

xeCJK 宏包

ctex.org

2013/05/29 3.2.1*

目录

1	简介	1	5.4	字符类别设定	18
2	基本用法	2	5.5	字符类别处理	22
3	用户手册	2	5.6	字符输出规则	25
3.1	宏包选项	2	5.7	全角右标点后的断行	34
3.2	字体设置与选择	5	5.8	段末孤字处理	35
3.2.1	X _Y TeX 的字体名查找	7	5.9	增加 CJK 子分区	38
3.3	CJK 分区字体设置	8	5.10	标点处理	40
3.4	设置 CJK 字符范围	9	5.11	后备字体	50
3.5	标点符号的处理	9	5.12	CJK 字体族声明方式	52
3.5.1	设置特定标点符号的宽度和间距	9	5.13	字体切换	59
3.5.2	定义标点符号处理格式	10	5.14	数字字体设置	65
3.6	其它	11	5.15	抄录环境中的间距调整	66
4	已知问题和兼容性	12	5.16	xeCJK 其它选项	68
5	xeCJK 代码实现	12	5.17	xeCJK 初始化设置	69
5.1	运行环境检查	12	5.18	兼容性修补	72
5.2	内部工具	13	5.19	xeCJKfntef	76
5.3	功能开关	18	5.20	xeCJK.cfg	83
			版本历史	84	
			代码索引	84	

1 简介

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

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

xeCJK 使用了 X_YTeX 的一些最新特性,需要 X_YTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 依赖 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。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 rev418.

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

其中 $\langle key_1 \rangle$, $\langle key_2 \rangle$ 是设置选项,而 $\langle val_1 \rangle$, $\langle val_2 \rangle$ 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。使用粗体来表示 `xeCJK` 的默认设置。

<code>LocalConfig \star</code>	<code>LocalConfig = {$\langle true false name \rangle$}</code>
---	---

New: 2012-11-22

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/> ★	CJKnumber = $\langle true false \rangle$ 为了与 CJKnumb 宏包兼容,xeCJK 作了一些特别处理。因此,如果需要使用到 CJKnumb,请通过这个选项调用它。
<hr/> indentfirst <hr/> ★	indentfirst = $\langle true false \rangle$ Updated: 2012-11-22 是否使用 indentfirst 宏包,使得跟在章节标题后面的第一段首行也缩进。
<hr/> xeCJKactive <hr/>	xeCJKactive = $\langle true false \rangle$ 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。
<hr/> CJKspace <hr/>	CJKspace = $\langle true false \rangle$ 缺省状态下,xeCJK 会忽略 CJK 文字之间的空格,使用这一选项来保留它们之间的空格。
<hr/> CJKmath <hr/> ★	CJKmath = $\langle true false \rangle$ 是否支持在数学环境中直接输入 CJK 字符。使用这个选项后,可以直接在数学环境中输出 CJK 字符。
<hr/> CJKglue <hr/>	CJKglue = $\{\backslash\text{hskip } 0\text{pt plus } 0.08\backslash\text{baselineskip}\}$ 设置 CJK 文字之间插入的 glue,上边是 xeCJK 的默认值。一般来说,除非有特殊需要(例如,改变文字间距等),否则不需要设置这个选项,使用默认值即可。如果要设置这个选项,为了行末的对齐,设置的 glue 最好有一定的弹性。
<hr/> CJKecglue <hr/>	CJKecglue = $\{\langle glue \rangle\}$ 设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距,默认值是一个空格。使用这个选项设置的 glue 最好也要用一定的弹性。请注意,这里设置的 glue 只影响 xeCJK 根据需要自动添加的空白,源文件中直接输入的 CJK 文字与西文之间的空格不受影响(直接输出)。有时候 xeCJK 可能不能正确地调整间距,需要手动加空格。
<hr/> xCJKecglue <hr/>	xCJKecglue = $\{\langle true false glue \rangle\}$ 缺省状态下,xeCJK 不对源文件中直接输入的 CJK 文字与西文之间的空格进行调整,如果需要调整,请使用这个选项。如果使用这个选项,将使用 CJKecglue 替换源文件中直接输入的 CJK 文字与西文之间的空格。
<hr/> CheckSingle <hr/>	CheckSingle = $\langle true false \rangle$ Updated: 2012-12-06 是否避免单个 CJK 文字单独占一个段落的最后一行。需要说明的是,这个选项只有在段末的最后一个字是 CJK 文字或者标点符号,并且倒数第二和第三个字都是文字才能正确处理孤字的问题。如果这倒数三个字有作为控制序列的参数情况,那么一般来说也不能正确处理。
<hr/> PlainEquation <hr/>	PlainEquation = $\langle true false \rangle$ New: 2012-12-06 如果使用了 $\$ \$ \dots \$ \$$ 的形式来输入行间数学公式,就需要启用本选项,以便 CheckSingle 选项能够正确识别。推荐使用 $\backslash[\dots \backslash]$ 的形式来输入行间数学公式。
<hr/> NewLineCS <hr/>	NewLineCS = $\{ \backslash\text{par } \backslash[\}$ NewLineCS+ NewLineCS- New: 2012-12-04 设置造成断行的控制序列,以便 CheckSingle 选项能够正确识别。以上是 xeCJK 的初始设置。
<hr/> EnvCS <hr/>	EnvCS = $\{ \backslash\text{begin } \backslash\text{end } \}$ EnvCS+ EnvCS- New: 2012-12-04 设置 \LaTeX 环境开始和结束的控制序列,以便 CheckSingle 选项能够正确识别。以上是 xeCJK 的初始设置。

