjiao</i> <i>kaiming</i> <i>hangmobanjiao</i> <i>CCT</i> <i>plain</i> ...⟩}
	设置标点处理格式。 <code>xeCJK</code> 中预先定义好的格式为
	<div> <div>quanjiao</div> <div>banjiao</div> <div>kaiming</div> <div>hangmobanjiao</div> <div>CCT</div> <div>plain</div> </div> <div> <div>全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;</div> <div>半角式: 所有标点占半个汉字宽度;</div> <div>开明式: 句末点号用全角,其他半角;</div> <div>行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</div> <div>CCT 格式;</div> <div>原样(不调整标点间距)。</div> </div>
	可以使用 3.5.2 中介绍的 <code>\xeCJKDeclarePunctStyle</code> 定义新的标点格式。
KaiMingPunct ☆	KaiMingPunct = {⟨. . ? !⟩}
KaiMingPunct+ ☆	设置开明(<i>kaiming</i>)标点处理格式时的句末点号, <code>KaiMingPunct</code> 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, <code>xeCJK</code> 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 = {⟨length⟩}
	缺省状态下, <code>xeCJK</code> 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 <code>length</code> 的单位最好用 <code>em</code> 等相对距离单位,而不建议使用诸如 <code>pt</code> 之类的绝对距离单位。这里的设置可用于除了 <code>plain</code> 以外的所有标点处理格式。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 3.5.1 节的 <code>\xeCJKsetwidth</code> 。
AllowBreakBetweenPuncts	AllowBreakBetweenPuncts = {⟨true false⟩}
	缺省状态下, <code>xeCJK</code> 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 <code>\rmfamily</code> 和 <code>\textrm</code> 的字体。后面两个参数继承自 <code>fontspec</code> 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 <code>fontspec</code> 宏包的文档。需要说明的是 <code>xeCJK</code> 修改了 <code>AutoFakeBold</code> 和 <code>AutoFakeSlant</code> 选项,以便配合全局伪粗体和伪斜体的设定。

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 [<i>{font features}</i>] {font name}
-------------------	--

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

\setCJKmonofont ★	\setCJKmonofont [<i>{font features}</i>] {font name}
-------------------	--

设置正文等宽族的 CJK 字体,影响 \ttfamily 和 \texttt 的字体。

\xeCJKVerbAddon	\xeCJKVerbAddon
-----------------	-----------------

调整文字间距以便于让 CJK 字符占的宽度等于西文等宽字体中两个空格的宽度。这有利于等宽字体的代码对齐等情形。抄录 (verbatim) 环境将自动使用 \xeCJKVerbAddon 的效果。需要注意的是,\xeCJKVerbAddon 对 xeCJK 的内部进行了比较大的修改,使用它之后,关于标点符号的禁则将会丢失。所以不能单独使用,应该放在分组里限制其作用域,否则是无效的。当然它可以和其它关于代码抄录的宏包配合使用。例如,可以使用于 fancyvrb 宏包的 formatcom 选项。此时设置的西文字体应该确实是等宽的以保证对齐。若西文等宽字体发生变动(包括字体大小),则需要在其后面使用 \xeCJKVerbAddon,重新计算间距的宽度。

\setCJKfamilyfont ★	\setCJKfamilyfont {family} [<i>{font features}</i>] {font name}
---------------------	---

声明新的 CJK 字体族 *family* 并指定字体。

\CJKfamily	\CJKfamily {family}
	\CJKfamily+ {family}
	\CJKfamily- {family}

用于在文档中切换 CJK 字体族, *family* 必须预先声明。 \CJKfamily 仅对 CJK 字符类有效, \CJKfamily+ 对所有字符类均有效, \CJKfamily- 对非 CJK 字符类有效。当 \CJKfamily+ 和 \CJKfamily- 的参数为空时,则使用当前的 CJK 字体族。

\newCJKfontfamily ★	\newCJKfontfamily [<i>family</i>] \font-switch [<i>{font features}</i>] {font name}
---------------------	---

声明新的 CJK 字体族 *family* 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 *family*,这时候 *family* 将等于 *font-switch*。事实上,\newCJKfontfamily 是 \setCJKfamilyfont 和 \CJKfamily 的合并。例如

```
\newCJKfontfamily[song]\songti{SimSun}
```

等价于

```
\setCJKfamilyfont{song}{SimSun}
\newcommand\songti{\CJKfamily{song}}
```

\CJKfontspec	\CJKfontspec [<i>{font features}</i>] {font name}
--------------	---

在文档中随机定义新的 CJK 字体族,并马上使用它。

\defaultCJKfontfeatures ☆	\defaultCJKfontfeatures {font features}
---------------------------	---

全局设置 CJK 字体族的默认选项。例如,使用

```
\defaultCJKfontfeatures{Scale=0.962216}
```

可以将全部 CJK 字体缩小为 0.962216。 xeCJK 宏包的初始化设置是

```
\defaultCJKfontfeatures{Script=CJK}
```

<hr/> <hr/>	<code>\addCJKfontfeatures</code>	<code>\addCJKfontfeatures {}</code> 临时增加当前使用的 CJK 字体的选项。例如,使用 <code>\addCJKfontfeatures{Scale=1.1}</code> 可以将文档中当前使用的 CJK 字体放大为 1.1。
<hr/> <hr/>	<code>\CJKrmdefault</code>	保存 <code>\rmfamily</code> 所使用的 CJK 字体族,默认值是 <code>rm</code> 。类似西文字体的 <code>\rmdefault</code> 。
<hr/> <hr/>	<code>\CJKsfdefault</code>	保存 <code>\sffamily</code> 所使用的 CJK 字体族,默认值是 <code>sf</code> 。类似西文字体的 <code>\sfdefault</code> 。
<hr/> <hr/>	<code>\CJKttdefault</code>	保存 <code>\ttfamily</code> 所使用的 CJK 字体族,默认值是 <code>tt</code> 。类似西文字体的 <code>\ttdefault</code> 。
<hr/> <hr/>	<code>\CJKfamilydefault</code>	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 <code>\renewcommand\familydefault{\sfdefault}</code> <code>\renewcommand\CJKfamilydefault{\CJKsfdefault}</code> 可以将全文的 CJK 和西文默认字体改为无衬线字体族。
<hr/> <hr/>	<code>\setCJKmathfont</code> ★	<code>\setCJKmathfont [] {}</code> 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。
<hr/> <hr/>	<code>\setCJKfallbackfamilyfont</code> ★	<code>\setCJKfallbackfamilyfont {<family>} [] {}</code> 设置 CJK 字体族 <i><family></i> 的备用字体。例如,使用 <code>\setCJKmainfont{SimSun}</code> <code>\setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}</code> 可以将 SimSun-ExtB 作为 SimSun 的备用字体。
<hr/> <hr/>	<code>FallBack</code>	<code>FallBack = { [] { } }</code> <code>xeCJK</code> 在 <i></i> 里增加了 <code>FallBack</code> 这个选项。用来在声明主字体的时候,同时设置备用字体。例如,上面的例子等价于: <code>\setCJKmainfont[FallBack=SimSun-ExtB]{SimSun}</code> 如果 <code>FallBack</code> 的值为空,将设置的是备用字体。例如, <code>\setCJKmainfont[FallBack,AutoFakeBold,Scale=.97]{SimSun-ExtB}</code> 等价于 <code>\setCJKfallbackfamilyfont{\CJKrmdefault}[AutoFakeBold,Scale=.97]{SimSun-ExtB}</code>
<hr/> <hr/>	<code>\setCJKfallbackfamilyfont</code> ★	<code>\setCJKfallbackfamilyfont {<family>} [<common font features>]</code> <code>{</code> <code> [] { } ,</code> <code> [] { } ,</code> <code> </code> <code>}</code> <code>\setCJKfallbackfamilyfont</code> 还可以用于设置多层的备用字体。例如,使用 <code>\setCJKmainfont[AutoFakeBold,AutoFakeSlant]{KaiTi_GB2312}</code> <code>\setCJKfallbackfamilyfont{\CJKrmdefault}[AutoFakeSlant]</code> <code> { [BoldFont=SimHei]{SimSun} ,</code> <code> [AutoFakeBold] {SimSun-ExtB} }</code> 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,Έντονα 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,
Krepeko poševno,nghiêng đặm,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντονα,Bold,Negrita,
Lihavoitu,Gras,Félkövéř,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
Полужирный,Fet,Kalın,Krepeko,đặm,Lodia
Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,
thườg,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³。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, 细明体
NSimSun, 新宋体
PMingLiU, 新细明体
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。

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>}
\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

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

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

`\xeCJKDeclareSubCJKBlock` 应该在声明所有的 CJK 字体族之前使用。如果有某个 CJK 字体族没有设置 `<block>` 选项,将使用 `\CJKfamilydefault` 的 `<block>` 选项作为该 CJK 字体族的 `<block>` 选项。如果希望在使用某 CJK 字体族时,不在 CJK 主分区与 `<block>` 之间切换字体,可以使用 `<block>=*` 选项。带星号的命令除了设置 CJK 子分区以外,还重置标点符号所属的字符类。

`\xeCJKCancelSubCJKBlock`

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

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

`\xeCJKRestoreSubCJKBlock`

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

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

3.4 设置 CJK 字符范围

`\xeCJKDeclareCharClass` ★

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

`<class range>` 的格式和 3.3 节的 `<block range>` 相同。`<class>` 的有效值见源代码(第 5.4 节)。`xeCJK` 已经支持 Unicode 中所有 CJK 文字和标点。一般来说,不要轻易改变字符类别。带星号的命令除了设置字符类别以外,为了确保标点处理的正确性,还重置标点符号所属的字符类。

`\xeCJKResetCharClass` ★

```
\xeCJKResetCharClass
```

用于重置 `xeCJK` 对各个字符类别的设置。

`\xeCJKResetPunctClass` ★

```
\xeCJKResetPunctClass
```

用于重置标点符号所属的字符类。

`\normalspacedchars`

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

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

3.5 标点符号的处理

xeCJK 对标点符号的输出宽度的调整是通过调整其左边或右边的空白宽度来实现的。按照目前的处理方式,对于位于左边的标点符号(如左引号),xeCJK 只能调整它左边的空白;对于位于右边的标点符号(如右引号),xeCJK 只能调整它右边的空白;对于居中的标点符号,则调整其左右空白,以保证其居中。

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 plain 以外的所有标点处理格式。

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

⟨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 0.7 em。

`\xeCJKsetkern` ★ `\xeCJKsetkern {⟨前标点⟩} {⟨后标点⟩} {⟨length⟩}`

xeCJK 会根据选定的标点处理格式自动调整相邻的前后两个 CJK 标点符号的空白宽度。如果需要个别情况进行特殊调整,可以使用这个命令。例如,

`\xeCJKsetkern{:}{“}{0.3em}`

将设置冒号与左双引号之间的空白宽度为 0.3 em。

3.5.2 定义标点符号处理格式

`\xeCJKDeclarePunctStyle` ★ `\xeCJKDeclarePunctStyle {⟨style⟩} {⟨options⟩}`

定义新的标点符号处理格式,已经存在的同名格式将被覆盖。可以设置的选项将在下面介绍。

`\xeCJKEditPunctStyle` ★ `\xeCJKEditPunctStyle {⟨style⟩} {⟨options⟩}`

修改已有的标点符号处理格式。

下面是可以设置的标点符号格式选项。其中左边一栏是选项名称,中间是选项的输入值类型,右边则是相关说明。某些选项之间是互斥的,具有优先级关系。要使下一级的选项有效,则需要先禁用上一级的设置:对于 *⟨boolean⟩* 类型的选项,将其设置为 `false`,对于 *⟨length⟩* 类型的选项,将其设置为 `\maxdimen`,而对于 *⟨real⟩* 类型的选项,将其设置为 `0`。

`enabled-global-setting` *⟨boolean⟩* 是否使用 `\xeCJKsetup` 的 `PunctWidth` 选项和 `\xeCJKsetwidth`、`\xeCJKsetkern` 的设置。默认值是 `true`。

`fixed-punct-width` *⟨length⟩* 设置单个标点符号的宽度。默认值是 `\maxdimen`。

`fixed-punct-ratio` *⟨real⟩* 设置单个标点符号的输出宽度与实际宽度的比例。默认值是 `1.0`。

`mixed-punct-width` *⟨length⟩* 设置句末标点符号的宽度。其中句末标点符号通过 `\xeCJKsetup` 的 `KaiMingPunct` 来设置。默认值是与 `fixed-punct-width` 选项的值相同。

`mixed-punct-width` *⟨real⟩* 设置句末标点符号的宽度比例。默认值是与 `fixed-punct-ratio` 选项的值相同。

`middle-punct-width` *⟨length⟩* 设置居中标点符号的宽度。其中居中标点符号通过 `\xeCJKsetup` 的 `MiddlePunct` 来设置。默认值是与 `fixed-punct-width` 选项的值相同。

`middle-punct-width` *⟨real⟩* 设置居中标点符号的宽度比例。默认值是与 `fixed-punct-ratio` 选项的值相同。

以上三个选项设置的是标点的固定宽度或比例,xeCJK 会根据设定的选项计算标点符号左/右的空白宽度。下面的选项设置的是标点符号左/右的空白宽度或比例,因此不同标点符号的宽度可能会不同。为了使下面的选项生效,需要先禁用上面的相应选项。优先级自上而下。

`fixed-margin-width` *⟨length⟩* 设置标点的左/右空白宽度。默认值是 `\maxdimen`。

`fixed-margin-ratio` *⟨real⟩* 设置标点的左/右空白宽度与字体中该标点的相应实际边界宽度的比例。默认值是 `1.0`。

`mixed-margin-width` *⟨length⟩* 设置句末标点的左/右空白宽度。默认值是与 `fixed-margin-width` 的值相同。

`mixed-margin-ratio` *⟨real⟩* 设置句末标点的左/右空白宽度的比例。默认值是与 `fixed-margin-ratio` 的值相同。

`middle-margin-width` *⟨length⟩* 设置居中标点的两边空白宽度。默认值是与 `fixed-margin-width` 的值相同。

- middle-margin-ratio** *<real>* 设置居中标点的两边空白宽度之和与两边实际两边边界宽度之和的比例。默认值是与 **fixed-margin-ratio** 的值相同。
- add-min-bound-to-margin** *<boolean>* 是否在以上计算结果的基础上再加上标点的左右实际边界宽度中的最小值。这个选项对居中标点无效。默认值是 **false**。
- optimize-margin** *<boolean>* 使用以上设置空白宽度或比例的选项时,最终输出的标点符号左/右的空白宽度可能大于原来的实际边界宽度。若此时本选项被设置为 **true**,则使用原来的实际边界宽度。而使用 **fixed-punct-width** 选项计算得出的左/右宽度可能小于该标点的另一侧宽度,若此时本选项被启用,则使用该标点的另一侧宽度。默认值为 **false**。
- margin-minimum** *<length>* 指定标点符号左/右的最小空白宽度。当经过以上选项设置的空白宽度小于这个选项的值时,则使用这个选项的值。默认值是 **0 pt**。

下面的选项处理的是前后相邻的两个标点符号之间的空白宽度。这些选项是互斥的,优先级自上而下。

- enabled-kerning** *<boolean>* 是否调整前后相邻的两个标点之间的空白宽度。如果设置为 **false**,则每个标点都按原来的输出宽度输出。默认值是 **true**。
- min-bound-to-kerning** *<boolean>* 是否使用当前字体中前面标点实际左右边界的最小值与后面标点实际左右边界的最小值中的最大值作为两个标点之间的空白宽度。默认值是 **false**。
- kerning-total-width** *<length>* 设置两个标点的总共宽度。此时 **xeCJK** 会自动计算两个标点之间的空白宽度。默认值是 **\maxdimen**。
- kerning-total-ratio** *<real>* 设置两个标点的总共输出宽度与实际宽度的比例。默认值是 **0.75**。
- same-align-margin** *<length>* 当前后两个标点位于同侧时,它们之间的空白宽度。默认值是 **\maxdimen**。
- same-align-ratio** *<real>* 当前后两个标点位于同侧时,它们之间的空白宽度与实际输出宽度的比例。默认值是 **0**。
- different-align-margin** *<length>* 当前后两个标点位于异侧时,它们之间的空白宽度。默认值是 **\maxdimen**。
- different-align-ratio** *<real>* 当前后两个标点位于异侧时,它们之间的空白宽度与实际输出宽度的比例。默认值是 **0**。
- kerning-margin-width** *<length>* 设置前后两个标点之间的空白宽度。默认值是 **\maxdimen**。
- kerning-margin-ratio** *<real>* 设置前后两个标点之间的空白宽度与实际输出空白的比例。默认值是 **1.0**。
- optimize-kerning** *<boolean>* 使用以上选项计算出两个标点之间的空白宽度可能小于通过 **min-bound-to-kerning** 选项得出的结果。当出现这一情况时,若此选项被设置为 **true**,则使用该选项的空白宽度。默认值为 **false**。
- kerning-margin-minimum** *<length>* 指定两个标点之间的最小空白宽度。当经过以上选项设置的空白宽度小于这个选项的值时,则使用这个选项的值。默认值是 **0 pt**。

事实上,**xeCJK** 的默认设置就相当于中文全角(**quanjiao**)格式。可以使用上面说明的选项定义新的标点处理格式。例如,使用

```
\xeCJKDeclarePunctStyle { mine }
{
    fixed-punct-ratio      = 0 ,
    fixed-margin-width     = 0 pt ,
    mixed-margin-width     = \maxdimen ,
    mixed-margin-ratio     = 0.5 ,
    middle-margin-width    = \maxdimen ,
    middle-margin-ratio    = 0.5 ,
    add-min-bound-to-margin = true ,
    min-bound-to-kerning   = true ,
    kerning-margin-minimum = 0.1 em
}
```

就定义了一个名为 **mine** 的标点处理格式。可以在通过

```
\xeCJKsetup{PunctStyle=mine}
```

在文档中使用这个格式。它的意义是:使用标点符号的实际左右边界中的最小值作为其左/右空白的宽度,对于句末标点和居中标点,再加上实际边界空白的一半;使用相邻两个标点的实际边界中的较小值作为它们之间的空白宽度,并且最小的空白宽度是 **0.1 em**。再例如,使用

```
\xeCJKEditPunctStyle { hangmobanjiao } { enabled-global-setting = false }
```

将使得 \xeCJKsetkern 等的设置对 hangmobanjiao 这一格式无效。

4 已知问题和兼容性

为了与 CJKnumb 和 CJKfntef 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,以实现汉字加点和可断行的下划线等。

新版本(3.x)的 xeCJK 完全使用 L^AT_EX₃ 的语法来编写。L^AT_EX₃ 放弃了 \outer 宏的概念,因此相关工具(例如 peek 类函数)在遇到 \outer 宏时可能会存在问题。按照目前 xeCJK 的实现方式,在 CJK 文字后面遇到 \outer 宏时会出现类似

! Forbidden control sequence found while scanning use of \token_if_eq_meaning:NNTF 的错误。目前已知的有 cprotect 宏包提供的 \cprotect。它的定义是

```
\outer\long\def\cprotect{\icprotect}
```

因此,这时可以暂时用 \icprotect 代替 \cprotect。事实上,当 cprotect 被引入时,xeCJK 将使用

```
\let\cprotect\icprotect
```

来取消 \cprotect 的外部宏限制。但由于 \cprotect 的特殊性,应该只在外部使用它,即不要让它出现在任何宏的参数中。其它 \outer 宏的情况,可以在它前面加上 \relax 来回避上面的错误。

xeCJK 依赖 X_YL^AT_EX 的 \XeTeXinterchartoks 机制,与使用相同机制的宏包(例如 polyglossia)可能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 <@@=xeCJK>
```

5.1 运行环境检查

xeCJK 必须使用 X_YL^AT_EX 引擎的支持。

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

```
10 \xetex_if_engine:F { \msg_critical:nn { xeCJK } { Require-XeTeX } }
```

应该使用较新版本的 expl3 宏包。

```
11 \msg_new:nnn { xeCJK } { l3-too-old }
12 {
13   Support~package~'#1'~too~old. \\
14   Please~update~an~up~to~date~version~of~the~bundles\\
15   'l3kernel'~and~'l3packages'\\
16   using~your~TeX~package~manager~or~from~CTAN.\\
17   \str_if_eq:nnT {#1} { expl3 } { Loading~xeCJK~will~abort! }
18 }
19 \@ifpackagelater { expl3 } { 2012/11/10 } { }
20 { \msg_critical:nnn { xeCJK } { l3-too-old } { expl3 } }
```

以下日期以前的 xtemplate 宏包关于 \KeyValue 的 Bug 会影响到后面标点符号的处理。

```
21 \RequirePackage { xtemplate }
22 \@ifpackagelater { xtemplate } { 2012/11/10 } { }
23 { \msg_error:nnn { xeCJK } { l3-too-old } { xtemplate } }
24 \RequirePackage { xparse , l3keys2e }
25 \RequirePackage { xpatch }
```

5.2 内部工具

分配临时变量。

```

26 \tl_new:N \l__xeCJK_tmpa_tl
27 \tl_new:N \l__xeCJK_tmpp_tl
28 \int_new:N \l__xeCJK_tmpa_int
29 \int_new:N \l__xeCJK_tmpp_int
30 \int_new:N \l__xeCJK_tmppc_int
31 \box_new:N \l__xeCJK_tmp_box
32 \dim_new:N \l__xeCJK_tmp_dim
33 \skip_new:N \g__xeCJK_tmp_skip
34 \prop_new:N \l__xeCJK_tmp_prop
35 \clist_new:N \l__xeCJK_tmpa_clist
36 \clist_new:N \l__xeCJK_tmpp_clist

```

__xeCJK_msg_new:nn 各种信息函数的缩略形式。

```

\__xeCJK_error:n 37 \cs_new_protected_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:nx 38 \cs_new_protected_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_warning:nx 39 \cs_new_protected_nopar:Npn \__xeCJK_error:nx { \msg_error:nnx { xeCJK } }
\__xeCJK_info:nx 40 \cs_new_protected_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\__xeCJK_info:nxx 41 \cs_new_protected_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
42 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
43 \cs_new_protected_nopar:Npn \__xeCJK_info:nx { \msg_info:nnx { xeCJK } }
44 \cs_new_protected_nopar:Npn \__xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }

```

(End definition for __xeCJK_msg_new:nn and others.)

\xeCJK_allow_break:

```

\xeCJK_no_break: 45 \cs_new_protected_nopar:Npn \xeCJK_allow_break: { \tex_penalty:D \c_zero }
46 \cs_new_protected_nopar:Npn \xeCJK_no_break: { \tex_penalty:D \c_ten_thousand }

```

(End definition for \xeCJK_allow_break: and \xeCJK_no_break:)

__xeCJK_at_end_preamble:n

在 \document 前后加上各种钩子。

```

\__xeCJK_after_preamble:n 47 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
\__xeCJK_after_end_preamble:n 48 \tl_new:N \g__xeCJK_after_preamble_hook_tl
49 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl
50 \cs_new_protected:Npn \__xeCJK_at_end_preamble:n #1
51 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
52 \cs_new_protected:Npn \__xeCJK_after_preamble:n #1
53 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
54 \cs_new_protected:Npn \__xeCJK_after_end_preamble:n #1
55 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
56 \AtEndPreamble { \g__xeCJK_at_end_preamble_hook_tl }
57 \AfterPreamble { \g__xeCJK_after_preamble_hook_tl }
58 \AfterEndPreamble { \g__xeCJK_after_end_preamble_hook_tl }

```

(End definition for __xeCJK_at_end_preamble:n, __xeCJK_after_preamble:n, and __xeCJK_after_end_preamble:n)

\xeCJK_reverse:nnn #1 为 #2 或 #3, 若 #1 和 #2 相等, 则返回 #3, 否则返回 #2。

```

59 \cs_new_nopar:Npn \xeCJK_reverse:nnn #1#2#3
60 { \str_if_eq_x:nnTF {#1} {#2} {#3} {#2} }

```

(End definition for \xeCJK_reverse:nnn)

\xeCJK_tl_remove_outer_braces:N

去掉 #1 外层的分组括号。

```

\xeCJK_tl_remove_outer_braces:N 61 \cs_new_protected_nopar:Npn \xeCJK_tl_remove_outer_braces:N #1
62 { \tl_set:Nx #1 { \exp_args:NV \xeCJK_tl_remove_outer_braces:n #1 } }
63 \cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
64 { \__xeCJK_tl_remove_outer_braces:w #1 \q_stop }
65 \cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \q_stop
66 {
67   \bool_if:nTF { \tl_if_single_p:n {#1} && \tl_if_head_is_group_p:n {#1} }
68   { \__xeCJK_tl_remove_outer_braces:w #1 \q_stop }
69   { \tl_trim_spaces:n {#1} }
70 }

```

(End definition for \xeCJK_tl_remove_outer_braces:N and \xeCJK_tl_remove_outer_braces:n)

\xeCJK_cs_add:Nnn 在已有宏内容打补丁：在其前后附加内容或者替换掉它的某些内容，只替换掉第一次匹配的内容。