InlineEnv	InlineEnv = { $\langle env_1 \rangle$, $\langle env_2 \rangle$, $\langle env_3 \rangle$, ...}
InlineEnv+	在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 \LaTeX 环境的开始 $\begin{...}$ 和结束 $\end{...}$ 视为断行的地方,如果有某些特殊的 \LaTeX 环境没有造成断行,可以使用这个选项来声明它,以便 CheckSingle 能正确识别。
InlineEnv-	
Updated: 2012-12-06	
AutoFallBack	AutoFallBack = $\langle true false \rangle$
	当文档中有个别生僻字时,可以使用这个选项,自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。
AutoFakeBold ☆	AutoFakeBold = { $\langle true false \text{数字} \rangle$ }
	全局设定当没有声明对应的粗体时,是否使用 伪粗体 ;当输入的是数字时,将使用伪粗体,并将使用输入的数字作为伪粗体的默认粗细程度。
AutoFakeSlant ☆	AutoFakeSlant = { $\langle true false \text{数字} \rangle$ }
	全局设定当没有声明对应的斜体时,是否使用 伪斜体 ;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。
EmboldenFactor ☆	EmboldenFactor = { $\langle \text{数字} 4 \rangle$ }
	设置伪粗体的默认粗细程度。
SlantFactor ☆	SlantFactor = { $\langle \text{数字} 0.167 \rangle$ }
	设置伪斜体的粗细程度,范围是 $-0.999 \sim 0.999$ 。
PunctStyle	PunctStyle = { $\langle \text{quanjiao} \text{banjiao} \text{kaiming} \text{hangmobanjiao} \text{CCT} \text{plain} \dots \rangle$ }
Updated: 2012-11-10	设置标点处理格式。xeCJK 中预先定义好的格式为
quanjiao	全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;
banjiao	半角式: 所有标点占半个汉字宽度;
kaiming	开明式: 句末点号用全角,其他半角;
hangmobanjiao	行末半角式: 所有标点占一个汉字宽度,行首行末对齐;
CCT	CCT 格式;
plain	原样(不调整标点间距)。
	可以使用 3.5.2 中介绍的 $\backslash\text{xeCJKDeclarePunctStyle}$ 定义新的标点格式。
KaiMingPunct ☆	KaiMingPunct = { $\langle . \text{ } . \text{ } ? \text{ } ! \text{ } \rangle$ }
KaiMingPunct+	设置开明(kaiming)标点处理格式时的句末点号,KaiMingPunct 后带的 + 与 - 分别表示从已有的开明句末点号中增加或减少标点。
KaiMingPunct-	
LongPunct ☆	LongPunct = { $\langle \text{——} \dots \rangle$ }
LongPunct+	设置长标点,例如破折号“——”与省略号“……”,允许在长标点前后断行,但是禁止在它们之间断行。
LongPunct-	
MiddlePunct ☆	MiddlePunct = { $\langle \text{——} \cdot \cdot \cdot \rangle$ }
MiddlePunct+	设置居中显示的标点,例如间隔号“·”。对于在 CJK 文字之间的居中标点,xeCJK 会根据不同的标点处理格式,调整居中标点与前后文字之间的空白,保证其确实居中。对于行末出现的居中标点,允许在其后面断行,但禁止在它前面断行。
MiddlePunct-	
PunctWidth ☆	PunctWidth = { $\langle \text{length} \rangle$ }
	缺省状态下,xeCJK 会根据所选择的标点处理格式自动计算标点所占的宽度,如果对缺省设置不满意,可以通过这一选项来改变它。为了使得标点所占的宽度能够适应字体大小的变化,这里设置的 length 的单位最好用 em 等相对距离单位,而不建议使用诸如 pt 之类的绝对距离单位。这里的设置可用于除了 plain 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 $\backslash\text{xeCJKsetwidth}$ 。