```
\xeCJK_cs_patch:Nnn
71 \cs_new_protected:Npn \xeCJK_cs_add:Nnn #1#2#3
72 {
73   \tl_if_empty:nTF {#3}
74     { \tl_if_empty:nTF {#2} { \use_ii:nn } { \xpretocmd {#1} {#2} } }
75     {
76       \tl_if_empty:nTF {#2} { \xapptocmd {#1} {#3} }
77       { \xpretocmd {#1} {#2} { \xapptocmd {#1} {#3} } { \use_ii:nn } }
78     }
79     { \__xeCJK_info:nxx { patch-cmd } { \token_to_str:N #1 } { succeeded } }
80     { \__xeCJK_warning:nxx { patch-cmd } { \token_to_str:N #1 } { failed } }
81   }
82 \cs_new_protected:Npn \xeCJK_cs_patch:Nnn #1#2#3
83 {
84   \xpatchcmd {#1} {#2} {#3}
85     { \__xeCJK_info:nxx { patch-cmd } { \token_to_str:N #1 } { succeeded } }
86     { \__xeCJK_warning:nxx { patch-cmd } { \token_to_str:N #1 } { failed } }
87   }
88 \__xeCJK_msg_new:nn { patch-cmd } { Patching~'#1'~#2! }
```

(End definition for \xeCJK_cs_add:Nnn and \xeCJK_cs_patch:Nnn)

\xeCJK_cs_clear:N 让控制序列的意义为空。

```
\xeCJK_cs_gclear:N
89 \cs_new_eq:NN \xeCJK_cs_clear:N \tl_clear:N
90 \cs_new_eq:NN \xeCJK_cs_gclear:N \tl_gclear:N
```

(End definition for \xeCJK_cs_clear:N and \xeCJK_cs_gclear:N)

\xeCJK_swap_cs:NN 交换 #1 和 #2 的意义。

```
91 \cs_new_protected:Npn \xeCJK_swap_cs:NN #1#2
92 {
93   \cs_set_eq:NN \__xeCJK_swap_cs_aux:w #1
94   \cs_set_eq:NN #1 #2
95   \cs_set_eq:NN #2 \__xeCJK_swap_cs_aux:w
96   \cs_undefine:N \__xeCJK_swap_cs_aux:w
97 }
```

(End definition for \xeCJK_swap_cs:NN)

\xeCJK_font_gset_to_current:c #1 是控制序列的名字，令它等于当前字体命令。

```
98 \cs_new_protected_nopar:Npn \xeCJK_font_gset_to_current:c #1
99 {
100   \exp_after:wN \cs_gset_eq:NN
101   \cs:w #1 \exp_after:wN \cs_end: \tex_the:D \tex_font:D
102 }
```

(End definition for \xeCJK_font_gset_to_current:c)

\xeCJK_glyph_if_exist_p:N 判断当前字体中是否含有字符 #1。fontspec 中的类似函数在判断为真的时候，会留有一个 \scan_stop:，造成不必要的边界，同时也不完全可展。因此，我们重新定义它。

\xeCJK_glyph_if_exist:NTF

```
103 \prg_new_conditional:Npnn \xeCJK_glyph_if_exist:N #1 { p , T , F , TF }
104 {
105   \etex_ifontchar:D \tex_font:D `#1 \exp_stop_f:
106   \prg_return_true: \else: \prg_return_false: \fi:
107 }
```

(End definition for \xeCJK_glyph_if_exist:N)

\c_xeCJK_space_skip_tl 当前字体状态下，一个字间空格产生的 glue 的长度，包括伸展和收缩部分。没有考虑到 \xspaceskip 和 \spacefactor 的情况。

```
108 \tl_const:Nn \c_xeCJK_space_skip_tl
109 {
110   \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
111   {
112     \tex_fontdimen:D \c_two \tex_font:D
```

```

113         plus \tex_fontdimen:D \c_three \tex_font:D
114         minus \tex_fontdimen:D \c_four \tex_font:D
115     }
116     { \tex_spaceskip:D }
117 }
(End definition for \c_xeCJK_space_skip_tl)

```

`\xeCJK_glue_to_skip:nN` 取得一个 glue 的长度,包括伸展和收缩部分。

```

118 \cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2
119 {
120     \hbox_set:Nn \l__xeCJK_tmp_box
121     { #1 \skip_gset_eq:NN \g__xeCJK_tmp_skip \tex_lastskip:D }
122     \skip_set_eq:NN #2 \g__xeCJK_tmp_skip
123 }
(End definition for \xeCJK_glue_to_skip:nN)

```

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

```

\xeCJK_if_blank_x:nTF
124 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
125 {
126     \if_int_compare:w \pdfTeX_strcmp:D { \c_empty_tl } {#1} = \c_zero
127     \prg_return_true:
128     \else:
129         \if_int_compare:w \pdfTeX_strcmp:D { \c_space_tl } {#1} = \c_zero
130         \prg_return_true: \else: \prg_return_false: \fi:
131     \fi:
132 }
(End definition for \xeCJK_if_blank_x:n)

```

`\xeCJK_if_package_loaded_p:n` 判断宏包是否被引入,可用于文档正文中。

```

\xeCJK_if_package_loaded:nTF
133 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
134 {
135     \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
136     { \prg_return_true: }
137     { \prg_return_false: }
138 }
139 \tl_const:Nx \c__xeCJK_package_ext_tl { \@pkgextension }
(End definition for \xeCJK_if_package_loaded:n)

```

`\xeCJK_int_until_do:nn` 由于定义较为简单,可以比 `\int_until_do:nNn` 稍微快一点点。

```

\__xeCJK_int_until_do:wn
140 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
141 { \__xeCJK_int_until_do:wn \use_none:n { \reverse_if:N \if_int_compare:w #1#2 } }
142 \cs_new_protected:Npn \__xeCJK_int_until_do:wn \use_none:n #1
143 { #1 \exp_after:wN \__xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
(End definition for \xeCJK_int_until_do:nn and \__xeCJK_int_until_do:wn)

```

5.3 功能开关

xeCJKactive 事实上,将开启或关闭 Xe_{La}TeX 的整个字符类机制。

```

144 \keys_define:nn { xeCJK / options }
145 {
146     xeCJKactive .choice: ,
147     xeCJKactive / true .code:n = { \makexeCJKactive } ,
148     xeCJKactive / false .code:n = { \makexeCJKinactive } ,
149     xeCJKactive .default:n = { true }
150 }
(End definition for xeCJKactive This function is documented on page 3.)

```

`\makexeCJKactive`
`\makexeCJKinactive`

```

151 \NewDocumentCommand \makexeCJKactive { } { \XeTeXinterchartokenstate = \c_one }
152 \NewDocumentCommand \makexeCJKinactive { } { \XeTeXinterchartokenstate = \c_zero }
(End definition for \makexeCJKactive and \makexeCJKinactive)
抑制 BOM。
153 \char_set_catcode_ignore:n { "FEFF }

```


5.4 字符类别设定

xeCJK 需要以下字符类别用于字符输出。其中 Default、CJK、FullLeft、FullRight、Boundary 为 Xe_{La}TeX 中预定义类别,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

Default 这五类是 Xe_{La}TeX 预定义的类别。

```

    CJK
FullLeft
FullRight
Boundary
154 \int_const:Nn \c__xeCJK_Default_class_int { \c_zero }
155 \int_const:Nn \c__xeCJK_CJK_class_int { \c_one }
156 \int_const:Nn \c__xeCJK_FullLeft_class_int { \c_two }
157 \int_const:Nn \c__xeCJK_FullRight_class_int { \c_three }
158 \int_const:Nn \c__xeCJK_Boundary_class_int { \c_two_hundred_fifty_five }
(End definition for Default and others.)

```

\xeCJK_new_class:n 新建一个字符类别。

```

\g__xeCJK_class_seq
159 \cs_new_protected_nopar:Npn \xeCJK_new_class:n #1
160 {
161   \exp_args:Nc \newXeTeXintercharclass { c__xeCJK_#1_class_int }
162   \clist_gclear_new:c { g__xeCJK_#1_range_clist }
163   \seq_gput_right:Nv \g__xeCJK_class_seq { c__xeCJK_#1_class_int }
164 }
165 \seq_new:N \g__xeCJK_class_seq
(End definition for \xeCJK_new_class:n and \g__xeCJK_class_seq)

```

HalfLeft 新增西文半角左/右标点和前后原始间距的符号类。

```

HalfRight
NormalSpace
166 \xeCJK_new_class:n { HalfLeft }
167 \xeCJK_new_class:n { HalfRight }
168 \xeCJK_new_class:n { NormalSpace }
(End definition for HalfLeft, HalfRight, and NormalSpace)

```

\c__xeCJK_HalfLeft_chars_clist xeCJK 设置的各个字符类的默认范围。

```

\c__xeCJK_HalfRight_chars_clist
\c__xeCJK_NormalSpace_chars_clist
\c__xeCJK_FullLeft_chars_clist
\c__xeCJK_FullRight_chars_clist
\c__xeCJK_CJK_chars_clist
169 \clist_const:Nn \c__xeCJK_HalfLeft_chars_clist
170 { "28 , "2D , "5B , "60 , "7B }
171 \clist_const:Nn \c__xeCJK_HalfRight_chars_clist
172 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
173 \clist_const:Nn \c__xeCJK_NormalSpace_chars_clist { "2F }
174 \clist_const:Nn \c__xeCJK_FullLeft_chars_clist
175 {
176   "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
177   "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
178   "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
179   "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6
180 }
181 \clist_const:Nn \c__xeCJK_FullRight_chars_clist
182 {
183   "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2030 , "2500 ,
184   "3001 , "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 ,
185   "3015 , "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 ,
186   "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C ,
187   "309D , "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 ,
188   "30E5 , "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE ,
189   "FE50 , "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C ,

```

```

190      "FE5E , "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B ,
191      "FF1F , "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 ,
192      "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E ,
193      "FF9F
194  }
195  \clist_const:Nn \c__xeCJK_CJK_chars_clist
196  {

```

- Hangul Jamo (谚文字母)

```

197      "1100 -> "11FF ,

```

- CJK Radicals Supplement (中日韩部首补充)

```

198      "2E80 -> "2EFF ,

```

- Kangxi Radicals (康熙字典部首)

```

199      "2F00 -> "2FDF ,

```

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

```

200      "2FF0 -> "2FFF ,

```

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

```

201      "3000 -> "303F ,

```

- Hiragana (日文平假名)

```

202      "3040 -> "309F ,

```

- Katakana (日文片假名)

```

203      "30A0 -> "30FF ,

```

- Bopomofo (注音字母)

```

204      "3100 -> "312F ,

```

- Hangul Compatibility Jamo (谚文兼容字母)

```

205      "3130 -> "318F ,

```

- Kanbun (象形字注释标志)

```

206      "3190 -> "319F ,

```

- Bopomofo Extended (注音字母扩展)

```

207      "31A0 -> "31BF ,

```

- CJK Strokes (中日韩笔画)

```

208      "31C0 -> "31EF ,

```

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

```

209      "31F0 -> "31FF ,

```

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

```

210      "3200 -> "32FF ,

```

- CJK Compatibility (中日韩兼容)

```

211      "3300 -> "33FF ,

```

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

```

212      "3400 -> "4DBF ,

```

- Yijing Hexagrams Symbols (易经六十四卦符号)

```

213      "4DC0 -> "4DFF ,

```

- CJK Unified Ideographs (中日韩统一表意文字)


```
214      "4E00 -> "9FFF ,
```
- Yi Syllables (彝文音节)


```
215      "A000 -> "A48F ,
```
- Yi Radicals (彝文字根)


```
216      "A490 -> "A4CF ,
```
- Hangul Jamo Extended-A (谚文扩展 A)


```
217      "A960 -> "A97F ,
```
- Hangul Syllables (谚文音节)


```
218      "AC00 -> "D7AF ,
```
- Kana Supplement (日文假名增补)


```
219      "B000 -> "B0FF ,
```
- Hangul Jamo Extended-B (谚文扩展 B)


```
220      "D7B0 -> "D7FF ,
```
- CJK Compatibility Ideographs (中日韩兼容表意文字)


```
221      "F900 -> "FAFF ,
```
- CJK Compatibility Forms (中日韩兼容形式)


```
222      "FE30 -> "FE4F ,
```
- Halfwidth and Fullwidth Forms (半角及全角形式)


```
223      "FF00 -> "FFEF ,
```
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)


```
224      "20000 -> "2A6DF ,
```
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)


```
225      "2A700 -> "2B73F ,
```
- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)


```
226      "2B740 -> "2B81F ,
```
- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)


```
227      "2F800 -> "2FA1F
228      }
```

(End definition for \c__xeCJK_HalfLeft_chars_clist and others.)

5.5 字符类别处理

```
\g__xeCJK_base_class_seq
\g__xeCJK_CJK_class_seq 229 \seq_new:N \g__xeCJK_base_class_seq
\g__xeCJK_CJK_class_prop 230 \seq_gset_split:Nnn \g__xeCJK_base_class_seq { , }
231 {
232     Default , CJK , FullLeft , FullRight , Boundary ,
233     HalfLeft , HalfRight , NormalSpace
234 }
235 \seq_map_inline:Nn \g__xeCJK_base_class_seq
236 { \clist_gclear_new:c { g__xeCJK_#1_range_clist } }
237 \seq_new:N \g__xeCJK_CJK_class_seq
238 \prop_new:N \g__xeCJK_CJK_class_prop
```

```

239 \cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
240 {
241   \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
242   \prop_gput:Nvn \g__xeCJK_CJK_class_prop { c__xeCJK_#1_class_int } {#1}
243 }
244 \cs_generate_variant:Nn \prop_gput:Nnn { Nvn }
245 \clist_map_function:nN { CJK , FullLeft , FullRight } \__xeCJK_save_CJK_class:n
(End definition for \g__xeCJK_base_class_seq, \g__xeCJK_CJK_class_seq, 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。

```

246 \cs_new_nopar:Npn \xeCJK_class_num:n #1 { \use:c { c__xeCJK_#1_class_int } }
(End definition for \xeCJK_class_num:n)

```

\xeCJK_value_class:n #1 为字符的 Unicode,用于取得该字符所在的字符类。

```

247 \cs_new_nopar:Npn \xeCJK_value_class:n #1
248 { \XeTeXcharclass \tex_numexpr:D #1 \scan_stop: }
(End definition for \xeCJK_value_class:n)

```

\xeCJK_if_same_class:p:NN 判断两个字符是否同属于一个字符类。

```

\xeCJK_if_same_class:NNTF
249 \prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
250 {
251   \if_int_compare:w \xeCJK_value_class:n {`#1} = \xeCJK_value_class:n {`#2}
252   \prg_return_true: \else: \prg_return_false: \fi:
253 }
(End definition for \xeCJK_if_same_class:NN)

```

\xeCJKDeclareCharClass

```

254 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
255 {
256   \xeCJK_declare_char_class:nx {#2} {#3}
257   \IfBooleanT {#1} { \xeCJKResetPunctClass }
258 }
(End definition for \xeCJKDeclareCharClass This function is documented on page 8.)

```

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

```

\__xeCJK_set_char_class_aux:Nnw
259 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
260 {
261   \clist_set:Nn \l__xeCJK_tmpa_clist {#2}
262   \clist_gconcat:ccN
263     { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmpa_clist
264   \clist_map_inline:Nn \l__xeCJK_tmpa_clist
265     {
266       \str_if_eq:nnF {##1} { -> }
267       {
268         \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
269         { \xeCJK_class_num:n {#1} }
270       }
271     }
272 }
273 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw
274 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
275 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
276 \cs_generate_variant:Nn \xeCJK_declare_char_class:nn { nx , nv }
(End definition for \xeCJK_declare_char_class:nn and \__xeCJK_set_char_class_aux:Nnw)

```

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

```

277 \NewDocumentCommand \xeCJKsetcharclass { s m m m }
278 {
279   \xeCJK_set_char_class:nnn {#2} {#3} {#4}
280   \IfBooleanF {#1} { \xeCJKResetPunctClass }
281 }

```

```

282 \cs_new_protected_nopar:Npn \xeCJK_set_char_class:nnn #1#2#3
283 {
284   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
285   \int_set:Nn \l__xeCJK_tmpc_int {#3}
286   \prop_get:NVNTF \g__xeCJK_CJK_class_prop \l__xeCJK_tmpc_int \l__xeCJK_tmpa_tl
287   { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \char_set_catcode_other:n }
288   { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \use_none:n }
289   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
290   {
291     \__xeCJK_set_char_catcode:n { \l__xeCJK_tmpa_int }
292     \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
293     \int_incr:N \l__xeCJK_tmpa_int
294   }
295 }
(End definition for \xeCJK_set_char_class:nnn)

```

`__xeCJK_check_num_range:nnNN`

```

296 \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
297 {
298   \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
299   {
300     \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
301     \int_set_eq:NN #3 #4
302   }
303   {
304     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
305     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
306   }
307 }
(End definition for \__xeCJK_check_num_range:nnNN)

```

`__xeCJK_set_char_class_eq:nn` 将字符类 #2 中的字符全部设置成字符类 #1。只使用于 #2 的字符类范围为离散的逗号列表的情况。

```

308 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
309 {
310   \int_set:Nn \l__xeCJK_tmpa_int { \xeCJK_class_num:n {#1} }
311   \clist_map_inline:cn { c__xeCJK_#2_chars_clist }
312   { \XeTeXcharclass ##1 = \l__xeCJK_tmpa_int }
313 }
(End definition for \__xeCJK_set_char_class_eq:nn)

```

`\normalspacedchars` 声明前后不加间距的字符。

```

314 \NewDocumentCommand \normalspacedchars { m }
315 {
316   \tl_map_inline:xn {#1}
317   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
318 }
319 \cs_generate_variant:Nn \tl_map_inline:nn { x }
(End definition for \normalspacedchars This function is documented on page 8.)

```

`\xeCJKResetPunctClass` 用于重置标点符号所属的字符类。

```

320 \NewDocumentCommand \xeCJKResetPunctClass { }
321 {
322   \xeCJK_declare_char_class:nV { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
323   \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
324   \xeCJK_declare_char_class:nV { FullLeft } \c__xeCJK_FullLeft_chars_clist
325   \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
326 }
(End definition for \xeCJKResetPunctClass This function is documented on page 8.)

```

`\xeCJKResetCharClass` 用于恢复 `\xeCJK` 对字符类别的设置。

```
327 \NewDocumentCommand \xeCJKResetCharClass {  
328   {  
329     \xeCJK_declare_char_class:nV {CJK} \c__xeCJK_CJK_chars_clist  
330     \xeCJK_declare_char_class:nV {NormalSpace} \c__xeCJK_NormalSpace_chars_clist  
331     \xeCJKResetPunctClass  
332   }  
(End definition for \xeCJKResetCharClass This function is documented on page 8.)  
设置字符类别。  
333 \xeCJKResetCharClass
```

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

```
334 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3  
335   { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }  
336 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc , nnx }  
(End definition for \xeCJK_inter_class_toks:nnn)
```

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

```
337 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2  
338   { \tex_the:D \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} }  
(End definition for \xeCJK_get_inter_class_toks:nn)
```

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

```
339 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2  
340   { \xeCJK_inter_class_toks:nnn {#1} {#2} { \prg_do_nothing: } }  
(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。

```
341 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3  
342   {  
343     \xeCJK_inter_class_toks:nnx {#1} {#2}  
344     { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }  
345   }  
346 \cs_generate_variant:Nn \xeCJK_pre_inter_class_toks:nnn { nnx }  
(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。

```
347 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3  
348   {  
349     \xeCJK_inter_class_toks:nnx {#1} {#2}  
350     { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }  
351   }  
352 \cs_generate_variant:Nn \xeCJK_app_inter_class_toks:nnn { nnc , nnx }  
(End definition for \xeCJK_app_inter_class_toks:nnn)
```

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

```
353 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4  
354   {  
355     \xeCJK_inter_class_toks:nnx {#1} {#2}  
356     { \xeCJK_get_inter_class_toks:nn {#3} {#4} }  
357   }  
(End definition for \xeCJK_copy_inter_class_toks:nnnn)
```

`\xeCJK_replace_inter_class_toks:nnnn` 将 #1 和 #2 之间出现的 #3 用 #4 替换。

```
358 \cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4  
359   {  
360     \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }  
361     \tl_replace_all:Nnn \l__xeCJK_tmpa_tl {#3} {#4}  
362     \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_tl }  
363   }  
(End definition for \xeCJK_replace_inter_class_toks:nnnn)
```


\xeCJK_clear_Boundary_and_CJK_toks: 清除边界与 CJK 文字、全角左右标点之间的内容。

```

364 \cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
365 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
366 \cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
367 { \xeCJK_clear_inter_class_toks:nn { Boundary } {#1} }
(End definition for \xeCJK_clear_Boundary_and_CJK_toks:)

```

5.6 字符输出规则

	Default	CJK	FullL	FullR	HalfL	HalfR	Normal	Bound
Default		✓	✓	✓				✓
CJK	✓	✓	✓	✓	✓	✓	✓	✓
FullLeft	✓	✓	✓	✓	✓	✓	✓	✓
FullRight	✓	✓	✓	✓	✓	✓	✓	✓
HalfLeft		✓	✓	✓				
HalfRight		✓	✓	✓				✓
NormalSpace		✓	✓	✓				✓
Boundary	✓	✓	✓	✓	✓		✓	

\xeCJK_class_group_begin:

\xeCJK_class_group_end:

```

368 \cs_new_eq:NN \xeCJK_class_group_begin: \c_group_begin_token
369 \cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token
(End definition for \xeCJK_class_group_begin: and \xeCJK_class_group_end:)

370 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
371 {
372   \clist_map_inline:nn { CJK , FullLeft , FullRight }
373   {
374     \xeCJK_inter_class_toks:nnn {#1} {##1}
375     {
376       \xeCJK_class_group_begin:
377       \xeCJK_select_font:
378       \xeCJK_clear_inter_class_toks:nn {#1} {##1}
379       \xeCJK_clear_Boundary_and_CJK_toks:
380     }
381     \xeCJK_inter_class_toks:nnn {##1} {#1} { \xeCJK_class_group_end: }
382   }
383   \xeCJK_app_inter_class_toks:nnn {#1} { CJK } { \CJKsymbol }
384   \clist_map_inline:nn { FullLeft , FullRight }
385   { \xeCJK_app_inter_class_toks:nnc {#1} {##1} { xeCJK_CJK_and_##1:N } }
386 }

387 \clist_map_inline:nn { Default , HalfLeft , NormalSpace }
388 {
389   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
390   {
391     \bool_if:nTF
392     {
393       \l__xeCJK_xecglue_bool &&
394       \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
395     }
396     {
397       \tex_unskip:D
398       \bool_if:nTF
399       {
400         \xeCJK_if_last_node_p:n { CJK } ||
401         \xeCJK_if_last_node_p:n { CJK-space }
402       }
403       { \CJKecglue } { \c_space_token }
404     }
405     {
406       \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }

```

```

407         { \xeCJK_if_last_node:nT { CJK-space } { \_xeCJK_space_or_xecglue: } }
408     }
409 }
410 \str_if_eq:nnF {#1} { NormalSpace }
411 { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
412 }
413 \clist_map_inline:nn { Default , HalfRight , NormalSpace }
414 {
415     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
416     {
417         \peek_meaning_remove:NTF \ /
418         { \ / { \xeCJK_make_node:n { default } } }
419         {
420             \token_if_space:NTF \l_peek_token
421             { { \xeCJK_make_node:n { default-space } } }
422             { { \xeCJK_make_node:n { default } } }
423         }
424     }
425     \str_if_eq:nnF {#1} { NormalSpace }
426     { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
427 }
428 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
429 {
430     { \xeCJK_check_for_glue: }
431     \xeCJK_class_group_begin:
432     \xeCJK_clear_Boundary_and_CJK_toks:
433     \xeCJK_select_font:
434     \CJKsymbol
435 }

```

\xeCJK_check_for_glue:

```

436 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
437 {
438     \bool_if:nTF
439     { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
440     { \CJKglue }
441     {
442         \bool_if:nTF
443         {
444             \xeCJK_if_last_node_p:n { default } ||
445             \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
446         }
447         { \CJKecglue }
448         {
449             \bool_if:nT
450             {
451                 \l__xeCJK_xecglue_bool &&
452                 ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
453                   \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_ecglue_skip )
454             }
455             {
456                 \tex_unskip:D
457                 \bool_if:nTF
458                 {
459                     \xeCJK_if_last_node_p:n { default-space } ||
460                     \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
461                     \xeCJK_if_last_node_p:n { default }
462                 }
463                 { \CJKecglue }
464                 {
465                     \bool_if:nTF
466                     {

```

```

467             \xeCJK_if_last_node:p:n { CJK }          ||
468             \xeCJK_if_last_node:p:n { CJK-space }
469         }
470         { \bool_if:NTF \l__xeCJK_reserve_space_bool \c_space_token \CJKglue }
471         { \c_space_token }
472     }
473 }
474 }
475 }
476 }
(End definition for \xeCJK_check_for_glue:)

```

\xeCJK_if_last_node:p:n
 \xeCJK_if_last_node:nTF

```

477 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
478 {
479     \if_dim:w \use:c { c__xeCJK_#1_node_dim } = \tex_lastkern:D
480     \prg_return_true: \else: \prg_return_false: \fi:
481 }
(End definition for \xeCJK_if_last_node:n)

```

\xeCJK_def_node:nn
 \xeCJK_make_node:n

用于判断插入的各种 kern。

```

482 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
483 {
484     \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
485     { \dim_gset:cn } { \dim_const:cn }
486     { c__xeCJK_#1_node_dim } {#2}
487 }
488 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
489 {
490     \tex_kern:D - \use:c { c__xeCJK_#1_node_dim }
491     \tex_kern:D \use:c { c__xeCJK_#1_node_dim }
492 }
493 \xeCJK_def_node:nn { CJK }          { 11 sp }
494 \xeCJK_def_node:nn { CJK-space }    { 12 sp }
495 \xeCJK_def_node:nn { default }      { 13 sp }
496 \xeCJK_def_node:nn { default-space } { 14 sp }
(End definition for \xeCJK_def_node:nn and \xeCJK_make_node:n)

```

CJKglue CJK 文字之间插入的 glue。

```

497 \keys_define:nn { xeCJK / options }
498 { CJKglue .code:n = { \cs_set_protected:Npn \CJKglue {#1} } }
(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

```

499 \keys_define:nn { xeCJK / options }
500 {
501     CJKecglue .code:n =
502     {
503         \cs_set_protected:Npn \CJKecglue {#1}
504         \xeCJK_glue_to_skip:nN \CJKecglue \l__xeCJK_ecglue_skip
505     } ,
506     xCJKecglue .choice: ,
507     xCJKecglue / true .code:n =
508     {
509         \bool_set_true:N \l__xeCJK_xecglue_bool
510         \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
511     } ,
512     xCJKecglue / false .code:n =
513     {
514         \bool_set_false:N \l__xeCJK_xecglue_bool
515         \cs_set_eq:NN \__xeCJK_space_or_xecglue: \c_space_token
516     } ,

```

```

517   xCJKecglue / unknown .code:n =
518   {
519     \bool_set_true:N \l__xeCJK_xecglue_bool
520     \cs_set_protected:Npn \CJKecglue {#1}
521     \xeCJK_glue_to_skip:nN \CJKecglue \l__xeCJK_ecglue_skip
522     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
523   } ,
524   xCJKecglue .default:n = { true }
525 }
526 \skip_new:N \l__xeCJK_ecglue_skip
527 \bool_new:N \l__xeCJK_xecglue_bool
(End definition for CJKecglue and xCJKecglue These functions are documented on page 3.)

```

CJKspace 是否保留 CJK 文字间的空白,默认不保留。

```

528 \keys_define:nn { xeCJK / options }
529 {
530   CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
531   space .meta:n = { CJKspace = true } ,
532   nospace .meta:n = { CJKspace = false }
533 }
(End definition for CJKspace This function is documented on page 3.)

```

```

534 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
535 {
536   \xeCJK_class_group_end:
537   { \xeCJK_make_node:n { CJK } }
538   \xeCJK_ignore_spaces:w
539 }

```

\xeCJK_ignore_spaces:w

```

540 \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
541 {
542   \peek_catcode:NTF \c_space_token
543   {
544     \xeCJK_if_last_node:nT { CJK }
545     { \tex_unkern:D \tex_unkern:D { \xeCJK_make_node:n { CJK-space } } }
546     \peek_catcode_ignore_spaces:NTF \c_math_toggle_token
547     { \__xeCJK_space_or_xecglue: }
548     {
549       \bool_if:nT
550       {
551         \token_if_macro_p:N \l_peek_token ||
552         ( \l__xeCJK_reserve_space_bool &&
553           \token_if_other_p:N \l_peek_token )
554       }
555       { \__xeCJK_space_or_xecglue: }
556     }
557   }
558   {
559     \token_if_math_toggle:NTF \l_peek_token
560     { \CJKecglue }
561     { \cs_if_exist:NF \l_peek_token \exp_not:N }
562   }
563 }
564 \cs_new_eq:NN \xeCJK_ignorespaces: \xeCJK_ignore_spaces:w
(End definition for \xeCJK_ignore_spaces:w)

```

```

565 \xeCJK_inter_class_toks:nnn { CJK } { CJK } { \xeCJK_CJK_and_CJK:N }

```

\xeCJK_CJK_and_CJK:N

```

566 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }

```

(End definition for \xeCJK_CJK_and_CJK:N)

```

567 \clist_map_inline:nn { FullLeft , FullRight }
568 {
569   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
570   {
571     \xeCJK_class_group_begin:
572     \xeCJK_clear_Boundary_and_CJK_toks:
573     \xeCJK_select_font:
574     \use:c { xeCJK_CJK_and_#1:N }
575   }
576 }
577 \xeCJK_inter_class_toks:nnn { FullLeft } { Boundary }
578 { \xeCJK_class_group_end: \tex_ignorespaces:D }
579 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
580 { \xeCJK_class_group_end: \CJKglue \tex_ignorespaces:D }
581 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK } { \CJKsymbol }
582 \xeCJK_inter_class_toks:nnn { FullRight } { CJK } { \CJKglue \CJKsymbol }
583 \seq_map_inline:Nn \g__xeCJK_base_class_seq
584 {
585   \bool_if:nF
586   { \str_if_eq_p:nn {#1} { FullLeft } || \str_if_eq_p:nn {#1} { FullRight } }
587   {
588     \xeCJK_pre_inter_class_toks:nnn { FullLeft } {#1} { \xeCJK_after_FullLeft: }
589     \xeCJK_pre_inter_class_toks:nnn { FullRight } {#1} { \xeCJK_after_FullRight: }
590   }
591 }
592 \clist_map_inline:nn { CJK , FullLeft , FullRight }
593 {
594   \clist_map_inline:nn { FullLeft , FullRight }
595   { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
596 }

```

__xeCJK_punct_rule:NN 用于抹去标点符号的左/右空白。

```

597 \cs_new_protected_nopar:Npn \__xeCJK_punct_rule:NN #1#2
598 {
599   \tex_vrule:D
600   width - \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
601   depth \c_zero_dim
602   height \c_zero_dim \scan_stop:
603 }

```

(End definition for __xeCJK_punct_rule:NN)

__xeCJK_punct_glue:NN 根据所选的标点处理方式在标点符号左/右增加的空白。

```

604 \cs_new_protected_nopar:Npn \__xeCJK_punct_glue:NN #1#2
605 {
606   \skip_horizontal:n
607   {
608     \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
609     minus \dim_eval:n { ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} ) / \c_two }
610   }
611 }

```

(End definition for __xeCJK_punct_glue:NN)

__xeCJK_punct_kern:NN 相邻两个标点之间的间距。

```

612 \cs_new_protected_nopar:Npn \__xeCJK_punct_kern:NN #1#2
613 { \tex_kern:D \__xeCJK_use_punct_dim:nnn { kern } {#1} {#2} }

```

(End definition for __xeCJK_punct_kern:NN)

\g__xeCJK_last_punct_tl 用于记录当前的标点符号。

```

614 \tl_new:N \g__xeCJK_last_punct_tl

```

(End definition for \g__xeCJK_last_punct_tl)

\xeCJK_after_FullLeft:

```
615 \cs_new_protected_nopar:Npn \xeCJK_after_FullLeft:
616 {
617   \xeCJK_no_break:
618   \__xeCJK_punct_if_middle:NT \g__xeCJK_last_punct_tl
619   {
620     \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
621     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
622     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
623     \xeCJK_no_break:
624   }
625 }
(End definition for \xeCJK_after_FullLeft:)
```

\xeCJK_after_FullRight:

```
626 \cs_new_protected_nopar:Npn \xeCJK_after_FullRight:
627 {
628   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
629   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
630   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
631 }
(End definition for \xeCJK_after_FullRight:)
```

\xeCJK_CJK_and_FullLeft:N

```
632 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
633 {
634   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
635   \int_compare:nNf \etex_lastnodetype:D = \c_one
636   { \__xeCJK_CJK_FullLeft_glue: \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
637   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
638   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
639   \CJKpunctsymbol {#1}
640 }
641 \cs_new_protected_nopar:Npn \__xeCJK_CJK_FullLeft_glue: { \CJKglue }
(End definition for \xeCJK_CJK_and_FullLeft:N)
```

\xeCJK_CJK_and_FullRight:N

```
642 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
643 {
644   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
645   \__xeCJK_punct_if_long:NTF {#1}
646   { \CJKglue }
647   {
648     \xeCJK_no_break:
649     \__xeCJK_punct_if_middle:NT {#1}
650     {
651       \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
652       \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
653     }
654   }
655   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
656   \CJKpunctsymbol {#1}
657 }
(End definition for \xeCJK_CJK_and_FullRight:N)
```

\xeCJK_FullLeft_and_FullLeft:N

```
658 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
659 {
660   \xeCJK_no_break:
661   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
662   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
663   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
```



```

664 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
665 \CJKpunctsymbol {#1}
666 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N)

```

\xeCJK_FullLeft_and_FullRight:N

```

667 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
668 {
669 \xeCJK_no_break:
670 \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
671 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
672 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
673 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
674 \xeCJK_no_break:
675 \CJKpunctsymbol {#1}
676 }
(End definition for \xeCJK_FullLeft_and_FullRight:N)

```

\xeCJK_FullRight_and_FullLeft:N

```

677 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
678 {
679 \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
680 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
681 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
682 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
683 \__xeCJK_punct_break:
684 \CJKpunctsymbol {#1}
685 }
(End definition for \xeCJK_FullRight_and_FullLeft:N)

```

\xeCJK_FullRight_and_FullRight:N

```

686 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
687 {
688 \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
689 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
690 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
691 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
692 \xeCJK_no_break:
693 \CJKpunctsymbol {#1}
694 }
(End definition for \xeCJK_FullRight_and_FullRight:N)

```

5.7 段末孤字处理

CheckSingle 孤字处理功能选项。

```

695 \keys_define:nn { xeCJK / options }
696 {
697 CheckSingle .choice: ,
698 CheckSingle / true .code:n = { \xeCJKenablechecksingle } ,
699 CheckSingle / false .code:n = { \xeCJKdisablechecksingle } ,
700 CheckSingle .default:n = { true } ,
701 CJKchecksingle .meta:n = { CheckSingle = true }
702 }
(End definition for CheckSingle This function is documented on page 3.)

```

\xeCJKenablechecksingle 孤字处理功能开关。

\xeCJKdisablechecksingle

```

703 \bool_new:N \l__xeCJK_check_single_bool
704 \NewDocumentCommand \xeCJKenablechecksingle { }
705 {
706 \bool_if:NF \l__xeCJK_check_single_bool
707 {

```

```

708         \bool_set_true:N \l__xeCJK_check_single_bool
709         \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
710         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
711     }
712 }
713 \NewDocumentCommand \xeCJKdisablechecksingle { }
714 {
715     \bool_if:NT \l__xeCJK_check_single_bool
716     {
717         \bool_set_false:N \l__xeCJK_check_single_bool
718         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N
719     }
720 }
(End definition for \xeCJKenablechecksingle and \xeCJKdisablechecksingle)

\xeCJK_check_single:Nw
\xeCJK_check_single:NNw
\xeCJK_check_single:NNNw
721 \cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
722 {
723     \peek_catcode:NTF \c_catcode_other_token
724     { \xeCJK_check_single:NNw {#1} }
725     {
726         \bool_if:nTF
727         {
728             \xeCJK_if_blank_x_p:n { \token_get_arg_spec:N \l_peek_token } &&
729             \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
730             \exp_after:wN \token_if_other_p:N \l_peek_token
731         }
732         { \xeCJK_check_single:NNw {#1} }
733         { \__xeCJK_check_single_save:N {#1} }
734     }
735 }
736 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
737 {
738     \peek_catcode:NTF \c_catcode_other_token
739     { \__xeCJK_check_single_save:N {#1} #2 }
740     {
741         \bool_if:nTF
742         {
743             \token_if_space_p:N \l_peek_token ||
744             \token_if_math_toggle_p:N \l_peek_token ||
745             \token_if_cs_p:N \l_peek_token
746         }
747         {
748             \bool_if:nTF { \token_if_space_p:N \l_peek_token }
749             {
750                 \peek_catcode_ignore_spaces:NTF \c_catcode_other_token
751                 { \__xeCJK_check_single_save:N {#1} #2 \c_space_token }
752                 {
753                     \bool_if:nTF
754                     {
755                         \token_if_math_toggle_p:N \l_peek_token ||
756                         \token_if_cs_p:N \l_peek_token
757                     }
758                     { \xeCJK_check_single:NNNw {#1} {#2} \c_space_token }
759                     { \__xeCJK_check_single_save:N {#1} #2 \c_space_token }
760                 }
761             }
762             { \xeCJK_check_single:NNNw {#1} {#2} { } }
763         }
764         { \__xeCJK_check_single_save:N {#1} #2 }
765     }
766 }

```

```

767 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNNw #1#2#3
768 {
769   \bool_if:nTF
770   {
771     \token_if_eq_meaning_p:NN \l_peek_token \par ||
772     \token_if_eq_meaning_p:NN \l_peek_token \[ \% \]
773   }
774   { \CJKsymbol {#1} \xeCJK_no_break: #2 }
775   { \xeCJK_check_single_env:NNNw {#1} {#2} {#3} }
776 }
(End definition for \xeCJK_check_single:Nw, \xeCJK_check_single:NNw, and \xeCJK_check_single:NNNw)

```

\xeCJK_check_single_env:NNNw

```

\__xeCJK_check_single_env:NNNNn
777 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:NNNw #1#2#3
778 {
779   \peek_catcode_remove:NTF \c_math_toggle_token
780   {
781     \peek_catcode:NTF \c_math_toggle_token
782     { \CJKsymbol {#1} \xeCJK_no_break: #2 \c_math_toggle_token }
783     { \__xeCJK_check_single_save:N {#1} #2#3 \c_math_toggle_token }
784   }
785   {
786     \peek_meaning_remove:NTF \begin
787     { \xeCJK_check_single_env:NNNNn {#1} {#2} {#3} \begin }
788     {
789       \peek_meaning_remove:NTF \end
790       { \xeCJK_check_single_env:NNNNn {#1} {#2} {#3} \end }
791       { \__xeCJK_check_single_save:N {#1} #2#3 }
792     }
793   }
794 }
795 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:NNNNn #1#2#3#4#5
796 {
797   \clist_if_in:NnTF \g__xeCJK_inline_env_clist {#5}
798   { \__xeCJK_check_single_save:N {#1} #2#3 }
799   { \CJKsymbol {#1} \xeCJK_no_break: #2 }
800   \scan_stop: #4 {#5}
801 }
(End definition for \xeCJK_check_single_env:NNNw and \xeCJK_check_single_env:NNNNn)

```

InlineEnv

InlineEnv+

InlineEnv-

```

802 \keys_define:nn { xeCJK / options }
803 {
804   InlineEnv .clist_gset:N = \g__xeCJK_inline_env_clist ,
805   InlineEnv+ .code:n =
806   {
807     \clist_gput_right:Nx \g__xeCJK_inline_env_clist {#1}
808     \clist_gremove_duplicates:N \g__xeCJK_inline_env_clist
809   } ,
810   InlineEnv- .code:n =
811   {
812     \clist_map_inline:xn {#1}
813     { \clist_gremove_all:Nn \g__xeCJK_inline_env_clist {##1} }
814   }
815 }
816 \cs_generate_variant:Nn \clist_map_inline:nn { x }

```

(End definition for InlineEnv, InlineEnv+, and InlineEnv- These functions are documented on page 3.)

5.8 增加 CJK 子分区

\g__xeCJK_CJK_sub_class_seq

```

817 \seq_new:N \g__xeCJK_CJK_sub_class_seq

```

(End definition for \g__xeCJK_CJK_sub_class_seq)

\xeCJKDeclareSubCJKBlock 声明 CJK 子区范围, #1 为自定义名称, #2 为子区的 Unicode 范围。

```
818 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
819 { s > { \TrimSpaces } m > { \TrimSpaces } m }
820 {
821   \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
822   \IfBooleanT {#1} { \xeCJKResetPunctClass }
823 }
824 \onlypreamble \xeCJKDeclareSubCJKBlock
```

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

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

```
\xeCJKRestoreSubCJKBlock
825 \bool_new:N \l__xeCJK_sub_cancel_bool
826 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
827 {
828   \bool_if:NF \l__xeCJK_sub_cancel_bool
829   {
830     \bool_set_true:N \l__xeCJK_sub_cancel_bool
831     \__xeCJK_sub_restore_or_cancel:x {#2}
832     \IfBooleanT {#1} { \xeCJKResetPunctClass }
833   }
834 }
835 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
836 {
837   \bool_if:NT \l__xeCJK_sub_cancel_bool
838   {
839     \bool_set_false:N \l__xeCJK_sub_cancel_bool
840     \__xeCJK_sub_restore_or_cancel:x {#2}
841     \IfBooleanT {#1} { \xeCJKResetPunctClass }
842   }
843 }
```

(End definition for \xeCJKCancelSubCJKBlock and \xeCJKRestoreSubCJKBlock These functions are documented on page 8.)

__xeCJK_sub_restore_or_cancel:n

```
844 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
845 {
846   \clist_map_inline:nn {#1}
847   {
848     \int_if_exist:cTF { c__xeCJK_ CJK/##1 _class_int }
849     {
850       \xeCJK_declare_char_class:nx
851       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
852       { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
853     }
854     { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
855   }
856 }
857 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
858 \__xeCJK_msg_new:nn { SubBlock-undefined }
859 {
860   The~CJK~sub~block~'#1'~is~undefined.\\\
861   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
862   to~declare~it.
863 }
```

(End definition for __xeCJK_sub_restore_or_cancel:n)

\xeCJK_declare_sub_char_class:nnn

```
864 \cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
865 {
866   \int_if_exist:cF { c__xeCJK_ #1/#2 _class_int }
867   {
```

```

868         \xeCJK_new_class:n { #1/#2 }
869         \__xeCJK_set_sub_class_toks:nn {#1} {#2}
870         \__xeCJK_new_sub_key:n {#2}
871     }
872     \xeCJK_declare_char_class:nn { #1/#2 } {#3}
873 }
874 \cs_generate_variant:Nn \xeCJK_declare_sub_char_class:nnn { nxx }
(End definition for \xeCJK_declare_sub_char_class:nnn)

\_xeCJK_set_sub_class_toks:nn

875 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
876 {
877     \seq_map_inline:Nn \g__xeCJK_base_class_seq
878     {
879         \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
880         \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
881         \str_if_eq:nnTF {##1} { CJK }
882         {
883             \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
884             { \__xeCJK_switch_font:nn {#1} {#2} }
885         }
886         {
887             \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
888             { \CJKsymbol }
889             { \__xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
890         }
891     }
892     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
893     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
894     {
895         \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
896         \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
897         \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
898         { \__xeCJK_switch_font:nn {#2} {##1} }
899         \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
900         { \__xeCJK_switch_font:nn {##1} {#2} }
901     }
902     \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
903     \__xeCJK_save_CJK_class:n { #1/#2 }
904     \clist_map_inline:nn { CJK , FullLeft , FullRight }
905     {
906         \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
907         { \__xeCJK_switch_font:nn {#2} {#1} }
908     }
909 }
(End definition for \__xeCJK_set_sub_class_toks:nn)

```

5.9 标点处理

\XeTeXglyphbounds 可以得到一个字符的左右边距,用于标点压缩。如果它不可用,则在文档中只能使用 plain 这一标点格式原样输出标点。

```

910 \__xeCJK_msg_new:nn { XeTeX-too-old }
911 {
912     \token_to_str:N \XeTeXglyphbounds \ is~not~defined.\\
913     CJK~punctuation~kerning~will~not~be~available.\\
914     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
915 }
916 \cs_if_exist:NF \XeTeXglyphbounds
917 {
918     \__xeCJK_error:n { XeTeX-too-old }
919     \__xeCJK_after_end_preamble:n { \xeCJKsetup { PunctStyle = plain } }
920 }

```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```
921 \NewDocumentCommand \xeCJKsetwidth { m m }
922 { \tl_map_inline:xn {#1} { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#2} } }
(End definition for \xeCJKsetwidth This function is documented on page 9.)
```

\xeCJKsetkern 手动设置相邻标点的距离。

```
923 \NewDocumentCommand \xeCJKsetkern { m m m }
924 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
(End definition for \xeCJKsetkern This function is documented on page 9.)
```

\c__xeCJK_left_tl

\c__xeCJK_right_tl

```
925 \tl_const:Nn \c__xeCJK_left_tl { left }
926 \tl_const:Nn \c__xeCJK_right_tl { right }
```

(End definition for \c__xeCJK_left_tl and \c__xeCJK_right_tl)

AllowBreakBetweenPuncts 相关选项声明。

KaiMingPunct

LongPunct

MiddlePunct

PunctWidth

```
927 \keys_define:nn { xCJK / options }
928 {
929   AllowBreakBetweenPuncts .choice: ,
930   AllowBreakBetweenPuncts / true .code:n =
931     { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_allow_break: } ,
932   AllowBreakBetweenPuncts / false .code:n =
933     { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_no_break: } ,
934   AllowBreakBetweenPuncts .default:n = { true } ,
935   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
936   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
937   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
938   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
939   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
940   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
941   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
942   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
943   MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
944   PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl
945 }
```

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

\g__xeCJK_special_punct_clist

__xeCJK_set_special_punct:nn

__xeCJK_add_special_punct:nn

__xeCJK_sub_special_punct:nn

```
946 \clist_new:N \g__xeCJK_special_punct_clist
947 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
948 \cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
949 \cs_new_nopar:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
950 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
951 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
952 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
953 {
954   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
955   { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
956   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
957   \tl_map_inline:xn {#2}
958   {
959     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
960     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
961   }
962 }
963 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
964 {
965   \tl_map_inline:xn {#2}
966   {
967     \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
968     {
969       \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
```



```

970         \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
971     }
972 }
973 }
974 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
975 {
976     \tl_map_inline:xn {#2}
977     {
978         \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
979         \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
980     }
981 }

```

(End definition for \g__xeCJK_special_punct_clist and others.)

```

\__xeCJK_punct_if_right_p:N
\__xeCJK_punct_if_long_p:N
\__xeCJK_punct_if_middle_p:N
\__xeCJK_punct_if_mixed_width_p:N
\__xeCJK_punct_if_right:NTF
\__xeCJK_punct_if_long:NTF
\__xeCJK_punct_if_middle:NTF
\__xeCJK_punct_if_mixed_width:NTF
982 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
983 {
984     \if_int_compare:w \xeCJK_value_class:n {#1} = \xeCJK_class_num:n { FullRight }
985     \prg_return_true: \else: \prg_return_false: \fi:
986 }
987 \cs_generate_variant:Nn \prg_new_conditional:Npnn { c }
988 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
989 {
990     \prg_new_conditional:cpnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
991     {
992         \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
993         \prg_return_true: \else: \prg_return_false: \fi:
994     }
995 }

```

(End definition for __xeCJK_punct_if_right:N and others.)

```

\__xeCJK_punct_dim_csname:nn
\__xeCJK_punct_dim_csname:nnn
\__xeCJK_punct_dim_csname:nnnn
\__xeCJK_use_punct_dim:nnn
\__xeCJK_use_punct_dim:nn
\__xeCJK_gset_punct_dim:nnx
\__xeCJK_gset_punct_dim:nnnx
\__xeCJK_gset_punct_dim:nnnnx
996 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
997 { g__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
998 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn
999 { \__xeCJK_punct_dim_csname:nnnn { \l_xeCJK_punct_style_tl } }
1000 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnnn #1#2#3#4
1001 { g__\l_xeCJK_current_font_tl/#1/#2/#3/#4/tl }
1002 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
1003 { \tl_use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
1004 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
1005 { \tl_use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
1006 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnx #1#2#3
1007 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } {#3} }
1008 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnx #1#2#3#4
1009 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } {#4} }
1010 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnnx #1#2#3#4#5
1011 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnnn {#1} {#2} {#3} {#4} } {#5} }

```

(End definition for __xeCJK_punct_dim_csname:nn and others.)