AllowBreakBetweenPuncts

AllowBreakBetweenPuncts = $\langle true | false \rangle$

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

CheckFullRight

CheckFullRight = $\langle true | false \rangle$

New: 2012-12-02

某些控制序列要求不能在它的前面断行。但是在缺省状态下, 单个全角右标点的后面总是可以断行的。因此当这些控制序列出现在全角右标点后面时, 可能会出现意料之外的断行。此时可以使用这个选项来避免这个情况。

NoBreakCS
NoBreakCS+
NoBreakCS-

New: 2012-12-02

NoBreakCS = { `\footnote` `\footnotemark` `\nobreak` }

设置不能在全角右标点后面断行的控制序列。以上是 `\xeCJK` 的默认设置。如果这些控制序列在文档中只出现少量几次, 也可以不必使用 **CheckFullRight** 选项, 而是手工在这些控制序列前面加上 3.6 节介绍的 `\xeCJKnobreak`。

Verb

Verb = $\langle true | false | env | env+ \rangle$

Updated: 2013-05-29

true 表示在 `\verb` 命令或 `verbatim` 环境里不自动调整中英文之间的间距。**env** 选项在 `verbatim` 环境里自动计算中西文间距和中文之间的间距, 以便于保持代码的对齐; **env** 选项不调整 `\verb` 里的间距, **env+** 选项还将正文里设置的间距应用到 `\verb` 里。这个选项对使用到 `\verbatim@font` 命令的情形均有效, 更一般的情况可以使用 3.6 节介绍的 `\xeCJKVerbAddon`。

3.2 字体设置与选择

\setCJKmainfont ★

\setCJKmainfont [$\langle font\ features \rangle$] $\langle font\ name \rangle$

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

AutoFakeBold**AutoFakeBold** = $\langle true | false | 数字 \rangle$

AutoFakeSlant

AutoFakeSlant = $\langle true | false | 数字 \rangle$

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 [$\langle font\ features \rangle$] $\langle font\ name \rangle$

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

\setCJKmonofont ★

\setCJKmonofont [$\langle font\ features \rangle$] $\langle font\ name \rangle$

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

\setCJKfamilyfont ★

\setCJKfamilyfont $\langle family \rangle$ [$\langle font\ features \rangle$] $\langle font\ name \rangle$

声明新的 CJK 字体族 $\langle family \rangle$ 并指定字体。

\CJKfamily

\CJKfamily $\langle family \rangle$

\CJKfamily+ $\langle family \rangle$

Updated: 2012-10-27

\CJKfamily- $\langle family \rangle$

用于在文档中切换 CJK 字体族, $\langle family \rangle$ 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。