定义标点处理模板。

```

1012 \DeclareObjectType { xeCJK / punctuation } { 3 }
1013 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { 3 }
1014 {
1015     enabled-global-setting : boolean = true ,
1016     fixed-punct-width      : length  = \c_max_dim ,
1017     fixed-punct-ratio      : real    = \c_one_fp ,
1018     mixed-punct-width      : length  = \KeyValue { fixed-punct-width } ,
1019     mixed-punct-ratio      : real    = \KeyValue { fixed-punct-ratio } ,
1020     middle-punct-width     : length  = \KeyValue { fixed-punct-width } ,
1021     middle-punct-ratio     : real    = \KeyValue { fixed-punct-ratio } ,
1022     fixed-margin-width     : length  = \c_max_dim ,

```

```

1023     fixed-margin-ratio      : real      = \c_one_fp ,
1024     mixed-margin-width     : length    = \KeyValue { fixed-margin-width } ,
1025     mixed-margin-ratio     : real      = \KeyValue { fixed-margin-ratio } ,
1026     middle-margin-width    : length    = \KeyValue { fixed-margin-width } ,
1027     middle-margin-ratio    : real      = \KeyValue { fixed-margin-ratio } ,
1028     add-min-bound-to-margin : boolean = false ,
1029     optimize-margin        : boolean = false ,
1030     margin-minimum         : length    = \c_zero_dim ,
1031     enabled-kerning        : boolean = true ,
1032     min-bound-to-kerning   : boolean = false ,
1033     kerning-total-width    : length    = \c_max_dim ,
1034     kerning-total-ratio    : real      = 0.75 ,
1035     optimize-kerning       : boolean = false ,
1036     same-align-margin      : length    = \c_max_dim ,
1037     same-align-ratio       : real      = \c_zero_fp ,
1038     different-align-margin : length    = \c_max_dim ,
1039     different-align-ratio  : real      = \c_zero_fp ,
1040     kerning-margin-width   : length    = \c_max_dim ,
1041     kerning-margin-ratio   : real      = \c_one_fp ,
1042     kerning-margin-minimum : length    = \c_zero_dim
1043 }

1044 \DeclareTemplateCode { xeCJK / punctuation } { basic } { 3 }
1045 {
1046     enabled-global-setting = \l__xeCJK_enabled_global_setting_bool ,
1047     fixed-punct-width     = \l__xeCJK_fixed_punct_width_dim ,
1048     fixed-punct-ratio     = \l__xeCJK_fixed_punct_ratio_fp ,
1049     mixed-punct-width     = \l__xeCJK_mixed_punct_width_dim ,
1050     mixed-punct-ratio     = \l__xeCJK_mixed_punct_ratio_fp ,
1051     middle-punct-width    = \l__xeCJK_middle_punct_width_dim ,
1052     middle-punct-ratio    = \l__xeCJK_middle_punct_ratio_fp ,
1053     fixed-margin-width    = \l__xeCJK_fixed_margin_width_dim ,
1054     fixed-margin-ratio    = \l__xeCJK_fixed_margin_ratio_fp ,
1055     mixed-margin-width    = \l__xeCJK_mixed_margin_width_dim ,
1056     mixed-margin-ratio    = \l__xeCJK_mixed_margin_ratio_fp ,
1057     middle-margin-width   = \l__xeCJK_middle_margin_width_dim ,
1058     middle-margin-ratio   = \l__xeCJK_middle_margin_ratio_fp ,
1059     add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
1060     optimize-margin       = \l__xeCJK_optimize_margin_bool ,
1061     margin-minimum        = \l__xeCJK_margin_minimum_dim ,
1062     enabled-kerning       = \l__xeCJK_enabled_kerning_bool ,
1063     min-bound-to-kerning  = \l__xeCJK_min_bound_to_kerning_bool ,
1064     kerning-total-width   = \l__xeCJK_kerning_total_width_dim ,
1065     kerning-total-ratio   = \l__xeCJK_kerning_total_ratio_fp ,
1066     optimize-kerning      = \l__xeCJK_optimize_kerning_bool ,
1067     same-align-margin     = \l__xeCJK_same_align_margin_dim ,
1068     same-align-ratio      = \l__xeCJK_same_align_ratio_fp ,
1069     different-align-margin = \l__xeCJK_different_align_margin_dim ,
1070     different-align-ratio = \l__xeCJK_different_align_ratio_fp ,
1071     kerning-margin-width  = \l__xeCJK_kerning_margin_width_dim ,
1072     kerning-margin-ratio  = \l__xeCJK_kerning_margin_ratio_fp ,
1073     kerning-margin-minimum = \l__xeCJK_kerning_margin_minimum_dim
1074 }
1075 {
1076     \AssignTemplateKeys
1077     \tl_if_blank:nTF {#3}
1078     { \xeCJK_punct_margin_process:NN {#1} {#2} }
1079     { \xeCJK_punct_kerning_process:NN {#2} {#3} }
1080 }

```

\xeCJK_punct_margin_process:NN

```

1081 \cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
1082 {

```

```

1083 \tl_set:Nx \l__xeCJK_tmpa_tl { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
1084 \tl_set:Nx \l__xeCJK_tmpb_tl
1085 {
1086   \__xeCJK_use_punct_dim:nnn { bound }
1087   { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl } {#2}
1088 }
1089 \dim_set:Nn \l__xeCJK_tmp_dim
1090 {
1091   \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1092   {
1093     \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
1094     { \use_none:n }
1095     {
1096       \xeCJK_if_blank_x:nTF \g__xeCJK_punct_width_tl
1097       { \use:n }
1098       { \g__xeCJK_punct_width_tl \use_none:n }
1099     }
1100   }
1101   { \use:n }
1102   {
1103     \__xeCJK_punct_if_middle:NTF {#2}
1104     { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
1105     {
1106       \__xeCJK_punct_if_mixed_width:NTF {#2}
1107       { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
1108       { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }
1109     }
1110   }
1111 }
1112 \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2}
1113 {
1114   \dim_max:nn
1115   { \l__xeCJK_margin_minimum_dim }
1116   {
1117     \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1118     {
1119       \__xeCJK_punct_if_middle:NTF {#2}
1120       {
1121         (
1122           \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1123         ) / \c_two
1124       }
1125       {
1126         \bool_if:NTF \l__xeCJK_optimize_margin_bool
1127         { \dim_max:nn { \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl } }
1128         { \use:n }
1129         {
1130           \l__xeCJK_tmp_dim - ( \l__xeCJK_tmpb_tl )
1131           - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1132         }
1133       }
1134     }
1135   }
1136   \bool_if:NTF \l__xeCJK_optimize_margin_bool
1137   { \dim_min:nn { \l__xeCJK_tmpa_tl } }
1138   { \use:n }
1139   {
1140     \__xeCJK_punct_if_middle:NTF {#2}
1141     {
1142       \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
1143       { \l__xeCJK_middle_margin_width_dim }
1144       {

```

```

1145         \fp_use:N \l__xeCJK_middle_margin_ratio_fp
1146         \etex_dimexpr:D
1147         ( \l__xeCJK_tmpa_tl + \l__xeCJK_tmpl_tl ) / \c_two
1148         \scan_stop:
1149     }
1150 }
1151 {
1152     \__xeCJK_punct_if_mixed_width:NTF {#2}
1153     { \__xeCJK_margin_width_or_ratio:n { mixed } }
1154     { \__xeCJK_margin_width_or_ratio:n { fixed } }
1155 }
1156 }
1157 }
1158 }
1159 }
1160 }

```

(End definition for \xeCJK_punct_margin_process:NN)

__xeCJK_punct_width_or_ratio:nN

```

1161 \cs_new_nopar:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
1162 {
1163     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
1164     { \use:c { l__xeCJK_#1_punct_width_dim } }
1165     {
1166         \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } = \c_zero_fp
1167         { \c_max_dim }
1168         {
1169             \fp_use:c { l__xeCJK_#1_punct_ratio_fp }
1170             \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
1171         }
1172     }
1173 }

```

(End definition for __xeCJK_punct_width_or_ratio:nN)

__xeCJK_margin_width_or_ratio:n

```

1174 \cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
1175 {
1176     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
1177     { \use:c { l__xeCJK_#1_margin_width_dim } }
1178     {
1179         \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
1180         \etex_dimexpr:D \l__xeCJK_tmpa_tl \scan_stop:
1181     }
1182     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
1183     { + \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpl_tl }
1184 }

```

(End definition for __xeCJK_margin_width_or_ratio:n)

\xeCJK_punct_kerning_process:NN

```

1185 \cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
1186 {
1187     \__xeCJK_gset_punct_dim:nnn { kern } {#1} {#2}
1188     {
1189         \dim_eval:n
1190         {
1191             \bool_if:nTF
1192             {
1193                 \l__xeCJK_enabled_global_setting_bool &&
1194                 \tl_if_exist_p:c { g__xeCJK_punct/kern/#1/#2/tl }
1195             }
1196             { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }
1197         }

```

```

1198         \bool_if:NTF \l__xeCJK_enabled_kerning_bool
1199         { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
1200         { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
1201     }
1202     - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1203     - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} )
1204 }
1205 }
1206 }

```

(End definition for \xeCJK_punct_kerning_process:NN)

__xeCJK_original_kerning_margin:NN 相邻两个标点符号之间的本来空白宽度。

```

1207 \cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
1208 {
1209     \dim_eval:n
1210     {
1211         \__xeCJK_use_punct_dim:nnn
1212         { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
1213         { \c__xeCJK_right_tl } {#1} +
1214         \__xeCJK_use_punct_dim:nnn
1215         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1216         { \c__xeCJK_left_tl } {#2}
1217     }
1218 }

```

(End definition for __xeCJK_original_kerning_margin:NN)

__xeCJK_calc_kerning_margin:NN

```

1219 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:NN #1#2
1220 {
1221     \dim_max:nn
1222     { \l__xeCJK_kerning_margin_minimum_dim }
1223     {
1224         \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
1225         { \__xeCJK_punct_min_bound:NN {#1} {#2} }
1226         {
1227             \bool_if:NTF \l__xeCJK_optimize_kerning_bool
1228             { \dim_max:nn { \__xeCJK_punct_min_bound:NN {#1} {#2} } }
1229             { \use:n }
1230             {
1231                 \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
1232                 { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
1233                 {
1234                     \fp_compare:nNnTF \l__xeCJK_kerning_total_ratio_fp = \c_zero_fp
1235                     {
1236                         \xeCJK_if_same_class:NNTF {#1} {#2}
1237                         { \__xeCJK_kerning_width_or_ratio:nNN { same } }
1238                         { \__xeCJK_kerning_width_or_ratio:nNN { different } }
1239                     }
1240                     {
1241                         \__xeCJK_calc_kerning_margin:nNN
1242                         {
1243                             \fp_use:N \l__xeCJK_kerning_total_ratio_fp
1244                             \etex_dimexpr:D
1245                             \__xeCJK_use_punct_dim:nn { width } {#1} +
1246                             \__xeCJK_use_punct_dim:nn { width } {#2}
1247                             \scan_stop:
1248                         }
1249                     }
1250                 }
1251                 {#1} {#2}
1252             }
1253         }
1254     }

```

```

1254     }
1255 }
(End definition for \__xeCJK_calc_kerning_margin:NN)

```

_xeCJK_kerning_width_or_ratio:nnN

```

1256 \cs_new_nopar:Npn \__xeCJK_kerning_width_or_ratio:nnN #1#2#3
1257 {
1258   \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
1259   { \use:c { l__xeCJK_#1_align_margin_dim } \use_none:nn }
1260   {
1261     \fp_compare:nNnTF { \use:c { l__xeCJK_#1_align_ratio_fp } } = \c_zero_fp
1262     { \use:n }
1263     { \fp_use:c { l__xeCJK_#1_align_ratio_fp } \use_ii:nn }
1264   }
1265   {
1266     \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
1267     { \l__xeCJK_kerning_margin_width_dim \use_none:n }
1268     { \fp_use:N \l__xeCJK_kerning_margin_ratio_fp \use:n }
1269   }
1270   { \etex_dimexpr:D \__xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
1271 }
(End definition for \__xeCJK_kerning_width_or_ratio:nnN)

```

_xeCJK_punct_min_bound:NN

```

1272 \cs_new_nopar:Npn \__xeCJK_punct_min_bound:NN #1#2
1273 {
1274   \dim_max:nn
1275   {
1276     \dim_min:nn
1277     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} }
1278     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
1279   }
1280   {
1281     \dim_min:nn
1282     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} }
1283     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
1284   }
1285 }
(End definition for \__xeCJK_punct_min_bound:NN)

```

_xeCJK_calc_kerning_margin:nnN #2 和 #3 为相邻的两个标点, #1 为要确定的相邻两个标点总共占的宽度。

```

1286 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:nnN #1#2#3
1287 {
1288   \dim_eval:n
1289   {
1290     (#1)
1291     - ( \__xeCJK_use_punct_dim:nnn
1292         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1293         { \c__xeCJK_left_tl } {#2} )
1294     - ( \__xeCJK_use_punct_dim:nnn
1295         { \__xeCJK_punct_if_right:NTF {#3} { glue } { bound } }
1296         { \c__xeCJK_right_tl } {#3} )
1297     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1298     - ( \__xeCJK_use_punct_dim:nn { dimen } {#3} )
1299   }
1300 }
(End definition for \__xeCJK_calc_kerning_margin:nnN)

```

\xeCJK_get_punct_bounds:NN #1 为 \c__xeCJK_left_tl 或 \c__xeCJK_right_tl, #2 为标点符号。

```

1301 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
1302 {
1303   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }

```

```

1304 {
1305   \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1306   {
1307     \__xeCJK_gset_punct_dim:nnx { glue } {#1} {#2} { \c_zero_dim }
1308     \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
1309     \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
1310   }
1311   {
1312     { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
1313     \__xeCJK_punct_if_long:NTF {#2}
1314     { \__xeCJK_gset_punct_dim:nnx { glue } {#1} {#2} { \c_zero_dim } }
1315     {
1316       \UseInstance { xeCJK / punctuation }
1317       { \l_xeCJK_punct_style_tl } {#1} {#2} { }
1318     }
1319   }
1320 }
1321 }
(End definition for \xeCJK_get_punct_bounds:NN)

```

`\xeCJK_calc_punct_dimen:N` 计算标点的左右实际边距和实际尺寸。对于破折号,计算两标点之间的空白,保证它中间不被断开。

```

1322 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
1323 {
1324   \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_left_tl {#1}
1325   { \xeCJK_glyph_bounds:NN \c_one {#1} }
1326   \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_right_tl {#1}
1327   { \xeCJK_glyph_bounds:NN \c_three {#1} }
1328   \dim_set:Nn \l__xeCJK_tmp_dim
1329   {
1330     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} ) +
1331     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1332   }
1333   \__xeCJK_gset_punct_dim:nnx { width } {#1}
1334   { \dim_use:N \etex_fontcharwd:D \tex_font:D `#1 }
1335   \__xeCJK_gset_punct_dim:nnx { dimen } {#1}
1336   { \dim_eval:n { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim } }
1337   \__xeCJK_punct_if_long:NT {#1}
1338   {
1339     \seq_map_inline:Nn \g__xeCJK_punct_style_seq
1340     {
1341       \__xeCJK_gset_punct_dim:nnnnx {##1} { kern } {#1} {#1}
1342       {
1343         \bool_if:nTF
1344         {
1345           \str_if_eq_p:nn {#1} { ^^^^2025 } ||
1346           \str_if_eq_p:nn {#1} { ^^^^2026 }
1347         }
1348         { \c_zero_dim }
1349         { - \dim_use:N \l__xeCJK_tmp_dim }
1350       }
1351     }
1352   }
1353 }
1354 \cs_generate_variant:Nn \xeCJK_calc_punct_dimen:N { f }
(End definition for \xeCJK_calc_punct_dimen:N)

```

`\xeCJK_glyph_bounds:NN` 用 `\XeTeXglyphbounds` 取得标点符号的上下左右空白。

```

1355 \cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
1356 { \dim_use:N \XeTeXglyphbounds #1 ~ \XeTeXcharglyph `#2 \exp_stop_f: }
(End definition for \xeCJK_glyph_bounds:NN)

```

`\xeCJK_get_punct_kerning:NN` 标点挤压。

```
1357 \cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
1358 {
1359   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
1360   {
1361     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1362     { \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#2} { \c_zero_dim } }
1363     {
1364       \UseInstance { xeCJK / punctuation }
1365       { \l_xeCJK_punct_style_tl } { } {#1} {#2}
1366     }
1367   }
1368 }
1369 \cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
(End definition for \xeCJK_get_punct_kerning:NN)
```

PunctStyle

```
1370 \keys_define:nn { xeCJK / options }
1371 {
1372   PunctStyle .choice: ,
1373   PunctStyle .default:n = { quanjiao } ,
1374   PunctStyle / halfwidth .meta:n = { PunctStyle = banjiao } ,
1375   PunctStyle / fullwidth .meta:n = { PunctStyle = quanjiao } ,
1376   PunctStyle / mixedwidth .meta:n = { PunctStyle = kaiming } ,
1377   PunctStyle / marginkerning .meta:n = { PunctStyle = hangmoban_jiao } ,
1378   PunctStyle / plain .code:n =
1379   { \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl } ,
1380   PunctStyle / unknown .code:n =
1381   {
1382     \IfInstanceExistTF { xeCJK / punctuation } { \l_keys_value_tl }
1383     { \tl_set:Nx \l_xeCJK_punct_style_tl { \l_keys_value_tl } }
1384     { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
1385   }
1386 }
1387 \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
1388 \__xeCJK_msg_new:nn { punct-style-unknown }
1389 {
1390   Punctuation~style~"#1"~is~unknown. \\\
1391   The~available~styles~are~listed~as~follow.\\
1392   "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
1393 }
(End definition for PunctStyle This function is documented on page 4.)
```

`\xeCJKDeclarePunctStyle` 定义新的标点处理风格,已经存在的同名风格将被覆盖。

```
1394 \NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
1395 {
1396   \IfInstanceExistTF { xeCJK / punctuation } {#1}
1397   { \__xeCJK_warning:nx { punct-style-already-defined } {#1} }
1398   { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
1399   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
1400 }
1401 \seq_new:N \g__xeCJK_punct_style_seq
1402 \__xeCJK_msg_new:nn { punct-style-already-defined }
1403 {
1404   Punctuation~style~"#1"~is~already~defined!. \\\
1405   The~existing~style~of~"#1"~will~be~overwritten.\\
1406 }
1407 \@onlypreamble \xeCJKDeclarePunctStyle
(End definition for \xeCJKDeclarePunctStyle This function is documented on page 9.)
```

`\xeCJKEditPunctStyle` 对已有的标点处理风格进行修改。

```
1408 \NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
```



```

1409 {
1410     \IfInstanceExistTF { xeCJK / punctuation } {#1}
1411     { \EditInstance { xeCJK / punctuation } {#1} {#2} }
1412     { \_xeCJK_error:nx { punct-style-unknown } {#1} }
1413 }
1414 \@onlypreamble \xeCJKEditPunctStyle
(End definition for \xeCJKEditPunctStyle This function is documented on page 9.)
默认设置即为全角格式。
1415 \xeCJKDeclarePunctStyle { quanjiao } { }
1416 \xeCJKDeclarePunctStyle { hangmobanjiang } { enabled-kerning = false }
1417 \xeCJKDeclarePunctStyle { banjiao }
1418 {
1419     fixed-punct-ratio = 0.5 ,
1420     optimize-margin = true ,
1421     kerning-total-ratio = 0.5 ,
1422     optimize-kerning = true
1423 }
1424 \xeCJKDeclarePunctStyle { kaiming }
1425 {
1426     fixed-punct-ratio = 0.5 ,
1427     mixed-punct-ratio = 0.8 ,
1428     optimize-margin = true ,
1429     kerning-total-ratio = 0.5 ,
1430     optimize-kerning = true
1431 }
1432 \xeCJKDeclarePunctStyle { CCT }
1433 {
1434     fixed-punct-ratio = 0.7 ,
1435     optimize-margin = true ,
1436     kerning-total-ratio = 0.6 ,
1437     optimize-kerning = true
1438 }

```

5.10 后备字体

AutoFallback 后备字体的宏包选项声明。

```

1439 \keys_define:nn { xeCJK / options }
1440 {
1441     AutoFallback .choice: ,
1442     AutoFallback / true .code:n = { \xeCJKenablefallback } ,
1443     AutoFallback / false .code:n = { \xeCJKdisablefallback } ,
1444     AutoFallback .default:n = { true } ,
1445     fallback .meta:n = { AutoFallback = true }
1446 }

```

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

\xeCJKenablefallback 后备字体的启用开关,默认关闭。

```

\xeCJKdisablefallback
1447 \NewDocumentCommand \xeCJKenablefallback { }
1448 {
1449     \bool_if:NF \l__xeCJK_fallback_bool
1450     {
1451         \bool_set_true:N \l__xeCJK_fallback_bool
1452         \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1453         \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
1454     }
1455 }
1456 \NewDocumentCommand \xeCJKdisablefallback { }
1457 {
1458     \bool_if:NT \l__xeCJK_fallback_bool
1459     {

```

```

1460         \bool_set_false:N \l__xeCJK_fallback_bool
1461         \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N
1462     }
1463 }
1464 \bool_new:N \l__xeCJK_fallback_bool
(End definition for \xeCJKenablefallback and \xeCJKdisablefallback)

```

\xeCJK_fallback_test_glyph: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。

```

1465 \cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
1466 {
1467     \xeCJK_glyph_if_exist:NTF {#1}
1468     { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1469     { { \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_family_tl/FallBack } } }
1470 }
(End definition for \xeCJK_fallback_test_glyph:N)

```

\xeCJK_fallback_loop:Nn 循环测试后备字体是否包含字符 #1。若后备字体中存在该字符或者再没有后备字体,则结束循环。

```

1471 \cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
1472 {
1473     \xeCJK_family_if_exist:XTF {#2}
1474     {
1475         \xeCJK_select_font:x {#2}
1476         \xeCJK_glyph_if_exist:NTF {#1}
1477         { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1478         { \xeCJK_fallback_loop:Nn {#1} { #2/FallBack } }
1479     }
1480     {
1481         \__xeCJK_warning:nxx { fallback } {#1} {#2}
1482         \__xeCJK_fallback_save_CJKsymbol:N {#1}
1483     }
1484 }
1485 \__xeCJK_msg_new:nn { fallback }
1486 {
1487     CJKfamily~'#2'~
1488     ( \prop_get:No \g__xeCJK_family_font_name_prop {#2} )~
1489     does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {'#1}).\\
1490 }
1491 \cs_generate_variant:Nn \prop_get:Nn { No }
(End definition for \xeCJK_fallback_loop:Nn)

```

\setCJKfallbackfamilyfont

```

1492 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1493 {
1494     \clist_set:Nx \l__xeCJK_tmpa_clist {#3}
1495     \int_compare:nNnTF { \clist_count:N \l__xeCJK_tmpa_clist } > \c_one
1496     { \xeCJK_set_family_fallback:xxN {#1} {#2} \l__xeCJK_tmpa_clist }
1497     { \xeCJK_set_family:xxx { #1/FallBack } {#2} {#3} }
1498 }
(End definition for \setCJKfallbackfamilyfont This function is documented on page 6.)

```

\xeCJK_set_family_fallback:nnN

```

1499 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnN #1#2#3
1500 {
1501     \group_begin:
1502     \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
1503     \prop_get:NVNF \g__xeCJK_family_font_name_prop
1504     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
1505     { \tl_clear:N \l__xeCJK_font_name_tl }
1506     \clist_map_inline:Nn #3
1507     {
1508         \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }

```

```

1509         \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
1510         \clist_put_left:cn
1511         { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist } {#2}
1512         \xeCJK_set_family:Vvv \l__xeCJK_fallback_family_tl
1513         { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist }
1514         { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_name_tl }
1515     }
1516     \group_end:
1517 }
1518 \cs_generate_variant:Nn \xeCJK_set_family_fallback:nnN { xx }
(End definition for \xeCJK_set_family_fallback:nnN)

```

5.11 CJK 字体族声明方式

```

1519 \bool_new:N \g__xeCJK_auto_fake_bold_bool
1520 \bool_new:N \g__xeCJK_auto_fake_slant_bool
1521 \fp_new:N \g__xeCJK_embolden_factor_fp
1522 \fp_new:N \g__xeCJK_slant_factor_fp

```

AutoFakeBold 伪粗体和伪斜体的宏包选项声明。

AutoFakeSlant

EmboldenFactor

SlantFactor

```

1523 \keys_define:nn { xeCJK / options }
1524 {
1525     AutoFakeBold .choices:nn = { true , false }
1526     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,
1527     AutoFakeBold / unknown .code:n =
1528     {
1529         \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
1530         \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1531     } ,
1532     AutoFakeBold .default:n = { true } ,
1533     AutoFakeSlant .choices:nn = { true , false }
1534     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
1535     AutoFakeSlant / unknown .code:n =
1536     {
1537         \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
1538         \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
1539     } ,
1540     AutoFakeSlant .default:n = { true } ,
1541     EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
1542     SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
1543     BoldFont .meta:n = { AutoFakeBold = true } ,
1544     boldfont .meta:n = { AutoFakeBold = true } ,
1545     SlantFont .meta:n = { AutoFakeSlant = true } ,
1546     slantfont .meta:n = { AutoFakeSlant = true }
1547 }

```

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

__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_seq

```

1548 \seq_new:N \g__xeCJK_sub_key_seq
1549 \cs_new_protected_nopar:Npn \__xeCJK_new_sub_key:n #1
1550 {
1551     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
1552     \keys_define:nn { xeCJK / features }
1553     {
1554         #1 .code:n =
1555         {
1556             \tl_if_blank:nTF {##1}
1557             {
1558                 \bool_set_false:c { \l__xeCJK_copy_#1_bool }
1559                 \bool_set_false:c { \l__xeCJK_add_#1_bool }
1560                 \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
1561             }

```

```

1562         {
1563             \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
1564             \str_if_eq:nnTF {##1} { * }
1565             {
1566                 \bool_set_true:c { l__xeCJK_copy_#1_bool }
1567                 \bool_set_false:c { l__xeCJK_add_#1_bool }
1568             }
1569             {
1570                 \bool_set_false:c { l__xeCJK_copy_#1_bool }
1571                 \bool_set_true:c { l__xeCJK_add_#1_bool }
1572                 \__xeCJK_get_sub_features:nn {#1} {##1}
1573             }
1574         }
1575     } ,
1576     #1 .default:n = { }
1577 }
1578 }
1579 \cs_generate_variant:Nn \__xeCJK_new_sub_key:n { x }
(End definition for \__xeCJK_new_sub_key:n and \g__xeCJK_sub_key_seq)

```

__xeCJK_get_sub_features:nn
 __xeCJK_get_sub_features:wn

```

1580 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
1581 {
1582     \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
1583     \clist_clear:N \l__xeCJK_sub_font_options_clist
1584     \tl_if_head_eq_meaning:VNTF \l__xeCJK_tmpa_tl [ % ]
1585     {
1586         \exp_after:wN \__xeCJK_get_sub_features:wn \l__xeCJK_tmpa_tl \c_empty_tl
1587         \tl_if_blank:VT \l__xeCJK_sub_font_name_tl
1588         {
1589             \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl
1590             \clist_clear:N \l__xeCJK_sub_font_options_clist
1591         }
1592     }
1593     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl }
1594     \tl_if_blank:VTF \l__xeCJK_sub_font_name_tl
1595     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
1596     { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
1597     \clist_set_eq:cN { l__xeCJK_#1_font_options_clist } \l__xeCJK_sub_font_options_clist
1598     \tl_set_eq:cN { l__xeCJK_#1_font_name_tl } \l__xeCJK_sub_font_name_tl
1599 }
1600 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:wn [#1] #2
1601 {
1602     \clist_set:Nn \l__xeCJK_sub_font_options_clist {#1}
1603     \str_if_eq:nnTF {#2} { \c_empty_tl }
1604     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \c_empty_tl }
1605     { \tl_set:Nn \l__xeCJK_sub_font_name_tl {#2} }
1606 }
1607 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
1608 \cs_generate_variant:Nn \tl_if_head_eq_meaning:nNTF { V }
1609 \cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
(End definition for \__xeCJK_get_sub_features:nn and \__xeCJK_get_sub_features:wn)

```

FallBack

```

1610 \__xeCJK_new_sub_key:n { FallBack }
(End definition for FallBack This function is documented on page 6.)

```

\g__xeCJK_features_id_prop

调用字体的属性声明, 同 fontspec 宏包。

BoldFont	1611 \prop_new:N \g__xeCJK_features_id_prop
ItalicFont	1612 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
BoldItalicFont	1613 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
SlantedFont	1614 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
BoldSlantedFont	
BoldFeatures	
ItalicFeatures	

```

1615 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
1616 \prop_put:Nnn \g__xeCJK_features_id_prop { bfsl } { BoldSlanted }
1617 \prop_map_inline:Nn \g__xeCJK_features_id_prop
1618 {
1619   \keys_define:nn { xeCJK / features }
1620   {
1621     #2Font .tl_set:c = { l__xeCJK_font_name_#1_tl } ,
1622     #2Features .clist_set:c = { l__xeCJK_font_feat_#1_clist }
1623   }
1624 }
(End definition for \g__xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

```

1625 \keys_define:nn { xeCJK / features }
1626 {
1627   AutoFakeBold .choice: ,
1628   AutoFakeBold / false .code:n =
1629     { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
1630   AutoFakeBold / unknown .code:n =
1631     {
1632       \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
1633       \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1634     } ,
1635   AutoFakeBold .default:n = { \g__xeCJK_embolden_factor_fp } ,
1636   AutoFakeSlant .choice: ,
1637   AutoFakeSlant / false .code:n =
1638     { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
1639   AutoFakeSlant / unknown .code:n =
1640     {
1641       \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
1642       \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
1643     } ,
1644   AutoFakeSlant .default:n = { \g__xeCJK_slant_factor_fp }
1645 }
(End definition for AutoFakeBold and AutoFakeSlant)

```

__xeCJK_set_family_initial:

```

1646 \cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
1647 {
1648   \__xeCJK_gobble_CJKfamily:
1649   \int_gincr:N \g__xeCJK_family_int
1650   \prop_clear:N \l__xeCJK_add_font_prop
1651   \prop_map_inline:Nn \g__xeCJK_features_id_prop
1652   {
1653     \tl_clear:c { l__xeCJK_font_name_##1_tl }
1654     \clist_clear:c { l__xeCJK_font_feat_##1_clist }
1655   }
1656   \clist_clear:N \l__xeCJK_fontspec_options_clist
1657   \seq_clear:N \l__xeCJK_sub_key_seq
1658   \seq_map_inline:Nn \g__xeCJK_sub_key_seq
1659   {
1660     \bool_set_false:c { l__xeCJK_copy_##1_bool }
1661     \bool_set_false:c { l__xeCJK_add_##1_bool }
1662   }
1663   \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
1664   \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
1665   \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
1666   \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
1667 }
1668 \int_new:N \g__xeCJK_family_int
(End definition for \__xeCJK_set_family_initial:)

```

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

```
1669 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
1670 {
1671   \group_begin:
1672   \__xeCJK_set_family_initial:
1673   \tl_set:Nn \l__xeCJK_family_name_tl {#1}
1674   \clist_set:Nn \l__xeCJK_font_options_clist {#2}
1675   \tl_set:Nn \l__xeCJK_font_name_tl {#3}
1676   \clist_concat:NNN \l__xeCJK_font_features_clist
1677     \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
1678   \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_features_clist
1679   \keys_set_known:nVN { xeCJK / features }
1680     \l__xeCJK_font_features_clist \l__xeCJK_fontspec_options_clist
1681   \__xeCJK_parse_features:
1682   \__xeCJK_pass_features:
1683   \__xeCJK_check_family:V \l__xeCJK_family_name_tl
1684   \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
1685   \__xeCJK_save_family_info:
1686   \__xeCJK_set_sub_block_family:
1687   \group_end:
1688 }
1689 \cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , xxx , Vvv }
(End definition for \xeCJK_set_family:nnn)
```

__xeCJK_remove_duplicate_keys:N

```
1690 \cs_new_protected_nopar:Npn \__xeCJK_remove_duplicate_keys:N #1
1691 {
1692   \prop_clear:N \l__xeCJK_tmp_prop
1693   \keyval_parse:NNV \__xeCJK_prop_put_aux:n \__xeCJK_prop_put_aux:nn #1
1694   \clist_clear:N #1
1695   \prop_map_inline:Nn \l__xeCJK_tmp_prop
1696   {
1697     \tl_if_blank:nTF {##2}
1698       { \clist_put_right:Nn #1 {##1} }
1699       { \clist_put_right:Nn #1 { ##1 = {##2} } }
1700   }
1701 }
1702 \cs_generate_variant:Nn \keyval_parse:NNN { NNV }
1703 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
1704 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} { } }
1705 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
1706 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} {#2} }
(End definition for \__xeCJK_remove_duplicate_keys:N)
```

__xeCJK_gset_family_cs:x

```
1707 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_cs:x #1
1708 {
1709   \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n {#1} }
1710   {
1711     \group_begin:
1712     \__xeCJK_gobble_CJKfamily:
1713     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
1714       { \exp_not:V \l__xeCJK_fontspec_options_clist }
1715       { \exp_not:V \l__xeCJK_font_name_tl }
1716     \__xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
1717     \group_end:
1718   }
1719 }
(End definition for \__xeCJK_gset_family_cs:x)
```

__xeCJK_check_family:n

```
1720 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
```

```

1721 {
1722   \prop_get:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmpa_tl
1723   {
1724     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl
1725     { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
1726     \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmpa_tl }
1727   }
1728 }
1729 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
1730 \__xeCJK_msg_new:nn { CJKfamily-redef }
1731 { Redefining~CJKfamily~'\__xeCJK_msg_family_map:n {#1}'~(#{2}). }
(End definition for \__xeCJK_check_family:n)

```

__xeCJK_add_font:nn

__xeCJK_add_font_if_new:nn

```

1732 \cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
1733 { \prop_put:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
1734 \cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
1735 { \prop_put_if_new:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
1736 \cs_generate_variant:Nn \__xeCJK_add_font:nn { nx , nV , nv }
1737 \cs_generate_variant:Nn \__xeCJK_add_font_if_new:nn { nx , nV , nv }
(End definition for \__xeCJK_add_font:nn and \__xeCJK_add_font_if_new:nn)

```

__xeCJK_add_fake_bold:n

__xeCJK_add_fake_slant:n

```

1738 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
1739 {
1740   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
1741   { FakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
1742 }
1743 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
1744 {
1745   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
1746   { FakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
1747 }
(End definition for \__xeCJK_add_fake_bold:n and \__xeCJK_add_fake_slant:n)

```

__xeCJK_parse_features:

__xeCJK_parse_features:nn

```

1748 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:
1749 { \prop_map_function:NN \g__xeCJK_features_id_prop \__xeCJK_parse_features:nn }
1750 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:nn #1#2
1751 {
1752   \__xeCJK_if_font_select:nTF {#1}
1753   {
1754     \__xeCJK_add_font:nv {#2} { l__xeCJK_font_name_#1_tl }
1755     \__xeCJK_if_it_or_sl:nTF {#1}
1756     {
1757       \__xeCJK_if_font_select:nF { bf #1 }
1758       { \__xeCJK_add_font_if_new:nv {#2} { l__xeCJK_font_name_#1_tl } }
1759     }
1760     {
1761       \str_if_eq:nnT {#1} { bf }
1762       {
1763         \__xeCJK_if_font_select:nF { bfit }
1764         { \__xeCJK_add_font:nv { BoldItalic } \l__xeCJK_font_name_bf_tl }
1765         \__xeCJK_if_font_select:nF { bfls }
1766         { \__xeCJK_add_font:nv { BoldSlanted } \l__xeCJK_font_name_bf_tl }
1767       }
1768     }
1769   }
1770   { \__xeCJK_set_fake_font:nn {#1} {#2} }
1771 }
(End definition for \__xeCJK_parse_features: and \__xeCJK_parse_features:nn)

```

```

__xeCJK_if_font_select_p:n
  __xeCJK_if_it_or_sl_p:n 1772 \prg_new_conditional:Npnn __xeCJK_if_font_select:n #1 { p , T , F , TF }
__xeCJK_if_font_select:nTF 1773 {
  __xeCJK_if_it_or_sl:nTF 1774   \tl_if_blank:vTF { l__xeCJK_font_name_#1_tl }
                            1775   { \prg_return_false: } { \prg_return_true: }
                            1776 }
                            1777 \prg_new_conditional:Npnn __xeCJK_if_it_or_sl:n #1 { p , T , F , TF }
                            1778 {
                            1779   \if_predicate:w
                            1780     \bool_if_p:n { \str_if_eq_p:nn { it } {#1} || \str_if_eq_p:nn { sl } {#1} }
                            1781     \prg_return_true: \else: \prg_return_false: \fi:
                            1782   }
                            1783 \cs_generate_variant:Nn \tl_if_blank:nTF { v }
(End definition for __xeCJK_if_font_select:n and __xeCJK_if_it_or_sl:n)

__xeCJK_set_fake_font:nn

1784 \cs_new_protected_nopar:Npn __xeCJK_set_fake_font:nn #1#2
1785 {
1786   \str_if_eq:nnTF {#1} { bf }
1787   { \bool_if:NT \l__xeCJK_auto_fake_bold_bool { \__xeCJK_add_fake_bold:n {#1} } }
1788   {
1789     \bool_if:NTF \l__xeCJK_auto_fake_slant_bool
1790     {
1791       \bool_if:nT
1792       {
1793         __xeCJK_if_it_or_sl_p:n {#1}
1794         || ( \str_if_eq_p:nn {#1} { bfit }
1795             && ! ( \__xeCJK_if_font_select_p:n { it } ) )
1796         || ( \str_if_eq_p:nn {#1} { bfsl }
1797             && ! ( \__xeCJK_if_font_select_p:n { sl } ) )
1798       }
1799       { \__xeCJK_add_fake_slant:n {#1} }
1800     }
1801     { \__xeCJK_if_it_or_sl:nT {#1} { \__xeCJK_map_it_sl:nn {#1} {#2} } }
1802     \bool_if:nT
1803     {
1804       \l__xeCJK_auto_fake_bold_bool
1805       && ! ( \__xeCJK_if_it_or_sl_p:n {#1} )
1806       && ! ( \__xeCJK_if_font_select_p:n { bf } )
1807     }
1808     { \__xeCJK_add_fake_bold:n {#1} }
1809   }
  \__xeCJK_add_font_if_new:nn {#2} { * }
}
(End definition for __xeCJK_set_fake_font:nn)

__xeCJK_map_it_sl:n

1810 \cs_new_protected_nopar:Npn __xeCJK_map_it_sl:nn #1#2
1811 {
1812   \__xeCJK_if_map_font_select:nT {#1}
1813   {
1814     \__xeCJK_add_font:nx {#2} { \__xeCJK_get_map_font:n {#1} }
1815     \__xeCJK_if_font_select:nF { bf #1 }
1816     { \__xeCJK_add_font_if_new:nx { Bold #2 } { \__xeCJK_get_map_font:n {#1} } }
1817   }
1818 }
(End definition for __xeCJK_map_it_sl:n)

__xeCJK_get_map_font:n

1819 \cs_new_nopar:Npn __xeCJK_get_map_font:n #1
1820 { \exp_not:v { l__xeCJK_font_name_ \xeCJK_reverse:nnn {#1} { it } { sl } _tl } }
(End definition for __xeCJK_get_map_font:n)

```



```

\__xeCJK_if_map_font_select_p:n
\__xeCJK_if_map_font_select:nTF
1821 \prg_new_conditional:Npnn \__xeCJK_if_map_font_select:n #1 { p , T , F , TF }
1822 {
1823   \xeCJK_if_blank_x:nTF { \__xeCJK_get_map_font:n {#1} }
1824   { \prg_return_false: } { \prg_return_true: }
1825 }
(End definition for \__xeCJK_if_map_font_select:n)

\__xeCJK_pass_features:
1826 \cs_new_protected_nopar:Npn \__xeCJK_pass_features:
1827 {
1828   \prop_map_inline:Nn \g__xeCJK_features_id_prop
1829   {
1830     \clist_if_empty:cF { l__xeCJK_font_feat_##1_clist }
1831     {
1832       \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
1833       { ##2Features = { \exp_not:v { l__xeCJK_font_feat_##1_clist } } }
1834     }
1835   }
1836   \prop_map_inline:Nn \l__xeCJK_add_font_prop
1837   { \clist_put_right:Nn \l__xeCJK_fontspec_options_clist { ##1 = { ##2 } } }
1838 }
(End definition for \__xeCJK_pass_features:)

\g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop
1839 \prop_new:N \g__xeCJK_family_name_prop
1840 \prop_new:N \g__xeCJK_family_font_name_prop
1841 \prop_new:N \g__xeCJK_family_font_options_prop
(End definition for \g__xeCJK_family_name_prop, \g__xeCJK_family_font_name_prop, and \g__xeCJK_family_font_options_prop)

\__xeCJK_save_family_info:
1842 \cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
1843 {
1844   \group_begin:
1845   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
1846   { \clist_remove_all:Nn \l__xeCJK_font_options_clist {##1} }
1847   \prop_gput:NVV \g__xeCJK_family_font_name_prop
1848   \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
1849   \prop_gput:NVV \g__xeCJK_family_font_options_prop
1850   \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
1851   \group_end:
1852 }
(End definition for \__xeCJK_save_family_info:)

\__xeCJK_set_sub_block_family:
1853 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_block_family:
1854 {
1855   \seq_map_inline:Nn \l__xeCJK_sub_key_seq
1856   {
1857     \tl_set:Nx \l__xeCJK_sub_family_tl { \l__xeCJK_family_name_tl/##1 }
1858     \bool_if:cT { l__xeCJK_copy_##1_bool } { \__xeCJK_copy_sub_family:n {##1} }
1859     \bool_if:cT { l__xeCJK_add_##1_bool }
1860     {
1861       \xeCJK_set_family:Vvv \l__xeCJK_sub_family_tl
1862       { l__xeCJK_##1_font_options_clist } { l__xeCJK_##1_font_name_tl }
1863     }
1864   }
1865 }
1866 \cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
1867 {
1868   \__xeCJK_check_family:V \l__xeCJK_sub_family_tl
1869   \prop_get:NVT \g__xeCJK_family_font_name_prop

```

```

1870     \l__xeCJK_family_name_tl \l__xeCJK_tmpa_tl
1871     {
1872         \prop_gput:NVV \g__xeCJK_family_font_name_prop
1873         \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl
1874     }
1875     \prop_get:NVNT \g__xeCJK_family_font_options_prop
1876     \l__xeCJK_family_name_tl \l__xeCJK_tmpa_clist
1877     {
1878         \clist_remove_all:Nn \l__xeCJK_tmpa_clist { #1 = * }
1879         \prop_gput:NVV \g__xeCJK_family_font_options_prop
1880         \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
1881     }
1882     \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_tl } }
1883     {
1884         \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }
1885         {
1886             \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
1887             { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_tmpa_tl
1888             {
1889                 \__xeCJK_gset_family_nfss_cs:xx
1890                 { \l__xeCJK_sub_family_tl } { \exp_not:N \l__xeCJK_tmpa_tl }
1891             }
1892         }
1893     }
1894 }
(End definition for \__xeCJK_set_sub_block_family:)

```

__xeCJK_copy_family:nn

```

1895     \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
1896     {
1897         \xeCJK_family_if_exist:xT {#2}
1898         {
1899             \tl_map_inline:nn
1900             {
1901                 \g__xeCJK_family_name_prop
1902                 \g__xeCJK_family_font_name_prop
1903                 \g__xeCJK_family_font_options_prop
1904             }
1905             {
1906                 \prop_get:NnNT ##1 {#2} \l__xeCJK_tmpa_tl
1907                 { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_tl }
1908             }
1909             \cs_gset_eq:cc
1910             { \__xeCJK_family_nfss_csname:n {#1} }
1911             { \__xeCJK_family_nfss_csname:n {#2} }
1912         }
1913     }
1914     \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }
(End definition for \__xeCJK_copy_family:nn)

```

5.12 字体切换

\l_xeCJK_current_font_tl

缓存当前字体的原始格式,以加速编译。

\xeCJK_select_font:

```

1915     \tl_new:N \l_xeCJK_current_font_tl
1916     \tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
1917     \cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
1918     \cs_new_protected_nopar:Npn \xeCJK_select_font:
1919     {
1920         \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
1921         {
1922             \tl_set:Nx \l__xeCJK_current_font_tl { \l_xeCJK_current_font_tl }
1923             \__xeCJK_family_use:x { \l_xeCJK_family_tl }

```

```

1924         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
1925     }
1926 }
1927 \cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
1928 {
1929     \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
1930     {
1931         \tl_set:Nx \l__xeCJK_current_coor_tl { \__xeCJK_font_csname:n {#1} }
1932         \__xeCJK_family_use:x {#1}
1933         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
1934     }
1935 }
1936 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
(End definition for \l__xeCJK_current_font_tl, \xeCJK_select_font:, and \xeCJK_select_font:x)

```

__xeCJK_switch_font:nn 两个 CJK 分区之间的字体切换。

```

1937 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
1938 {
1939     \str_if_eq:nnF {#1} {#2}
1940     {
1941         \__xeCJK_info:nxx { CJK-block } {#1} {#2}
1942         \str_if_eq:nnTF {#2} { CJK }
1943         { \xeCJK_select_font: }
1944         { \__xeCJK_block_select_font:n {#2} }
1945     }
1946 }
1947 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~'#1'~to~'#2'. }
(End definition for \__xeCJK_switch_font:nn)

```

__xeCJK_block_select_font:n 若当前 CJK 字体族没有定义子分区 #1 的字体,则使用 \CJKfamilydefault 的对应分区字体; 若 \CJKfamilydefault 也没有定义该分区字体,则使用当前 CJK 字体族的主分区字体。

```

1948 \cs_new_protected_nopar:Npn \__xeCJK_block_select_font:n #1
1949 {
1950     \cs_if_exist_use:cF { \__xeCJK_font_csname:n { \l__xeCJK_family_tl/#1 } }
1951     {
1952         \tl_set:Nx \l__xeCJK_current_coor_tl
1953         { \__xeCJK_font_csname:n { \l__xeCJK_family_tl/#1 } }
1954         \xeCJK_family_if_exist:xF { \l__xeCJK_family_tl/#1 }
1955         {
1956             \__xeCJK_copy_family:xx { \l__xeCJK_family_tl/#1 }
1957             {
1958                 \cs_if_exist:cTF { \__xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
1959                 { \CJKfamilydefault/#1 } { \l__xeCJK_family_tl }
1960             }
1961         }
1962         \__xeCJK_family_use:x { \l__xeCJK_family_tl/#1 }
1963         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
1964     }
1965 }
(End definition for \__xeCJK_block_select_font:n)

```

__xeCJK_family_csname:n

```

\__xeCJK_family_nfss_csname:n 1966 \cs_new_nopar:Npn \__xeCJK_family_csname:n #1 { xeCJK/family/#1 }
\__xeCJK_family_use:x 1967 \cs_new_nopar:Npn \__xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
\__xeCJK_gset_family_nfss_cs:xx 1968 \cs_new_nopar:Npn \__xeCJK_family_use:x #1 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
1969 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_nfss_cs:xx #1#2
1970 {
1971     \prop_gput:Nxx \g__xeCJK_family_name_prop {#1} {#2}
1972     \cs_gset_protected_nopar:cpx { \__xeCJK_family_nfss_csname:n {#1} }
1973     {
1974         \exp_not:N \fontencoding { \c__xeCJK_encoding_tl }
1975         \exp_not:N \fontfamily {#2}

```

```

1976         \exp_not:N \selectfont
1977     }
1978 }
1979 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
(End definition for \__xeCJK_family_csname:n and others.)

```

\xeCJK_family_if_exist:xTF

```

1980 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
1981 {
1982     \cs_if_exist:cTF { \__xeCJK_family_nfss_csname:n {#1} }
1983     { \use_i:nn }
1984     { \cs_if_exist_use:cTF { \__xeCJK_family_csname:n {#1} } }
1985     { \prg_return_true: } { \prg_return_false: }
1986 }
(End definition for \xeCJK_family_if_exist:xTF)

```

\CJKfamily 用于切换 CJK 字体族。

```

1987 \NewDocumentCommand \CJKfamily { t+ t- m }
1988 {
1989     \xeCJK_if_blank_x:nTF {#3}
1990     {
1991         \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
1992         \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
1993     }
1994     {
1995         \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
1996         {
1997             \xeCJK_family_if_exist:xTF {#3}
1998             {
1999                 \tl_set:Nx \l_xeCJK_family_tl {#3}
2000                 \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2001                 \IfBooleanT {#1} { \__xeCJK_family_use:x {#3} }
2002             }
2003             { \__xeCJK_family_unknown_warning:x {#3} }
2004         }
2005     }
2006     \tex_ignorespaces:D
2007 }
(End definition for \CJKfamily This function is documented on page 5.)

```

\l_xeCJK_family_tl 用于保存文档当前正在使用的 CJK 字体族。

```

2008 \tl_new:N \l_xeCJK_family_tl
2009 \tl_set:Nn \l_xeCJK_family_tl { \CJKfamilydefault }
(End definition for \l_xeCJK_family_tl)

```

__xeCJK_gobble_CJKfamily:

```

2010 \cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
2011 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
2012 \DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }
(End definition for \__xeCJK_gobble_CJKfamily:)

```

\xeCJK_family_if_exist_use:x

```

\__xeCJK_family_unknown_warning:n
2013 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
2014 {
2015     \xeCJK_family_if_exist:xTF {#1}
2016     { \__xeCJK_family_use:x {#1} }
2017     { \__xeCJK_family_unknown_warning:x {#1} }
2018 }
2019 \cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:n #1
2020 {
2021     \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
2022     {

```

```

2023         \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
2024         \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
2025     }
2026 }
2027 \cs_generate_variant:Nn \__xeCJK_family_unknown_warning:n { x }
2028 \seq_new:N \g__xeCJK_unknown_family_seq
2029 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
2030 {
2031     Unknown~CJK~family~'\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
2032     Try~to~use~
2033     \str_case:x:nnn {#1}
2034     {
2035         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
2036         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
2037         \CJKttdefault { \token_to_str:N \setCJKmonofont }
2038     }
2039     { \token_to_str:N \setCJKfamilyfont {#1} }
2040     [...]{}~to~define~it.
2041 }
2042 \cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1
2043 {
2044     \str_case:x:nnn {#1}
2045     {
2046         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
2047         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
2048         \CJKttdefault { \token_to_str:N \CJKttdefault }
2049     }
2050     {#1}
2051 }
(End definition for \xeCJK_family_if_exist_use:x and \__xeCJK_family_unknown_warning:n)

```

\setCJKmainfont 设置文档的 CJK 普通字体、无衬线和等宽字体。

```

\setCJKsansfont 2052 \NewDocumentCommand \setCJKmainfont { O{} m }
\setCJKmonofont 2053 { \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2} }
2054 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
2055 \NewDocumentCommand \setCJKsansfont { O{} m }
2056 { \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2} }
2057 \NewDocumentCommand \setCJKmonofont { O{} m }
2058 { \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2} }

```

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

```

2059 \@onlypreamble \setCJKmainfont
2060 \@onlypreamble \setCJKmathfont
2061 \@onlypreamble \setCJKsansfont
2062 \@onlypreamble \setCJKmonofont
2063 \@onlypreamble \setCJKromanfont

```

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

```

\newCJKfontfamily 2064 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
\CJKfontspec 2065 { \xeCJK_set_family:xxx {#1} {#2} {#3} }
2066 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
2067 {
2068     \tl_set:Nx \l__xeCJK_tmpa_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
2069     \cs_new_protected_nopar:Npx #2 { \CJKfamily { \l__xeCJK_tmpa_tl } }
2070     \xeCJK_set_family:xxx { \l__xeCJK_tmpa_tl } {#3} {#4}
2071 }
2072 \NewDocumentCommand \CJKfontspec { O{} m }
2073 {
2074     \xeCJK_fontspec:xx {#1} {#2}
2075     \tex_ignorespaces:D
2076 }

```

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

\xeCJK_fontspec:nn

```
2077 \cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
2078 {
2079   \tl_set:Nx \l__xeCJK_tmpa_tl { \tl_to_str:n { CJKfontspec/#1/#2/id } }
2080   \tl_if_exist:cF { \l__xeCJK_tmpa_tl }
2081   {
2082     \tl_set:Nx \l__xeCJK_tmpb_tl
2083       { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
2084     \tl_gset_eq:cN { \l__xeCJK_tmpa_tl } \l__xeCJK_tmpb_tl
2085     \xeCJK_set_family:Vnn \l__xeCJK_tmpb_tl {#1} {#2}
2086   }
2087   \exp_args:Nc \CJKfamily { \l__xeCJK_tmpa_tl }
2088 }
2089 \cs_generate_variant:Nn \xeCJK_fontspec:nn { VV , xx }
(End definition for \xeCJK_fontspec:nn)
```

\defaultCJKfontfeatures
\addCJKfontfeatures

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。

```
2090 \clist_new:N \g__xeCJK_default_features_clist
2091 \NewDocumentCommand \defaultCJKfontfeatures { m }
2092 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
2093 \@onlypreamble \defaultCJKfontfeatures
2094 \NewDocumentCommand \addCJKfontfeatures { m }
2095 {
2096   \xeCJK_add_font_features:x {#1}
2097   \tex_ignorespaces:D
2098 }
2099 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
```

(End definition for \defaultCJKfontfeatures and \addCJKfontfeatures These functions are documented on page 6.)

\xeCJK_add_font_features:n

```
2100 \cs_new_protected_nopar:Npn \xeCJK_add_font_features:n #1
2101 {
2102   \prop_get:NVNTF \g__xeCJK_family_font_name_prop \l_xeCJK_family_tl \l__xeCJK_tmpa_tl
2103   {
2104     \clist_set:Nn \l__xeCJK_tmpa_clist {#1}
2105     \seq_map_function:NN
2106       \g__xeCJK_CJK_sub_class_seq \__xeCJK_parse_sub_class_features:n
2107     \prop_get:NVNT \g__xeCJK_family_font_options_prop
2108       \l_xeCJK_family_tl \l__xeCJK_tmpb_clist
2109     {
2110       \clist_concat:NNN
2111         \l__xeCJK_tmpa_clist \l__xeCJK_tmpb_clist \l__xeCJK_tmpa_clist
2112     }
2113     \xeCJK_fontspec:VV \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_tl
2114   }
2115   { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
2116 }
2117 \cs_generate_variant:Nn \xeCJK_add_font_features:n { x }
2118 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
2119 {
2120   \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
2121   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
2122 }
```

(End definition for \xeCJK_add_font_features:n)

__xeCJK_parse_sub_class_features:n

```
2123 \cs_new_protected_nopar:Npn \__xeCJK_parse_sub_class_features:n #1
2124 {
2125   \clist_if_in:NnT \l__xeCJK_tmpa_clist {#1}
2126   {
2127     \clist_remove_all:Nn \l__xeCJK_tmpa_clist {#1}
2128     \prop_get:NoNF \g__xeCJK_family_font_name_prop
```

```

2129     { \l_xeCJK_family_tl/#1 } \l__xeCJK_tmpb_tl
2130     {
2131         \prop_get:NxNF \g__xeCJK_family_font_name_prop
2132         { \CJKfamilydefault/#1 } \l__xeCJK_tmpb_tl
2133         { \tl_set_eq:NN \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
2134     }
2135     \clist_set:Nx \l__xeCJK_tmpa_clist
2136     { #1 = { [ \exp_not:V \l__xeCJK_tmpa_clist ] { \exp_not:V \l__xeCJK_tmpb_tl } } }
2137     \clist_map_break:
2138 }
2139 }
2140 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
(End definition for \__xeCJK_parse_sub_class_features:n)
在导言区结束的时候,若没有声明 CJK 字体,则给出一个警告。并在此时设置数学字体。
2141 \__xeCJK_at_end_preamble:n
2142 {
2143     \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
2144     { \__xeCJK_warning:n { no-CJKfamily } }
2145     {
2146         \CJKfamily { \CJKfamilydefault }
2147         \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
2148     }
2149 }
2150 \__xeCJK_msg_new:nn { no-CJKfamily }
2151 {
2152     It~seems~that~you~have~not~declare~a~CJKfamily.\\
2153     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
2154     \token_to_str:N \xeCJKmainfont[...]{...}\\
2155     in~the~preamble~to~declare~the~main~CJKfamily.\\
2156 }

```

5.13 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```

2157 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }
(End definition for CJKmath This function is documented on page 3.)