<hr/> <hr/> <code>\newCJKfontfamily</code> ★	<code>\newCJKfontfamily [<i><family></i>] [<i><font-switch></i>] [<i></i>] {<i></i>}</code> <p>声明新的 CJK 字体族 <i><family></i> 并指定字体,并定义 <i><font-switch></i>,在文档中可以使用它来切换 CJK 字体族。可以不必指定 <i><family></i>,这时候 <i><family></i> 将等于 <i><font-switch></i>。事实上,<code>\newCJKfontfamily</code> 是 <code>\setCJKfamilyfont</code> 和 <code>\CJKfamily</code> 的合并。例如</p> <pre> \newCJKfontfamily[song]\songti{SimSun} </pre> <p>等价于</p> <pre> \setCJKfamilyfont{song}{SimSun} \newcommand*{\songti}{\CJKfamily{song}} </pre>
<hr/> <code>\CJKfontspec</code> <hr/>	<code>\CJKfontspec [<i></i>] {<i></i>}</code> <p>在文档中随机定义新的 CJK 字体族,并马上使用它。</p>
<hr/> <code>\defaultCJKfontfeatures</code> ★ <hr/>	<code>\defaultCJKfontfeatures {<i></i>}</code> <p>全局设置 CJK 字体族的默认选项。例如,使用</p> <pre> \defaultCJKfontfeatures{Scale=0.962216} </pre> <p>可以将全部 CJK 字体缩小为 0.962216。<code>xeCJK</code> 宏包的初始化设置是</p> <pre> \defaultCJKfontfeatures{Script=CJK} </pre>
<hr/> <code>\addCJKfontfeatures</code> <hr/>	<code>\addCJKfontfeatures {<i></i>}</code> <p>临时增加当前使用的 CJK 字体的选项。例如,使用</p> <pre> \addCJKfontfeatures{Scale=1.1} </pre> <p>可以将文档中当前使用的 CJK 字体放大为 1.1。</p>
<hr/> <code>\CJKrmdefault</code> <hr/>	保存 <code>\textrm</code> 和 <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/> <code>\CJKsfdefault</code> <hr/>	保存 <code>\textsf</code> 和 <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/> <code>\CJKttdefault</code> <hr/>	保存 <code>\texttt</code> 和 <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/> <code>\CJKfamilydefault</code> <hr/> <div>Updated: 2013-01-01</div> <hr/>	<p>保存 <code>\textnormal</code> 和 <code>\normalfont</code> 所使用的 CJK 字体族。类似西文字体的 <code>\familydefault</code>。初始值是 <code>\CJKrmdefault</code>。如果没有在导言区中修改它,<code>xeCJK</code> 会在导言区结束的时候根据西文字体的情况自动更新 <code>\CJKfamilydefault</code>。因此,在导言区里使用</p> <pre> \renewcommand\familydefault{\sfdefault} </pre> <p>就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。</p>
<hr/> <code>\setCJKmathfont</code> ★ <hr/>	<code>\setCJKmathfont [<i></i>] {<i></i>}</code> <p>设置数学公式中的 CJK 字体族。如果使用了 <code>CJKmath</code> 选项,但是没有使用 <code>\setCJKmathfont</code> 设置数学公式中的 CJK 字体,那么将使用 <code>\CJKfamilydefault</code> 作为数学公式中的 CJK 字体。</p>
<hr/> <code>\setCJKfallbackfamilyfont</code> ★ <hr/>	<code>\setCJKfallbackfamilyfont {<i><family></i>} [<i></i>] {<i></i>}</code> <p>设置 CJK 字体族 <i><family></i> 的备用字体。例如,使用</p> <pre> \setCJKmainfont{SimSun} \setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB} </pre> <p>可以将 <code>SimSun-ExtB</code> 作为 <code>SimSun</code> 的备用字体。</p>

FallBack = {[font features]}{font name}}

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

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

如果 FallBack 的值为空,将设置的是备用字体。例如,

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

等价于

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

```
\setCJKfallbackfamilyfont ★ \setCJKfallbackfamilyfont {family} [common font features]
{
  {[font features1]} {font name1}} ,
  {[font features2]} {font name2}} ,
  .....
}
```

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

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

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

3.2.1 XeTeX 的字体名查找

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

XeTeX 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,Ítálico,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,Félkövér dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrto Ítálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın Ítalik,
Krepko poševno,nghiêng đâp,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντονα,Bold,Negrita,
Lihavoitu,Gras,Félkövér,Grassetto,Vet,Halvfet,Pogrubiona,Negrto,
Полужирный,Fet,Kalın,Krepko,đâp,Lodia
Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,
thường,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, 細明體
NSimSun, 新宋体
PMingLiU, 新細明體
SimHei, 黑体
SimSun, 宋体
```

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

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

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

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

3.3 CJK 分区字体设置

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