```

\setCJKmathfont 设置 CJK 数学字体。

```

2158 \NewDocumentCommand \setCJKmathfont { O{} m }
2159 { \xeCJK_set_family:xxx { \c__xeCJK_math_tl } {#1} {#2} }
2160 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }

```

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

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

```

2161 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
2162 {
2163     \xeCJK_family_if_exist:xF { \c__xeCJK_math_tl }
2164     { \tl_set:Nx \l__xeCJK_tmpa_tl { \c__xeCJK_math_tl } }
2165     {
2166         \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2167         { \tl_set:Nx \l__xeCJK_tmpa_tl { \CJKfamilydefault } }
2168         { \use_none:n }
2169     }
2170     \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl
2171     {
2172         \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmpb_tl }
2173         \DeclareSymbolFont \c__xeCJK_math_tl
2174         \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \mddefault \shapedefault
2175         \SetSymbolFont \c__xeCJK_math_tl { bold }
2176         \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \bfdefault \shapedefault
2177         \int_const:Nn \c__xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }

```

```

2178 \clist_concat:NNN
2179 \l__xeCJK_tmpa_clist \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
2180 \clist_concat:NNN
2181 \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_clist \c__xeCJK_FullRight_chars_clist
2182 \clist_map_inline:Nn \l__xeCJK_tmpa_clist
2183 {
2184 \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
2185 { \c_seven } { \c__xeCJK_math_fam_int }
2186 }
2187 }
2188 }
(End definition for \xeCJK_set_mathfont:)

```

\xeCJK_gset_mathcode:nnnn

```

2189 \cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
2190 {
2191 \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
2192 \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
2193 {
2194 \tex_global:D \XeTeXmathcode \l__xeCJK_tmpa_int = #3 ~ #4 \l__xeCJK_tmpa_int
2195 \int_incr:N \l__xeCJK_tmpa_int
2196 }
2197 }
(End definition for \xeCJK_gset_mathcode:nnnn)

```

5.14 抄录环境中的间距调整

\xeCJKVerbAddon \xeCJKVerbAddon 进行了比较大的调整,应该只在分组环境里使用。为了方便调整间距以利于对齐,这里只把字符分成了两类,并且在 CJK 类与边界(空格)之间也插入 \CJKecglue。当然,这样做之后,关于标点符号的禁则就没有了。

```

2198 \NewDocumentCommand \xeCJKVerbAddon { }
2199 {
2200 \int_compare:nNf \etex_currentgrouplevel:D = \c_zero
2201 {
2202 \__xeCJK_set_verb_exspace:
2203 \bool_if:NF \l__xeCJK_verb_addon_bool
2204 {
2205 \bool_set_true:N \l__xeCJK_verb_addon_bool
2206 \__xeCJK_set_char_class_eq:nn { CJK } { FullLeft }
2207 \__xeCJK_set_char_class_eq:nn { CJK } { FullRight }
2208 \__xeCJK_set_char_class_eq:nn { Default } { HalfLeft }
2209 \__xeCJK_set_char_class_eq:nn { Default } { HalfRight }
2210 \__xeCJK_set_char_class_eq:nn { Default } { NormalSpace }
2211 \xeCJKsetup { xCJKecglue = false }
2212 \cs_set_protected_nopar:Npn \CJKglue
2213 { \skip_horizontal:N \g__xeCJK_verb_exspace_skip }
2214 \cs_set_protected_nopar:Npn \CJKecglue
2215 { \skip_horizontal:N 0.5 \g__xeCJK_verb_exspace_skip }
2216 \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
2217 \cs_set_eq:NN \xeCJK_ignore_spaces:w \CJKecglue
2218 \skip_set:Nn \tex_rightskip:D { \c_zero_dim plus 1 fil }
2219 }
2220 }
2221 }
2222 \bool_new:N \l__xeCJK_verb_addon_bool
2223 \cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon

```

(End definition for \xeCJKVerbAddon This function is documented on page 5.)

总是在 \LaTeX 的抄录环境(包括 \verb)里使用 \xeCJKVerbAddon,以保证代码的对齐。

```

2224 \__xeCJK_after_preamble:n
2225 { \xeCJK_cs_add:Nnn \verbatim@font { } { \xeCJKVerbAddon } }

```


`__xeCJK_set_verb_exspace:` 在抄录环境中, CJK 文字之间的间距为当前西文字体两个空格的宽度与当前字体大小之差, 而与西文和空格的间距为 CJK 文字之间的间距的一半。这里的问题是, `\g__xeCJK_verb_exspace_skip` 有可能为负么? 现在的处理是保证它非负。如果真的出现负间距的情况, 也可以像下面对后备可视空格符号一样, 在 `\xeCJK_select_font:` 里对 CJK 文字的字体大小做调整, 并让间距为零。

```

2226 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
2227 {
2228   \tl_if_exist:cTF { xeCJK/verb/\curr@fontshape/\f@size }
2229   {
2230     \skip_gset:Nn \g__xeCJK_verb_exspace_skip
2231     { \tl_use:c { xeCJK/verb/\curr@fontshape/\f@size } }
2232   }
2233   {
2234     \skip_gset:Nn \g__xeCJK_verb_exspace_skip
2235     {
2236       \dim_max:nn
2237       { \c_zero_dim }
2238       { \c_two \tex_fontdimen:D \c_two \tex_font:D - \f@size pt }
2239     }
2240     \tl_gset:cV { xeCJK/verb/\curr@fontshape/\f@size } \g__xeCJK_verb_exspace_skip
2241   }
2242 }
2243 \skip_new:N \g__xeCJK_verb_exspace_skip
(End definition for \__xeCJK_set_verb_exspace:)

```

`\xeCJK_visible_space_fallback:` `fontspec` 使用 `lmtt` 字体中的可视空格符号(U+2423)作为当前字体中相应符号的后备。但是 `lmtt` 的字体大小未必与当前字体匹配。因此, 我们在这里做一些调整, 以保证使用后备可视空格符号时, 也能保证对齐。这里加入 `\scan_stop:` 的目的是强制发生状态转移。这样当空格出现在 CJK 文字后面时, 使字体回到西文, 以便让 `fontspec` 在当前西文字体而不是在 CJK 字体中检查有没有 U+2423。

```

2244 \cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
2245 { {
2246   \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
2247   { \xeCJK_set_visible_space_font: }
2248   \textvisiblespace
2249 } }
2250 \AtEndOfPackage
2251 {
2252   \xeCJK_cs_add:Nnn \fontspec_visible_space: { \scan_stop: } { }
2253   \cs_gset_eq:NN \fontspec_visible_space_fallback: \xeCJK_visible_space_fallback:
2254 }
(End definition for \xeCJK_visible_space_fallback:)

```

`\xeCJK_set_visible_space_font:` 当前字体空格的宽度与后备字体 `lmtt` 不一样时, 就对 `\textvisiblespace` 的字体尺寸按相应的比例放缩。

```

2255 \cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
2256 {
2257   \tl_set:Nx \l__xeCJK_current_coor_tl { xeCJK/space/\curr@fontshape/\f@size }
2258   \dim_set:Nn \l__xeCJK_tmp_dim { \tex_fontdimen:D \c_two \tex_font:D }
2259   \fontencoding { \g_fontspec_encoding_tl }
2260   \fontfamily { lmtt }
2261   \selectfont
2262   \dim_compare:nNnF \l__xeCJK_tmp_dim = { \tex_fontdimen:D \c_two \tex_font:D }
2263   {
2264     \fontsize
2265     {
2266       \dim_eval:n
2267       {
2268         \f@size pt * \dim_ratio:nn
2269         { \l__xeCJK_tmp_dim } { \tex_fontdimen:D \c_two \tex_font:D }
2270       }
2271     }
2272   }

```

```

2272         { \f@baselineskip }
2273     \selectfont
2274 }
2275 \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2276 }
(End definition for \xeCJK_set_visible_space_font:)

```

5.15 xeCJK 其它选项

LocalConfig 声明载入本地配置文件的选项。

```

2277 \keys_define:nn { xeCJK / options }
2278 {
2279     LocalConfig .choice: ,
2280     LocalConfig / false .code:n =
2281         { \bool_gset_false:N \g__xeCJK_config_bool } ,
2282     LocalConfig / true .code:n =
2283         {
2284             \bool_gset_true:N \g__xeCJK_config_bool
2285             \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
2286         } ,
2287     LocalConfig / unknown .code:n =
2288         {
2289             \bool_gset_true:N \g__xeCJK_config_bool
2290             \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
2291         } ,
2292     LocalConfig .default:n = { true }
2293 }
2294 \tl_new:N \g__xeCJK_config_name_tl
2295 \bool_new:N \g__xeCJK_config_bool
(End definition for LocalConfig This function is documented on page 2.)

```

CJKNumber 是否启用 CJKnumb 宏包和首行是否缩进。
indentfirst

```

2296 \keys_define:nn { xeCJK / options }
2297 {
2298     CJKNumber .bool_gset:N = \g__xeCJK_number_bool ,
2299     indentfirst .bool_gset:N = \g__xeCJK_indent_bool ,
2300     normalindentfirst .meta:n = { indentfirst = false }
2301 }
(End definition for CJKNumber and indentfirst These functions are documented on page 3.)

```

quiet 将调用 xeCJK 时使用的未知的选项传递给 fontspec 宏包。对 fontspec 的 quiet 和 silent 选项进行
silent 修改,使其适用于 xeCJK。

```

2302 \keys_define:nn { xeCJK / options }
2303 {
2304     quiet .code:n =
2305         {
2306             \msg_redirect_module:nnn { xeCJK } { warning } { info }
2307             \msg_redirect_module:nnn { xeCJK } { info } { none }
2308             \xeCJK_if_package_loaded:nF { fontspec }
2309             { \PassOptionsToPackage { quiet } { fontspec } }
2310         } ,
2311     silent .code:n =
2312         {
2313             \msg_redirect_module:nnn { xeCJK } { warning } { none }
2314             \msg_redirect_module:nnn { xeCJK } { info } { none }
2315             \xeCJK_if_package_loaded:nF { fontspec }
2316             { \PassOptionsToPackage { silent } { fontspec } }
2317         } ,
2318     unknown .code:n =
2319         {
2320             \xeCJK_if_package_loaded:nTF { fontspec }
2321             { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }

```

```

2322     { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2323   }
2324 }
2325 \__xeCJK_msg_new:nn { key-unknown }
2326 {
2327   Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~'#1'.\\
2328   The~key~'#1'~is~being~ignored.
2329 }

```

(End definition for quiet and silent These functions are documented on page ??.)

5.16 xeCJK 初始化设置

\CJKsymbol
\CJKpunctsymbol

```

2330 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2331 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}

```

(End definition for \CJKsymbol and \CJKpunctsymbol)

xeCJK 宏包的初始化设置。

```

2332 \keys_set:nn { xeCJK / options }
2333 {
2334   CJKglue      = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
2335   CJKecglue    = \c_space_token ,
2336   xCJKecglue   = false ,
2337   CheckSingle  = false ,
2338   CJKspace     = false ,
2339   CJKmath      = false ,
2340   CJKnumber    = false ,
2341   xeCJKactive  = true ,
2342   LocalConfig  = true ,
2343   indentfirst  = true ,
2344   EmboldenFactor = 4 ,
2345   SlantFactor  = 0.167 ,
2346   PunctStyle   = quanjiao ,
2347   KaiMingPunct = { ^^^^3002 ^^^^ff0e ^^^^ff1f ^^^^ff01 } ,
2348   LongPunct    = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^2025 ^^^^2026 } ,
2349   MiddlePunct  = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
2350   AllowBreakBetweenPuncts = false
2351 }
2352 \defaultCJKfontfeatures { Script = CJK }

```

执行宏包选项,并载入 fontspec 宏包。

```

2353 \ProcessKeysOptions { xeCJK / options }
2354 \RequirePackage { fontspec } [ 2011/09/13 ]

```

\c__xeCJK_encoding_tl 保存 fontspec 声明字体时使用的字体编码。

```

2355 \tl_const:Nx \c__xeCJK_encoding_tl { \g_fontspec_encoding_tl }

```

(End definition for \c__xeCJK_encoding_tl)

章节标题后面的首个段落的首行是否缩进。

```

2356 \bool_if:NT \g__xeCJK_indent_bool { \RequirePackage { indentfirst } }

```

对不能通过 \xeCJKsetup 设置的选项给出警告。

```

2357 \keys_define:nn { xeCJK / options }
2358 {
2359   LocalConfig .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2360   CJKnumber   .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2361   indentfirst .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } }
2362 }
2363 \__xeCJK_msg_new:nn { option-invalid }
2364 {
2365   The~'#1'~option~only~can~be~set~in~the~optional~argument~to~the\\
2366   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
2367   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
2368 }

```

```

\CJKrmdefault
\JKsfdefault 2369 \tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
\JKttdefault 2370 \tl_if_exist:NF \JKsfdefault { \tl_gset:Nn \JKsfdefault { sf } }
\CJKfamilydefault 2371 \tl_if_exist:NF \JKttdefault { \tl_gset:Nn \JKttdefault { tt } }
2372 \tl_if_exist:NF \CJKfamilydefault { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
2373 \tl_new:c { \__xeCJK_family_csname:n { \CJKfamilydefault } }
(End definition for \CJKrmdefault and others. These variables are documented on page 6.)

```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```

2374 \NewDocumentCommand \xeCJKsetup { m }
2375 {
2376   \keys_set:nn { xeCJK / options } {#1}
2377   \tex_ignorespaces:D
2378 }
(End definition for \xeCJKsetup This function is documented on page 2.)

```

\xeCJKsetemboldenfactor
\xeCJKsetslantfactor

```

2379 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
2380 { \xeCJKsetup { EmboldenFactor = {#1} } }
2381 \NewDocumentCommand \xeCJKsetslantfactor { m }
2382 { \xeCJKsetup { SlantFactor = {#1} } }
(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor)

```

\punctstyle
\xeCJKplainchr

```

2383 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2384 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }
(End definition for \punctstyle and \xeCJKplainchr)

```

\CJKsetecglue

```

2385 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2386 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
(End definition for \CJKsetecglue)

```

\CJKspace
\CJKnospace

```

2387 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
2388 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }
(End definition for \CJKspace and \CJKnospace)

```

\xeCJKallowbreakbetweenpuncts
\xeCJKnobreakbetweenpuncts

```

2389 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
2390 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
2391 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
2392 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }
(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts)

```

5.17 兼容性修补

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

```

2393 \xeCJK_cs_add:Nnn \rmfamily { \CJKfamily { \CJKrmdefault } } { }
2394 \xeCJK_cs_add:Nnn \sffamily { \CJKfamily { \JKsfdefault } } { }
2395 \xeCJK_cs_add:Nnn \ttfamily { \CJKfamily { \JKttdefault } } { }
2396 \xeCJK_cs_add:Nnn \normalfont { \CJKfamily { \CJKfamilydefault } } { }
2397 \cs_set_eq:NN \reset@font \normalfont

```

因为我们在上面把 \CJKfamily 加在了 \rmfamily 里面, 而 fontspec 在调用字体时, 某些选项要使用它。为减少影响, 下面使 \CJKfamily 在通过 fontspec 来实际调用字体时无效。

```

2398 \xeCJK_cs_add:Nnn \fontspec_init: { \__xeCJK_gobble_CJKfamily: } { }

```

\xeCJK@fix@penalty 对 L^AT_EX 2_ε 内核中的 \fix@penalty 被用于诸如 \textit 之类的文档字体转换命令的定义之中。这里对它进行补丁的目的是修复其中的倾斜校正,并使得这些文档命令与紧随其后的汉字之间可以正确的插入 \CJKecglue 或者忽略其中的空格。例如 这是₁\emph{强调}₂文本,第二个空格可以被忽略掉。如果使用 xCJKecglue 选项,第一个空格也可以被省略。事实上,在 \sw@slant 的定义中, \@@italiccorr 前面的 \lastskip 和 \lastpenalty 有四种情况,这里只对它们都为零的情况进行处理。

```
2400 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
2401 \xeCJK_cs_patch:Nnn \xeCJK@fix@penalty \@@italiccorr \xeCJK@patch@italiccorr
2402 \xeCJK_cs_patch:Nnn \sw@slant \fix@penalty \xeCJK@fix@penalty
(End definition for \xeCJK@fix@penalty)
```

\xeCJK@patch@italiccorr 修复倾斜校正,并处理汉字后面的空格。

```
2403 \cs_new_protected_nopar:Npn \xeCJK@patch@italiccorr
2404 {
2405   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
2406   {
2407     \xeCJK_if_last_node:nTF { default }
2408     {
2409       \tex_unkern:D \tex_unkern:D \@@italiccorr
2410       { \xeCJK_make_node:n { default } }
2411     }
2412     {
2413       \xeCJK_if_last_node:nTF { CJK }
2414       {
2415         \tex_unkern:D \tex_unkern:D \@@italiccorr
2416         { \xeCJK_make_node:n { CJK } } \use:n
2417       }
2418       {
2419         \xeCJK_if_last_node:nTF { CJK-space }
2420         {
2421           \tex_unkern:D \tex_unkern:D \@@italiccorr
2422           { \xeCJK_make_node:n { CJK-space } } \use:n
2423         }
2424         { \@@italiccorr \use_none:n }
2425       }
2426     }
2427   }
```

\xeCJK_ignore_spaces:w 里面用到 peek 函数来判断后面是不是空格,而此时它后面还有 4 个 \fi 或者 \else...\fi 没有被展开,将影响 peek 函数的判断。因此我们需要用 $2^4 - 1 = 15$ 个 \exp_after:wN 来展开它们。显然,这里用 \exp_last_unbraced:Nf 会比较方便,但是它会吃掉 \textit{...} 等后面原来存在的空格作为完全展开的结束。要正确使用它还需要另外的处理(使用 \exp_stop_f:)。

```
2426 {
2427   \exp_after:wN \exp_after:wN \exp_after:wN
2428   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2429   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2430   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2431   \xeCJK_ignore_spaces:w
2432 }
2433 }
2434 }
2435 { \@@italiccorr }
2436 }
(End definition for \xeCJK@patch@italiccorr)
```

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

```
\c__xeCJK_others_class_int
2438 \__xeCJK_after_end_preamble:n
2439 {
2440   \int_compare:nNnF
2441   { \c_three + ( \seq_count:N \g__xeCJK_class_seq ) } = \xe@alloc@intercharclass
2442   {
2443     \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
2444     {
```

```

2445         \seq_if_in:NnF \g__xeCJK_class_seq {#1}
2446         { \__xeCJK_set_others_toks:n {#1} }
2447     }
2448 }
2449 }
2450 \cs_new_protected_nopar:Npn \__xeCJK_set_others_toks:n #1
2451 {
2452     \int_set:Nn \c__xeCJK_Others_class_int {#1}
2453     \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
2454     {
2455         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2456         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2457         \xeCJK_app_inter_class_toks:nnx {##1} { Others }
2458         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
2459         \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
2460         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
2461         \xeCJK_if_blank_x:nT
2462         { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2463         {
2464             \xeCJK_copy_inter_class_toks:nnnn
2465             { Others } { Boundary } { Default } { Boundary }
2466         }
2467         \xeCJK_if_blank_x:nT
2468         { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2469         {
2470             \xeCJK_copy_inter_class_toks:nnnn
2471             { Boundary } { Others } { Boundary } { Default }
2472         }
2473     }
2474 }
2475 \cs_new_eq:NN \c__xeCJK_Others_class_int \l__xeCJK_tmpa_int
(End definition for \__xeCJK_set_others_toks:n and \c__xeCJK_Others_class_int)
    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。
2476 \__xeCJK_after_preamble:n
2477 {
2478     \tl_map_inline:nn
2479     {
2480         \textellipsis \textemdash \textperiodcentered \textcentereddot
2481         \textquoteleft \textquoteright \textquotedblleft \textquotedblright
2482     }
2483     {
2484         \xeCJK_cs_add:Nnn #1
2485         { \__xeCJK_group_begin: \makexeCJKinactive } { \__xeCJK_group_end: }
2486     }
2487     \xeCJK_cs_add:Nnn \tipaencoding { \makexeCJKinactive } { }
2488     \cs_new_eq:NN \__xeCJK_aux_r:n \r
2489     \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \__xeCJK_aux_r:n {#1} } }
2490     \xeCJK_if_package_loaded:nT { pifont }
2491     { \xeCJK_cs_add:Nnn \Pifont { \makexeCJKinactive } { } }
2492 }
2493 \cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
2494 \cs_new_eq:NN \__xeCJK_group_end: \group_end:
    简单处理与 hyperref 宏包的兼容问题。
2495 \__xeCJK_after_end_preamble:n
2496 {
2497     \xeCJK_if_package_loaded:nT { hyperref }
2498     {
2499         \tl_gput_right:Nn \pdfstringdefPreHook
2500         {
2501             \__xeCJK_gobble_CJKfamily:

```

```

2502         \xeCJK_cs_clear:N \makexeCJKinactive
2503         \xeCJK_cs_clear:N \__xeCJK_group_begin:
2504         \xeCJK_cs_clear:N \__xeCJK_group_end:
2505     }
2506 }
2507 }

```

当探测到 `cprotect` 宏包被引入时,则取消 `\cprotect` 宏的 `\outer` 定义。

```

2508 \__xeCJK_after_end_preamble:n
2509 {
2510     \bool_if:nT
2511     { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
2512     { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
2513 }

```

禁止在 `xeCJK` 宏包后再载入 `CJK` 宏包,并使得 `CJKnumb` 和 `CJKfntef` 宏包可以使用。

```

2514 \tl_set:cn { ver@CJK.sty } { 9999/99/99 }
2515 \tl_set_eq:cc { ver@CJKpunct.sty } { ver@CJK.sty }

```

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

```

2516 \cs_if_exist:NF \CJK@ifundefined
2517 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2518 \NewDocumentCommand \xeCJKcaption { o m }
2519 {
2520     \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
2521     \use:x
2522     {
2523         \char_set_catcode_letter:n { 64 }
2524         \file_input:n { #2.cpx }
2525         \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
2526     }
2527     \XeTeXdefaultencoding "UTF-8"
2528 }

```

(End definition for \xeCJKcaption)

`\xeCJK_hook_for_ulem:` 对 `ulem` 宏包打补丁,以支持 `CJKfntef` 宏包。目前的问题是,在下划线状态下,相邻两个标点之间的空白不能压缩。

```

2529 \cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
2530 {
2531     \bool_if:NF \l__xeCJK_ulem_hook_used_bool
2532     {
2533         \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
2534         \xeCJK_cs_clear:N \__xeCJK_CJK_FullLeft_glue:
2535         \cs_set_protected_nopar:Npn \__xeCJK_FullLeft_ulem_hook: { { \CJKglue } }
2536         \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_ulem_CJKsymbol:N
2537         \xeCJK_swap_cs:NN \CJKpunctsymbol \__xeCJK_ulem_CJKpunctsymbol:N
2538     }
2539 }
2540 \bool_new:N \l__xeCJK_ulem_hook_used_bool
2541 \__xeCJK_after_preamble:n
2542 {
2543     \xeCJK_if_package_loaded:nT { ulem }
2544     {
2545         \xeCJK_cs_gclear:N \__xeCJK_FullLeft_ulem_hook:
2546         \xeCJK_pre_inter_class_toks:nnn { Boundary } { FullLeft }
2547         { \__xeCJK_FullLeft_ulem_hook: }
2548         \cs_if_exist:NTF \XeTeX@CJKfntef@hook
2549         {
2550             \xeCJK_cs_gclear:N \CJKf@global
2551             \AtBeginEnvironment
2552             { CJKfilltwosides } { \cs_set_eq:NN \CJKglue \tex_hfil:D }
2553             \cs_set_protected_nopar:Npn \XeTeX@CJKfntef@hook
2554             { \xeCJK_select_font: \makexeCJKinactive }

```

```

2555     }
2556     { \RequirePackage { CJKulem } }
2557     \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }
2558   }
2559 }
(End definition for \xeCJK_hook_for_ulem:)

```

`__xeCJK_ulem_CJKsymbol:N` 这两个宏定义里面的分组记号都是必须的, 否则会出现分组不匹配的情况。

```

\__xeCJK_ulem_CJKpunctsymbol:N
2560 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJKsymbol:N #1
2561 {
2562   { \__xeCJK_ulem_CJKsymbol:N {#1} }
2563   { \xeCJK_make_node:n { CJK } } \xeCJK_ignore_spaces:w
2564 }
2565 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJKpunctsymbol:N #1
2566 { { \__xeCJK_ulem_CJKpunctsymbol:N {#1} } \tex_ignorespaces:D }
(End definition for \__xeCJK_ulem_CJKsymbol:N and \__xeCJK_ulem_CJKpunctsymbol:N)

```

为使用 `CJKnumb` 宏包而作一些处理。另外 `CJKnumb` 使用的是传统汉字“萬”和“億”, 我们在这里把它们修正为简体字。

```

2567 \bool_if:NT \g__xeCJK_number_bool
2568 {
2569   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2570   \cs_set_protected:Npn \CJKaddEncHook #1#2
2571     { \cs_set:cpn { __xeCJK_enc_#1_hook: } {#2} }
2572   \cs_set_protected_nopar:Npn \Unicode #1#2
2573     { \tex_char:D \etex_numexpr:D (#1) * \c_two_hundred_fifty_six + (#2) \scan_stop: }
2574   \RequirePackage { CJKnumb }
2575   \cs_if_exist_use:c { __xeCJK_enc_ \CJK@UnicodeEnc _hook: }
2576   \tl_set:Nn \CJK@tenthousand { \Unicode { 78 } { 7 } }
2577   \tl_set:Nn \CJK@hundredmillion { \Unicode { 78 } { 191 } }
2578 }

```

最后引入本地配置文件。

```

2579 \bool_if:NT \g__xeCJK_config_bool
2580 {
2581   \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
2582   \@onefilewithoptions
2583     { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
2584 }
2585 \</package>

```

5.18 xeCJK.cfg

```

2586 \*config>

```

预设的配置文件 `xeCJK.cfg` 为一个空文件。可以在里面增加设置, 然后保存到本地目录下面。

```

2587
2588 \</config>

```


版本历史

v3.1.0

General: 使用 `xtemplate` 宏包的机制来组织标点符号的处理。 33
删除多余的 `default-itcorr` 结点。 23
取消 `\cprotect` 的外部宏限制。 63
改用 `\indentfirst` 宏包处理缩进的问题。 59
放弃使用放缩字体大小的方式,而只采用调整间距的方式与西文等宽字体对齐。并且只适用于与抄录环境下。 . 56
放弃对 `\outer` 宏的特殊处理。 1
`_xeCJK_switch_font:nn`: 改进 `_xeCJK_switch_font:nn` 的定义。 51
`\c_xeCJK_space_skip_tl`: 字间空格考虑 `\spaceskip` 不为零的情况。 13

LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。 58
`\xeCJK@fix@penalty`: 采用通过不修改原语 `\` 的方式对修复倾斜校正。 61
`\xeCJK_fallback_loop:Nn`: 调整备用字体的循环方式。 . . 42
`\xeCJK_glyph_if_exist:N`: 改进 `fontspec` 宏包中定义的 `\font_glyph_if_exist:NnTF`。 13
`\xeCJK_hook_for_ulem::`: 简化对 `ulem` 宏包的兼容补丁。 . 63
`\xeCJK_visible_space_fallback::`: 调整 `fontspec` 的后备可视空格符号,以便于使用时对齐。 57
`\xeCJKVerbAddon`: 新增 `\xeCJKVerbAddon` 用于抄录环境中的间距调整。 56

代码索引

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

Symbols

`\` 417, 418
`\@italiccorr` 2401, 2409, 2415, 2421, 2424, 2435
`\ifpackagelater` 19, 22
`\onefilewithoptions` 2582
`\onlypreamble`
. 824, 1407, 1414, 2059, 2060, 2061, 2062, 2063, 2093
`\pgkextension` 139
`\[` 772
`\` 5, 6, 7, 13,
14, 15, 16, 860, 912, 913, 1390, 1391, 1392, 1404, 1405,
1489, 2031, 2120, 2152, 2153, 2154, 2155, 2327, 2365, 2366
`\]` 772
`_xeCJK_CJK_FullLeft_glue:` 636, 641, 2534
`_xeCJK_FullLeft_ulem_hook:` 2535, 2545, 2547
`_xeCJK_add_fake_bold:n` 1738, 1738, 1787, 1806
`_xeCJK_add_fake_slant:n` 1738, 1743, 1798
`_xeCJK_add_font:nV` 1764, 1766
`_xeCJK_add_font:nn` 1732, 1732, 1736
`_xeCJK_add_font:nv` 1754
`_xeCJK_add_font:nx` 1814
`_xeCJK_add_font_if_new:nn` 1732, 1734, 1737, 1808
`_xeCJK_add_font_if_new:nv` 1758
`_xeCJK_add_font_if_new:nx` 1816
`_xeCJK_add_special_punct:nn` 936, 939, 942, 946, 963
`_xeCJK_after_end_preamble:n` 47, 54, 919, 2438, 2495, 2508
`_xeCJK_after_preamble:n` 47, 52, 2224, 2476, 2541
`_xeCJK_at_end_preamble:n` 47, 50, 2141
`_xeCJK_aux:r:n` 2488, 2489
`_xeCJK_block_select_font:n` 1944, 1948, 1948
`_xeCJK_calc_kerning_margin:NN` 1199, 1219, 1219
`_xeCJK_calc_kerning_margin:NNN` 1232, 1241, 1286, 1286
`_xeCJK_check_family:V` 1683, 1868
`_xeCJK_check_family:n` 1720, 1720, 1729
`_xeCJK_check_num_range:nnNN` 284, 296, 296, 2191
`_xeCJK_check_single_save:N`
. 709, 718, 733, 739, 751, 759, 764, 783, 791, 798
`_xeCJK_clear_Boundary_and_CJK_toks:n` 365, 366
`_xeCJK_copy_family:nn` 1895, 1895, 1914
`_xeCJK_copy_family:xx` 1956

`_xeCJK_copy_sub_family:n` 1858, 1866
`_xeCJK_error:n` 37, 38, 918
`_xeCJK_error:nx` 37, 39, 854, 1384, 1412, 2321
`_xeCJK_fallback_save_CJKsymbol:N`
. 1452, 1461, 1468, 1477, 1482
`_xeCJK_family_csname:n`
. 1709, 1882, 1958, 1966, 1966, 1984, 2373
`_xeCJK_family_nfss_csname:n`
. 1725, 1910, 1911, 1966, 1967, 1968, 1972, 1982
`_xeCJK_family_unknown_warning:n` 2013, 2019, 2027
`_xeCJK_family_unknown_warning:x` 2003, 2017
`_xeCJK_family_use:x` 1923, 1932, 1962, 1966, 1968, 2001, 2016
`_xeCJK_font_csname:n` 1916, 1917, 1929, 1931, 1950, 1953
`_xeCJK_get_map_font:n` 1814, 1816, 1819, 1819, 1823
`_xeCJK_get_sub_features:Vn` 1509
`_xeCJK_get_sub_features:nn` 1572, 1580, 1580, 1607
`_xeCJK_get_sub_features:wn` 1580, 1586, 1600
`_xeCJK_gobble_CJKfamily:` 1648, 1712, 2010, 2010, 2398, 2501
`_xeCJK_gobble_CJKfamily:wn` 2011, 2012
`_xeCJK_group_begin:` 2485, 2493, 2503
`_xeCJK_group_end:` 2485, 2494, 2504
`_xeCJK_gset_family_cs:x` 1684, 1707, 1707
`_xeCJK_gset_family_nfss_cs:xx` 1716, 1889, 1966, 1969
`_xeCJK_gset_punct_dim:nnnx` 996, 1010, 1341
`_xeCJK_gset_punct_dim:nnx` 996,
1008, 1112, 1187, 1307, 1308, 1309, 1314, 1324, 1326, 1362
`_xeCJK_gset_punct_dim:nnx` 996, 1006, 1333, 1335
`_xeCJK_if_font_select:n` 1772
`_xeCJK_if_font_select:nF` 1757, 1763, 1765, 1815
`_xeCJK_if_font_select:nTF` 1752, 1772
`_xeCJK_if_font_select_p:n` 1772, 1794, 1796, 1804
`_xeCJK_if_it_or_sl:n` 1777
`_xeCJK_if_it_or_sl:nT` 1800
`_xeCJK_if_it_or_sl:nTF` 1755, 1772
`_xeCJK_if_it_or_sl_p:n` 1772, 1792, 1803
`_xeCJK_if_map_font_select:n` 1821
`_xeCJK_if_map_font_select:nT` 1812
`_xeCJK_if_map_font_select:nTF` 1821
`_xeCJK_if_map_font_select_p:n` 1821
`_xeCJK_info:nx` 37, 43

C

_xeCJK_Boundary_class_int	158	\CJKecglue	403, 406, 411, 426, 447, 463, 499, 503, 504, 510, 520, 521, 522, 560, 2214, 2216, 2217
_xeCJK_CJK_chars_clist	169, 195, 329, 2179	xCJKecglue	3
_xeCJK_CJK_class_int	155	\CJKf@global	2550
_xeCJK_config_ext_tl	2581, 2583	\CJKfamily	5, 1987, 1987, 2011, 2069, 2087, 2146, 2393, 2394, 2395, 2396
_xeCJK_Default_class_int	154	\CJKfamilydefault	6, 1958, 1959, 2009, 2132, 2146, 2166, 2167, 2369, 2372, 2373, 2396
_xeCJK_encoding_tl	1974, 2174, 2176, 2355, 2355	\CJKfixedspacing	2223
_xeCJK_FullLeft_chars_clist	169, 174, 324, 2179	\CJKfontspec	5, 2064, 2072
_xeCJK_FullLeft_class_int	156	\CJKglue	440, 470, 497, 498, 566, 580, 582, 641, 646, 2212, 2535, 2552
_xeCJK_FullRight_chars_clist	169, 181, 325, 2181	\CJKmath	2157
_xeCJK_FullRight_class_int	157	\CJKnospace	2387, 2388
_xeCJK_HalfLeft_chars_clist	169, 169, 322	\CJKnumber	2296
_xeCJK_HalfRight_chars_clist	169, 171, 323	\CJKpunctsymbol	639, 656, 665, 675, 684, 693, 2330, 2331, 2537
_xeCJK_left_tl	620, 621, 634, 636, 638, 652, 661, 679, 925, 925, 1087, 1203, 1216, 1277, 1282, 1293, 1308, 1324, 1330	\CJKrmdefault	6, 2035, 2046, 2053, 2369, 2369, 2372, 2393
_xeCJK_math_family_tl	2172, 2174, 2176	\CJKsetecglue	2385, 2385, 2386
_xeCJK_math_tl	2159, 2160, 2163, 2164, 2173, 2175, 2177	\CJKsfdefault	6, 2036, 2047, 2056, 2369, 2370, 2394
_xeCJK_NormalSpace_chars_clist	169, 173, 330	\CJKspace	528, 2387, 2387
_xeCJK_Others_class_int	2438, 2452, 2475	\CJKsymbol	383, 434, 566, 581, 582, 774, 782, 799, 888, 889, 1452, 1453, 1461, 2330, 2330, 2536
_xeCJK_package_ext_tl	135, 139	\CJKttdefault	6, 2037, 2048, 2058, 2369, 2371, 2395
_xeCJK_punct_style_plain_tl	1305, 1361, 1379, 1387	\clist_clear:c	1654
_xeCJK_right_tl	622, 628, 629, 630, 644, 651, 670, 688, 925, 926, 1087, 1202, 1213, 1278, 1283, 1296, 1309, 1326, 1331	\clist_clear:N	1583, 1590, 1656, 1694
_catcode_other_token	723, 738, 750	\clist_concat:NNN	1676, 2110, 2178, 2180
_empty_tl	126, 1586, 1603, 1604	\clist_const:Nn	169, 171, 173, 174, 181, 195
_four	114, 2443	\clist_count:N	1495
_group_begin_token	368	\clist_gclear_new:c	162, 236
_group_end_token	369	\clist_gconcat:ccN	262
_math_toggle_token	546, 779, 781, 782, 783	\clist_gconcat:NNN	275
_max_dim	1016, 1022, 1033, 1036, 1038, 1040, 1117, 1142, 1163, 1167, 1176, 1231, 1258, 1266	\clist_gput_right:Nx	807
_one	151, 155, 635, 1325, 1495, 2083, 2443	\clist_gremove_all:Nn	813
_one_fp	1017, 1023, 1041	\clist_gremove_duplicates:N	808
_seven	2185	\clist_gset:Nn	947, 2092
_space_tl	129	\clist_if_empty:cF	1830
_space_token	403, 470, 471, 515, 542, 751, 758, 759, 2335	\clist_if_in:NnT	2125
_ten	445, 460	\clist_if_in:NnTF	797
_ten_thousand	46	\clist_map_break:	2137
_three	113, 157, 1327, 2441	\clist_map_function:nN	245
_two	112, 156, 609, 1123, 1147, 2238, 2258, 2262, 2269	\clist_map_inline:cn	311
_two_hundred_fifty_five	158	\clist_map_inline:Nn	264, 950, 988, 1506, 2182
_two_hundred_fifty_six	2573	\clist_map_inline:nn	370, 372, 384, 387, 413, 567, 592, 594, 816, 846, 904
_xeCJK_math_fam_int	2177, 2185	\clist_map_inline:xn	812
_xeCJK_space_skip_tl	108, 108, 394, 452	\clist_new:N	35, 36, 946, 2090
_zero	45, 126, 129, 152, 154, 2200, 2405	\clist_put_left:cn	1510
_zero_dim	601, 602, 1030, 1042, 1307, 1308, 1309, 1314, 1348, 1362, 2218, 2237, 2334	\clist_put_left:cx	1740, 1745
_zero_fp	1037, 1039, 1166, 1234, 1261	\clist_put_right:Nn	1698, 1699, 1837
_zero_skip	110	\clist_put_right:Nx	1832
\char_set_catcode:nn	2525	\clist_remove_all:Nn	1846, 1878, 2127
\char_set_catcode_ignore:n	153	\clist_set:Nn	261, 1602, 1674, 2104
\char_set_catcode_letter:n	2523	\clist_set:Nx	1494, 2135
\char_set_catcode_other:n	287	\clist_set_eq:cN	1597
\char_value_catcode:n	2525	\cs:w	101, 2512
\CheckSingle	695	\cs_end:	101, 992, 2512
\CJK	154	\cs_generate_variant:Nn	244, 275, 276, 319, 336, 346, 352, 816, 857, 874, 987, 1354, 1369, 1491, 1518, 1579, 1607, 1608, 1609, 1689, 1702, 1729, 1736, 1737, 1783, 1914, 1979, 2027, 2089, 2117, 2140
\CJK@hundredmillion	2577	\cs_gset_eq:cc	1909
\CJK@ifundefined	2516, 2517	\cs_gset_eq:NN	100, 2253
\CJK@tenthousand	2576		
\CJK@UnicodeEnc	2569, 2575		
\CJKaddEncHook	2570		

<code>\cs_gset_protected_nopar:cpx</code>	1709, 1882, 1972
<code>\cs_if_exist:cTF</code>	1958, 1982
<code>\cs_if_exist:Nf</code>	561, 916, 2516
<code>\cs_if_exist:NfT</code>	2548
<code>\cs_if_exist_p:N</code>	2511
<code>\cs_if_exist_use:c</code>	2575
<code>\cs_if_exist_use:cF</code>	1920, 1929, 1950, 2246
<code>\cs_if_exist_use:cTF</code>	1093, 1984
<code>\cs_if_free:NfT</code>	2517
<code>\cs_new:Npn</code>	63, 65
<code>\cs_new_eq:NN</code>	89, 90, 368, 369, 564, 1936, 2054, 2099, 2223, 2386, 2400, 2475, 2488, 2493, 2494
<code>\cs_new_nopar:Npn</code> ..	59, 246, 247, 337, 948, 949, 996, 998, 1000, 1002, 1004, 1161, 1174, 1207, 1219, 1256, 1272, 1286, 1355, 1819, 1917, 1966, 1967, 1968, 2042, 2330, 2331
<code>\cs_new_protected:Npn</code>	50, 52, 54, 71, 82, 91, 140, 142
<code>\cs_new_protected_nopar:Npn</code>	37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 61, 98, 118, 159, 239, 259, 282, 296, 308, 334, 339, 341, 347, 353, 358, 364, 366, 436, 482, 488, 540, 566, 597, 604, 612, 615, 626, 632, 641, 642, 658, 667, 677, 686, 721, 736, 767, 777, 795, 844, 864, 875, 952, 963, 974, 1006, 1008, 1010, 1081, 1185, 1301, 1322, 1357, 1465, 1471, 1499, 1549, 1580, 1600, 1646, 1669, 1690, 1703, 1705, 1707, 1720, 1732, 1734, 1738, 1743, 1748, 1750, 1784, 1810, 1826, 1842, 1853, 1866, 1895, 1918, 1927, 1937, 1948, 1969, 2010, 2013, 2019, 2077, 2100, 2123, 2161, 2189, 2226, 2244, 2255, 2403, 2450, 2529, 2560, 2565
<code>\cs_new_protected_nopar:Npx</code>	2069
<code>\cs_set:cpn</code>	2571
<code>\cs_set_eq:NN</code>	93, 94, 95, 287, 288, 510, 515, 522, 709, 710, 718, 931, 933, 1452, 1453, 1461, 2011, 2216, 2217, 2397, 2517, 2552
<code>\cs_set_nopar:Npn</code>	2489
<code>\cs_set_protected:Npn</code>	498, 503, 520, 2570
<code>\cs_set_protected_nopar:Npn</code> ..	2212, 2214, 2535, 2553, 2572
<code>\cs_to_str:N</code>	2068
<code>\cs_undefine:c</code>	955, 978, 1725
<code>\cs_undefine:N</code>	96
<code>\curr@fontshape</code>	2228, 2231, 2240, 2246, 2257

D

<code>\DeclareExpandableDocumentCommand</code>	2012
<code>\DeclareInstance</code>	1399
<code>\DeclareObjectType</code>	1012
<code>\DeclareSymbolFont</code>	2173
<code>\DeclareTemplateCode</code>	1044
<code>\DeclareTemplateInterface</code>	1013
<code>\Default</code>	154
<code>\defaultCJKfontfeatures</code>	5, 2090, 2091, 2093, 2352
<code>\dim_compare:nNf</code>	2262
<code>\dim_compare:nNfT</code> ..	1117, 1142, 1163, 1176, 1231, 1258, 1266
<code>\dim_const:cn</code>	485
<code>\dim_eval:n</code>	609, 1189, 1209, 1288, 1336, 2266
<code>\dim_gset:cn</code>	485
<code>\dim_if_exist:cTF</code>	484
<code>\dim_max:nn</code>	1114, 1127, 1221, 1228, 1274, 2236
<code>\dim_min:nn</code>	1127, 1137, 1183, 1276, 1281
<code>\dim_new:N</code>	32
<code>\dim_ratio:nn</code>	2268
<code>\dim_set:Nn</code>	1089, 1328, 2258
<code>\dim_use:N</code>	1334, 1349, 1356

E

<code>xeCJKactive</code>	3
<code>\EditInstance</code>	1411
<code>\else:</code>	106, 128, 130, 252, 480, 985, 993, 1781
<code>\EmboldenFactor</code>	1523
<code>\end</code>	789, 790
<code>\etex_currentgrouplevel:D</code>	2200
<code>\etex_dimexpr:D</code>	1146, 1170, 1180, 1244, 1270
<code>\etex_fontcharwd:D</code>	1334
<code>\etex_iffontchar:D</code>	105
<code>\etex_lastnodetype:D</code>	445, 460, 635
<code>\etex_numexpr:D</code>	248, 2573
<code>\exp_after:wN</code>	100, 101, 143, 730, 1586, 2427, 2428, 2429, 2430, 2512
<code>\exp_args:Nc</code>	161, 2087
<code>\exp_args:No</code>	729
<code>\exp_args:NV</code>	62
<code>\exp_not:N</code>	561, 1716, 1886, 1887, 1890, 1974, 1975, 1976
<code>\exp_not:n</code>	344, 350, 1713
<code>\exp_not:V</code>	362, 1714, 1715, 2136
<code>\exp_not:v</code>	1820, 1833
<code>\exp_stop_f:</code>	105, 1356

F

<code>\f@baselineskip</code>	2272
<code>\f@series</code>	1917
<code>\f@shape</code>	1917
<code>\f@size</code>	1917, 2228, 2231, 2238, 2240, 2246, 2257, 2268
<code>\Fallback</code>	1610
<code>\fi:</code>	106, 130, 131, 143, 252, 480, 985, 993, 1781
<code>\file_input:n</code>	2524
<code>\fix@penalty</code>	2400, 2402
<code>\fontencoding</code>	1974, 2259
<code>\fontfamily</code>	1975, 2260
<code>\fontsize</code>	2264
<code>\fontspec_init:</code>	2398
<code>\fontspec_set_family:Nnn</code>	1713
<code>\fontspec_visible_space:</code>	2252
<code>\fontspec_visible_space_fallback:</code>	2253
<code>\fp_compare:nNfT</code>	1166, 1234, 1261
<code>\fp_gset:Nn</code>	1530, 1538
<code>\fp_new:N</code>	1521, 1522
<code>\fp_set:Nn</code>	1633, 1642
<code>\fp_set_eq:NN</code>	1665, 1666
<code>\fp_use:c</code>	1169, 1179, 1263
<code>\fp_use:N</code>	1145, 1243, 1268, 1741, 1746
<code>\FullLeft</code>	154
<code>\FullRight</code>	154

G

<code>\g__xeCJK_after_end_preamble_hook_tl</code>	49, 55, 58
<code>\g__xeCJK_after_preamble_hook_tl</code>	48, 53, 57
<code>\g__xeCJK_at_end_preamble_hook_tl</code>	47, 51, 56
<code>\g__xeCJK_auto_fake_bold_bool</code>	1519, 1526, 1529, 1663
<code>\g__xeCJK_auto_fake_slant_bool</code>	1520, 1534, 1537, 1664
<code>\g__xeCJK_base_class_seq</code>	229, 229, 230, 235, 583, 877
<code>\g__xeCJK_CJK_class_prop</code>	229, 238, 242, 286
<code>\g__xeCJK_CJK_class_seq</code>	229, 237, 241, 365, 2453
<code>\g__xeCJK_CJK_sub_class_seq</code> ..	817, 817, 893, 902, 1845, 2106
<code>\g__xeCJK_class_seq</code>	159, 163, 165, 2441, 2445
<code>\g__xeCJK_config_bool</code>	2281, 2284, 2289, 2295, 2579
<code>\g__xeCJK_config_name_tl</code>	2285, 2290, 2294, 2583
<code>\g__xeCJK_default_features_clist</code>	1677, 2090, 2092

<code>\g__xeCJK_embolden_factor_fp</code> ..	1521, 1530, 1541, 1635, 1665
<code>\g__xeCJK_family_font_name_prop</code>	1488, 1503, 1722, 1839, 1840, 1847, 1869, 1872, 1902, 2102, 2128, 2131, 2143
<code>\g__xeCJK_family_font_options_prop</code>	1839, 1841, 1849, 1875, 1879, 1903, 2107
<code>\g__xeCJK_family_int</code>	1649, 1668, 2083
<code>\g__xeCJK_family_name_prop</code>	1724, 1839, 1839, 1886, 1901, 1971, 2170
<code>\g__xeCJK_features_id_prop</code>	1611, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1651, 1749, 1828
<code>\g__xeCJK_indent_bool</code>	2299, 2356
<code>\g__xeCJK_inline_env_clist</code>	797, 804, 807, 808, 813
<code>\g__xeCJK_last_punct_tl</code>	614, 614, 618, 620, 621, 622, 628, 629, 630, 637, 655, 662, 663, 664, 671, 672, 673, 680, 681, 682, 689, 690, 691
<code>\g__xeCJK_math_bool</code>	2147, 2157
<code>\g__xeCJK_number_bool</code>	2298, 2567
<code>\g__xeCJK_punct_style_seq</code>	1339, 1392, 1398, 1401
<code>\g__xeCJK_punct_width_tl</code>	944, 1096, 1098
<code>\g__xeCJK_slant_factor_fp</code>	1522, 1538, 1542, 1644, 1666
<code>\g__xeCJK_special_punct_clist</code>	946, 946, 947, 950, 988
<code>\g__xeCJK_sub_key_seq</code>	1548, 1548, 1551, 1658
<code>\g__xeCJK_tmp_skip</code>	33, 121, 122
<code>\g__xeCJK_unknown_family_seq</code>	2021, 2023, 2028
<code>\g__xeCJK_verb_exspace_skip</code>	2213, 2215, 2230, 2234, 2240, 2243
<code>\g_fontspec_encoding_tl</code>	2259, 2355
<code>\group_begin:</code>	1501, 1671, 1711, 1844, 2493
<code>\group_end:</code>	1516, 1687, 1717, 1851, 2494
H	
<code>\HalfLeft</code>	166
<code>\HalfRight</code>	166
<code>\hbox_set:Nn</code>	120
<code>CheckSingle</code>	3
I	
<code>\icprotect</code>	2511, 2512
<code>MiddlePunct</code>	4
<code>MiddlePunct+</code>	4
<code>MiddlePunct-</code>	4
<code>\if_cs_exist:w</code>	992
<code>\if_dim:w</code>	479
<code>\if_int_compare:w</code>	126, 129, 141, 251, 984
<code>\if_predicate:w</code>	1779
<code>\IfBooleanF</code>	280, 1991
<code>\IfBooleanT</code>	257, 822, 832, 841, 2001
<code>\IfBooleanTF</code>	1995
<code>\IfInstanceExistTF</code>	1382, 1396, 1410
<code>\IfNoValueF</code>	2520
<code>\IfNoValueTF</code>	304, 305, 2068
<code>\indentfirst</code>	2296
<code>\InlineEnv</code>	802
<code>\InlineEnv+</code>	802
<code>\InlineEnv-</code>	802
<code>\int_compare:nNnF</code>	635, 2200, 2440
<code>\int_compare:nNnTF</code>	1495, 2405
<code>\int_compare:p:Nn</code>	445, 460
<code>\int_const:Nn</code>	154, 155, 156, 157, 158, 2177
<code>\int_eval:n</code>	2083
<code>\int_gincr:N</code>	1649
<code>\int_if_exist:cF</code>	866
<code>\int_if_exist:cTF</code>	848
<code>\int_incr:N</code>	293, 2195
<code>\int_max:nn</code>	305
<code>\int_min:nn</code>	304
<code>\int_new:N</code>	28, 29, 30, 1668
<code>\int_set:Nn</code>	285, 300, 304, 305, 310, 2452
<code>\int_set_eq:NN</code>	301
<code>\int_step_inline:nnnn</code>	2443
<code>\int_to_hexadecimal:n</code>	1489
<code>\ItalicFeatures</code>	1611
<code>\ItalicFont</code>	1611
J	
<code>CJKecglue</code>	3
<code>CJKglue</code>	3
<code>CJKmath</code>	3
<code>CJKnumber</code>	3
<code>CJKspace</code>	3
K	
<code>\KaiMingPunct</code>	927
<code>\keys_define:nn</code> ...	144, 497, 499, 528, 695, 802, 927, 1370, 1439, 1523, 1552, 1619, 1625, 2157, 2277, 2296, 2302, 2357
<code>\keys_set:nn</code>	2332, 2376
<code>\keys_set_known:nVN</code>	1679
<code>\keyval_parse:NNn</code>	1702
<code>\keyval_parse:NNV</code>	1693
<code>\KeyValue</code>	1018, 1019, 1020, 1021, 1024, 1025, 1026, 1027
L	
<code>\l__keys_module_tl</code>	2327
<code>\l__xeCJK_add_font_prop</code>	1650, 1733, 1735, 1836
<code>\l__xeCJK_add_min_bound_to_margin_bool</code>	1059, 1182
<code>\l__xeCJK_auto_fake_bold_bool</code> .	1629, 1632, 1663, 1787, 1802
<code>\l__xeCJK_auto_fake_slant_bool</code>	1638, 1641, 1664, 1789
<code>\l__xeCJK_check_single_bool</code>	703, 706, 708, 715, 717
<code>\l__xeCJK_current_coor_tl</code>	1922, 1924, 1931, 1933, 1952, 1963, 2257, 2275
<code>\l__xeCJK_different_align_margin_dim</code>	1069
<code>\l__xeCJK_different_align_ratio_fp</code>	1070
<code>\l__xeCJK_ecglue_skip</code>	453, 504, 521, 526
<code>\l__xeCJK_embolden_factor_fp</code>	1633, 1665, 1741
<code>\l__xeCJK_enabled_global_setting_bool</code> ..	1046, 1091, 1193
<code>\l__xeCJK_enabled_kerning_bool</code>	1062, 1198
<code>\l__xeCJK_fallback_bool</code>	1449, 1451, 1458, 1460, 1464
<code>\l__xeCJK_fallback_family_tl</code>	1502, 1504, 1508, 1509, 1511, 1512, 1513, 1514
<code>\l__xeCJK_family_name_tl</code>	1560, 1673, 1683, 1684, 1848, 1850, 1857, 1870, 1876, 1884, 1887
<code>\l__xeCJK_fixed_margin_ratio_fp</code>	1054
<code>\l__xeCJK_fixed_margin_width_dim</code>	1053
<code>\l__xeCJK_fixed_punct_ratio_fp</code>	1048
<code>\l__xeCJK_fixed_punct_width_dim</code>	1047
<code>\l__xeCJK_font_features_clist</code>	1676, 1678, 1680
<code>\l__xeCJK_font_name_bf_tl</code>	1764, 1766
<code>\l__xeCJK_font_name_tl</code>	1504, 1505, 1595, 1596, 1675, 1715, 1848
<code>\l__xeCJK_font_options_clist</code>	1674, 1677, 1846, 1850
<code>\l__xeCJK_fontspec_family_tl</code>	1713, 1716
<code>\l__xeCJK_fontspec_options_clist</code>	1656, 1680, 1714, 1832, 1837
<code>\l__xeCJK_kerning_margin_minimum_dim</code>	1073, 1222
<code>\l__xeCJK_kerning_margin_ratio_fp</code>	1072, 1268
<code>\l__xeCJK_kerning_margin_width_dim</code>	1071, 1266, 1267

\l__xeCJK_kerning_total_ratio_fp	1065, 1234, 1243	\LongPunct	927
\l__xeCJK_kerning_total_width_dim	1064, 1231, 1232		
\l__xeCJK_margin_minimum_dim	1061, 1115	M	
\l__xeCJK_middle_margin_ratio_fp	1058, 1145	\makeCJKactive	147, 151, 151
\l__xeCJK_middle_margin_width_dim	1057, 1142, 1143	\makeCJKinactive	
\l__xeCJK_middle_punct_ratio_fp	1052		148, 151, 152, 2485, 2487, 2489, 2491, 2502, 2554
\l__xeCJK_middle_punct_width_dim	1051	EmboldenFactor	4
\l__xeCJK_min_bound_to_kerning_bool	1063, 1224	\mddefault	2174
\l__xeCJK_mixed_margin_ratio_fp	1056	\MiddlePunct	927
\l__xeCJK_mixed_margin_width_dim	1055	\msg_critical:nn	10
\l__xeCJK_mixed_punct_ratio_fp	1050	\msg_critical:nnn	20
\l__xeCJK_mixed_punct_width_dim	1049	\msg_error:nn	38
\l__xeCJK_optimize_kerning_bool	1066, 1227	\msg_error:nnn	23
\l__xeCJK_optimize_margin_bool	1060, 1126, 1136	\msg_error:nnx	39
\l__xeCJK_reserve_space_bool	470, 530, 552	\msg_info:nnx	43
\l__xeCJK_same_align_margin_dim	1067	\msg_info:nnxx	44
\l__xeCJK_same_align_ratio_fp	1068	\msg_new:nnn	3, 11, 37
\l__xeCJK_slant_factor_fp	1642, 1666, 1746	\msg_redirect_module:nnn	2306, 2307, 2313, 2314
\l__xeCJK_sub_cancel_bool	825, 828, 830, 837, 839, 851	\msg_warning:nn	40
\l__xeCJK_sub_family_tl		\msg_warning:nnx	41
	1857, 1861, 1868, 1873, 1880, 1882, 1890	\msg_warning:nnxx	42
\l__xeCJK_sub_font_name_tl			
	1587, 1589, 1593, 1594, 1595, 1596, 1598, 1604, 1605	N	
\l__xeCJK_sub_font_options_clist	1583, 1590, 1597, 1602	indentfirst	3
\l__xeCJK_sub_key_seq	1563, 1657, 1855	\newCJKfontfamily	5, 2064, 2066
\l__xeCJK_tmp_box	31, 120	\NewDocumentCommand	151, 152, 254,
\l__xeCJK_tmp_dim	32,		273, 277, 314, 320, 327, 704, 713, 818, 826, 835, 921,
	1089, 1117, 1122, 1130, 1328, 1336, 1349, 2258, 2262, 2269		923, 1394, 1408, 1447, 1456, 1492, 1987, 2052, 2055,
\l__xeCJK_tmp_prop	34, 1692, 1695, 1704, 1706		2057, 2064, 2066, 2072, 2091, 2094, 2158, 2198, 2374,
\l__xeCJK_tmpa_clist			2379, 2381, 2383, 2384, 2385, 2387, 2388, 2389, 2391, 2518
	35, 261, 263, 264, 1494, 1495, 1496, 1876, 1878, 1880,	\newXeTeXintercharclass	161
	2104, 2111, 2113, 2125, 2127, 2135, 2136, 2179, 2181, 2182	InlineEnv	3
\l__xeCJK_tmpa_int	28, 284,	InlineEnv+	3
	289, 291, 292, 293, 310, 312, 2191, 2192, 2194, 2195, 2475	InlineEnv-	3
\l__xeCJK_tmpa_tl		\normalfont	2396, 2397
	26, 286, 360, 361, 362, 1083, 1127, 1137, 1147, 1180,	\NormalSpace	166
	1183, 1582, 1584, 1586, 1589, 1593, 1722, 1724, 1726,	\normalspacedchars	8, 314, 314
	1870, 1873, 1887, 1890, 1906, 1907, 2068, 2069, 2070,		
	2079, 2080, 2084, 2087, 2102, 2113, 2133, 2164, 2167, 2170	O	
\l__xeCJK_tmpb_clist	36, 2108, 2111	LocalConfig	2
\l__xeCJK_tmpb_int	29, 284, 289, 2191, 2192	LongPunct	4
\l__xeCJK_tmpb_tl	27, 1084, 1127, 1130, 1147,	LongPunct+	4
	1183, 2082, 2084, 2085, 2129, 2132, 2133, 2136, 2170, 2172	LongPunct-	4
\l__xeCJK_tmppc_int	30, 285, 286, 292		
\l__xeCJK_ulem_hook_used_bool	2531, 2533, 2540	P	
\l__xeCJK_verb_addon_bool	2203, 2205, 2222	\par	771
\l__xeCJK_xecglue_bool	393, 451, 509, 514, 519, 527	\PassOptionsToPackage	2309, 2316, 2322
\l_keys_choice_tl	1526, 1534	\pdfstringdefPreHook	2499
\l_keys_key_tl	2321, 2322, 2359, 2360, 2361	\pdftex_strcmp:D	126, 129
\l_keys_value_tl	1382, 1383, 1384, 1530, 1538, 1633, 1642, 2290	\peek_catcode:NTF	542, 723, 738, 781
\l_peek_token	420, 551, 553, 559,	\peek_catcode_ignore_spaces:NTF	546, 750
	561, 728, 729, 730, 743, 744, 745, 748, 755, 756, 771, 772	\peek_catcode_remove:NTF	779
\l__xeCJK_current_font_tl		\peek_meaning_remove:NTF	417, 786, 789
	997, 1001, 1915, 1915, 1916, 1920, 1922	\Pifont	2491
\l__xeCJK_family_tl		\prg_do_nothing:	340
	1469, 1916, 1923, 1950, 1953, 1954, 1956, 1959,	\prg_new_conditional:cpnn	990
	1962, 1992, 1999, 2000, 2008, 2008, 2009, 2102, 2108, 2129		103, 124, 133, 249, 477, 982, 987, 1772, 1777, 1821
\l__xeCJK_punct_style_tl		\prg_new_conditional:Npnn	
	997, 999, 1305, 1317, 1361, 1365, 1379, 1383		1980
SlantFactor	4	\prg_new_protected_conditional:Npnn	
AllowBreakBetweenPuncts	4	\prg_return_false:	
\LocalConfig	2277		106, 130, 137, 252, 480, 985, 993, 1775, 1781, 1824, 1985
		\prg_return_true:	106,
			127, 130, 136, 252, 480, 985, 993, 1775, 1781, 1824, 1985

\ProcessKeysOptions	2353	\setCJKmathfont	6, 2060, 2158, 2158
\prop_clear:N	1650, 1692	\setCJKmonofont	5, 2037, 2052, 2057, 2062
\prop_get:Nn	1491	\setCJKromanfont	2054, 2063
\prop_get:NnNF	2140	\setCJKsansfont	5, 2036, 2052, 2055, 2061
\prop_get:NnNT	1722, 1886, 1906	\SetSymbolFont	2175
\prop_get:No	1488	\sffamily	2394
\prop_get:NoNF	2128	\shapedefault	2174, 2176
\prop_get:NVNF	1503	\silent	2302
\prop_get:NVNT	1869, 1875, 2107, 2170	\skip_gset:Nn	2230, 2234
\prop_get:NVNTF	286, 2102	\skip_gset_eq:NN	121
\prop_get:NxNF	2131	\skip_horizontal:N	2213, 2215
\prop_gpop:NnNT	1724	\skip_horizontal:n	606, 2334
\prop_gput:Nnn	244, 1979	\skip_if_eq:nnTF	110
\prop_gput:NnV	1907	\skip_if_eq_p:nn	394, 452, 453
\prop_gput:Nvn	242	\skip_new:N	33, 526, 2243
\prop_gput:NVV	1847, 1849, 1872, 1879	\skip_set:Nn	2218
\prop_gput:Nxx	1971	\skip_set_eq:NN	122
\prop_if_empty:NTF	2143	\SlantedFeatures	1611
\prop_map_function:NN	1749	\SlantedFont	1611
\prop_map_inline:Nn	1617, 1651, 1695, 1828, 1836	\SlantFactor	1523
\prop_new:N	34, 238, 1611, 1839, 1840, 1841	\SplitArgument	274
\prop_put:Nnn	1612, 1613, 1614, 1615, 1616, 1704, 1706, 1733	\str_case:x:nnn	2033, 2044
\prop_put_if_new:Nnn	1735	\str_if_eq:nnF	266, 410, 425, 1939
\PunctStyle	1370	\str_if_eq:nnT	17, 1761
\punctstyle	2383, 2383	\str_if_eq:nnTF	881, 1564, 1603, 1786, 1942
\PunctWidth	927	\str_if_eq_p:nn	586, 1345, 1346, 1780, 1793, 1795
		\str_if_eq_x:nnTF	60
		\sw@slant	2402
Q			
\q_stop	64, 65, 68	T	
\quiet	2302	\tex_baselineskip:D	2334
		\tex_char:D	2573
R			
\r	2488, 2489	\tex_font:D	101, 105, 112, 113, 114, 1334, 2238, 2258, 2262, 2269
\RequirePackage	21, 24, 25, 2354, 2356, 2556, 2574	\tex_fontdimen:D	112, 113, 114, 2238, 2258, 2262, 2269
\reset@font	2397	\tex_global:D	2194
\reverse_if:N	141	\tex_hfil:D	2552
\rmfamily	2393	\tex_ignorespaces:D	578, 580, 2006, 2075, 2097, 2377, 2566
S			
\scan_stop:	248, 602, 800, 1148, 1170, 1180, 1247, 1270, 2252, 2573	\tex_kern:D	490, 491, 613
\selectfont	1976, 2261, 2273	\tex_lastkern:D	479
\seq_clear:N	1657	\tex_lastskip:D	121, 394, 452, 453
\seq_count:N	2441	\tex_let:D	2512
\seq_gclear:c	956	\tex_penalty:D	45, 46
\seq_gput_right:cn	960, 970	\tex_rightskip:D	2218
\seq_gput_right:Nn	241, 902, 1398, 1551, 2023	\tex_spaceskip:D	110, 116
\seq_gput_right:Nv	163	\tex_the:D	101, 338
\seq_gremove_all:cn	979	\tex_unkern:D	545, 2409, 2415, 2421
\seq_gset_split:Nnn	230	\tex_unskip:D	397, 456
\seq_if_in:cnF	967	\tex_vrule:D	599
\seq_if_in:NnF	2021, 2445	\textcentereddot	2480
\seq_map_function:NN	365, 2105	\textellipsis	2480
\seq_map_inline:cn	954	\textemdash	2480
\seq_map_inline:Nn	235, 583, 877, 893, 1339, 1658, 1845, 1855, 2453	\textperiodcentered	2480
\seq_new:c	951	\textquotedblleft	2481
\seq_new:N	165, 229, 237, 817, 1401, 1548, 2028	\textquotedblright	2481
\seq_put_right:Nn	1563	\textquoteleft	2481
\seq_use:Nnnn	1392	\textquoteright	2481
\setCJKfallbackfamilyfont	6, 6, 1492, 1492	\textvisiblespace	2248
\setCJKfamilyfont	5, 2039, 2064, 2064	\tipaencoding	2487
\setCJKmainfont	4, 2035, 2052, 2052, 2054, 2059	\tl_clear:c	1653
		\tl_clear:N	89, 1505
		\tl_const:Nn	108, 925, 926, 1387, 2160, 2581
		\tl_const:Nx	139, 2172, 2355
		\tl_gclear:N	90

\xeCJK_family_if_exist:xF	1473, 1980, 1997, 2015, 2163, 2166
\xeCJK_family_if_exist_use:x	1992, 1995, 2013, 2013
\xeCJK_font_gset_to_current:c	98, 98, 1924, 1933, 1963, 2275
\xeCJK_fontspec:nn	2077, 2077, 2089
\xeCJK_fontspec:VV	2113
\xeCJK_fontspec:xx	2074
\xeCJK_FullLeft_and_FullLeft:N	658, 658
\xeCJK_FullLeft_and_FullRight:N	667, 667
\xeCJK_FullRight_and_FullLeft:N	677, 677
\xeCJK_FullRight_and_FullRight:N	686, 686
\xeCJK_get_inter_class_toks:nn	337, 337, 344, 350, 356, 360, 2458, 2460, 2462, 2468
\xeCJK_get_punct_bounds:NN	620, 628, 634, 644, 661, 670, 679, 688, 1301, 1301
\xeCJK_get_punct_kerning:NN	1357, 1357, 1369
\xeCJK_get_punct_kerning:oN	662, 671, 680, 689
\xeCJK_glue_to_skip:nN	118, 118, 504, 521
\xeCJK_glyph_bounds:NN	1325, 1327, 1355, 1355
\xeCJK_glyph_if_exist:N	103
\xeCJK_glyph_if_exist:NTF	103, 1467, 1476
\xeCJK_glyph_if_exist_p:N	103
\xeCJK_gset_mathcode:nnnn	2184, 2189, 2189
\xeCJK_hook_for_ulem:	2529, 2529, 2557
\xeCJK_if_blank_x:n	124
\xeCJK_if_blank_x:nT	2461, 2467
\xeCJK_if_blank_x:nTF	124, 300, 1096, 1823, 1989
\xeCJK_if_blank_x:p:n	124, 298, 728
\xeCJK_if_last_node:n	477
\xeCJK_if_last_node:nT	407, 544
\xeCJK_if_last_node:nTF	406, 477, 2407, 2413, 2419
\xeCJK_if_last_node_p:n	400, 401, 439, 444, 459, 461, 467, 468, 477
\xeCJK_if_package_loaded:n	133
\xeCJK_if_package_loaded:nF	2308, 2315
\xeCJK_if_package_loaded:nT	2490, 2497, 2543
\xeCJK_if_package_loaded:nTF	133, 2320
\xeCJK_if_package_loaded_p:n	133, 2511
\xeCJK_if_same_class:NN	249
\xeCJK_if_same_class:NNTF	249, 1236
\xeCJK_if_same_class_p:NN	249
\xeCJK_ignore_spaces:w	538, 540, 540, 564, 2217, 2431, 2563
\xeCJK_ignorespaces:	564
\xeCJK_int_until_do:nn	140, 140, 289, 2192
\xeCJK_inter_class_toks:nnc	595
\xeCJK_inter_class_toks:nnn	334, 334, 336, 340, 374, 381, 389, 415, 428, 534, 565, 569, 577, 579, 581, 582
\xeCJK_inter_class_toks:nnx	343, 349, 355, 362
\xeCJK_make_node:n	418, 421, 422, 482, 488, 537, 545, 2410, 2416, 2422, 2563
\xeCJK_new_class:n	159, 159, 166, 167, 168, 868
\xeCJK_no_break:	45, 46, 617, 623, 648, 660, 669, 674, 692, 774, 782, 799, 933
\xeCJK_pre_inter_class_toks:nnn	341, 341, 346, 426, 588, 589, 883, 897, 899, 906, 2546
\xeCJK_pre_inter_class_toks:nnx	2459
\xeCJK_punct_kerning_process:NN	1079, 1185, 1185
\xeCJK_punct_margin_process:NN	1078, 1081, 1081
\xeCJK_replace_inter_class_toks:nnnn	358, 358, 887
\xeCJK_reverse:nnn	59, 59, 1087, 1820
\xeCJK_select_font:	377, 433, 573, 1312, 1915, 1918, 1936, 1943, 2554
\xeCJK_select_font:x	1475, 1915, 1927
\xeCJK_set_char_class:nnn	268, 277, 279, 282
\xeCJK_set_family:nnn	1669, 1669, 1689
\xeCJK_set_family:Vnn	2085
\xeCJK_set_family:Vvv	1512, 1861
\xeCJK_set_family:xxx	1497, 2053, 2056, 2058, 2065, 2070, 2159
\xeCJK_set_family_fallback:nnN	1499, 1499, 1518
\xeCJK_set_family_fallback:xxN	1496
\xeCJK_set_mathfont:	2147, 2161, 2161
\xeCJK_set_visible_space_font:	2247, 2255, 2255
\xeCJK_swap_cs:NN	91, 91, 2536, 2537
\xeCJK_tl_remove_outer_braces:N	61, 61
\xeCJK_tl_remove_outer_braces:n	61, 62, 63, 1582
\xeCJK_value_class:n	247, 247, 251, 984
\xeCJK_visible_space_fallback:	2244, 2244, 2253
\xeCJKactive	144
\xeCJKallowbreakbetweenpuncts	2389, 2389
\xeCJKCancelSubCJKBlock	8, 825, 826
\xeCJKcaption	2516, 2518
\xeCJKDeclareCharClass	8, 254, 254
\xeCJKDeclarePunctStyle	9, 1394, 1394, 1407, 1415, 1416, 1417, 1424, 1432
\xeCJKDeclareSubCJKBlock	8, 818, 818, 824, 861
\xeCJKdisablechecksingle	699, 703, 713
\xeCJKdisablefallback	1443, 1447, 1456
\xeCJKEditPunctStyle	9, 1408, 1408, 1414
\xeCJKenablechecksingle	698, 703, 704
\xeCJKenablefallback	1442, 1447, 1447
\xeCJKmainfont	2154
\xeCJKnobreakbetweenpuncts	2389, 2391
\xeCJKplainchr	2383, 2384
\xeCJKResetCharClass	8, 327, 327, 333
\xeCJKResetPunctClass	8, 257, 280, 320, 320, 331, 822, 832, 841
\xeCJKRestoreSubCJKBlock	8, 825, 835
\xeCJKsetcharclass	277
\xeCJKsetecglue	2386
\xeCJKsetemboldenfactor	2379, 2379
\xeCJKsetkern	9, 923, 923
\xeCJKsetslantfactor	2379, 2381
\xeCJKsetup	2, 919, 2211, 2367, 2374, 2374, 2380, 2382, 2383, 2384, 2385, 2387, 2388, 2390, 2392
\xeCJKsetwidth	9, 921, 921
\xeCJKVerbAddon	5, 2198, 2198, 2223, 2225
\XeTeX@CJKfntef@hook	2548, 2553
\xetex_if_engine:F	10
\XeTeXcharclass	248, 292, 312, 317
\XeTeXcharglyph	1356
\XeTeXdefaultencoding	2520, 2527
\XeTeXglyphbounds	912, 916, 1356
\XeTeXinterchartokenstate	151, 152, 2405
\XeTeXinterchartoks	335, 338
\XeTeXmathcode	2194
\xpatchcmd	84
\xpretocmd	74, 77