```
\xeCJKDeclareSubCJKBlock ★ \xeCJKDeclareSubCJKBlock {⟨block⟩} {⟨block range⟩}
                             \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 `⟨font features⟩` 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}
```

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。

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

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

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

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

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

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

3.4 设置 CJK 字符范围

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

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

<code>\xeCJKResetCharClass</code> ★	<code>\xeCJKResetCharClass</code>
-------------------------------------	-----------------------------------

用于恢复 `xeCJK` 对各个字符类别的初始化设置。

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

用于重置标点符号所属的字符类。

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

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

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。

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

`<标点列表>` 可以是单个标点,也可以是多个标点。例如,

`\xeCJKsetwidth{。?}{0.7em}`

将设置句号和问号所占的宽度为 0.7 em。

<code>\xeCJKsetkern</code> ★	<code>\xeCJKsetkern {<前标点>} {<后标点>} {<length>}</code>
------------------------------	---

`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

<code>\xeCJKDeclarePunctStyle</code> ★	<code>\xeCJKDeclarePunctStyle {<style>} {<options>}</code>
--	--

New: 2012-11-10

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。

<code>\xeCJKEditPunctStyle</code> ★	<code>\xeCJKEditPunctStyle {<style>} {<options>}</code>
-------------------------------------	---

New: 2012-11-10

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-ratio *<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-ratio *<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。这个选项对居中的标点无效。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。

3.6 其它

<code>\xeCJKVerbAddon</code>	<code>\xeCJKVerbAddon</code>
New: 2012-11-19	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。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。
<code>\xeCJKKnobreak</code>	……汉字。 \xeCJKKnobreak\footnote{脚注}
New: 2012-12-03	\xeCJKKnobreak 用在全角标点符号后面,目的是确保不能在此处断行。如果已经启用了前面介绍的 CheckFullRight 选项,则不需要再用此命令。

4 已知问题和兼容性

xeCJK 使用并重新定义了 CJK 宏包的部分宏命令,如 `\CJKfamily`、`\CJKsymbol` 和 `\CJKglue` 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。

xeCJK 包含有一个子宏包 `xeCJKfntef`, 可以用它来实现汉字加点和可断行的下划线等。它是 `CJKfntef` 宏包在 \XeTeX 下的替换版本,基本用法完全一致,在 `CJKfntef` 的源文件 `CJKfntef.sty` 的注解部分里有说明。

新版本(3.x)的 xeCJK 完全使用 \LaTeX 的语法来编写。 \LaTeX 3 放弃了 `\outer` 宏的概念,因此相关工具在遇到 `\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 \use_i:nn
```

的错误。目前已知的有 `protect` 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 `protect` 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 \XeTeX 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 \XeTeX 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 } }
21 \RequirePackage { xtemplate }
22 \ifpackagelater { xtemplate } { 2012/11/10 } { }
23 { \msg_error:nnn { xeCJK } { l3-too-old } { xtemplate } }
24 \RequirePackage { xparse , l3keys2e }
```

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。

5.2 内部工具

分配临时变量。

```

25 \tl_new:N \l__xeCJK_tmpa_tl
26 \tl_new:N \l__xeCJK_tmpb_tl
27 \int_new:N \l__xeCJK_tmpa_int
28 \int_new:N \l__xeCJK_tmpb_int
29 \int_new:N \l__xeCJK_tmpc_int
30 \box_new:N \l__xeCJK_tmp_box
31 \dim_new:N \l__xeCJK_tmp_dim
32 \skip_new:N \l__xeCJK_tmp_skip
33 \prop_new:N \l__xeCJK_tmp_prop
34 \clist_new:N \l__xeCJK_tmpa_clist
35 \clist_new:N \l__xeCJK_tmpb_clist

```

__xeCJK_msg_new:nn 各种信息函数的缩略形式。

```

\__xeCJK_error:n      36 \cs_new_protected_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:nx     37 \cs_new_protected_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_warning:nx   38 \cs_new_protected_nopar:Npn \__xeCJK_error:nx { \msg_error:nxx { xeCJK } }
\__xeCJK_info:nx      39 \cs_new_protected_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\__xeCJK_info:nxx     40 \cs_new_protected_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nxx { xeCJK } }
                      41 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nxxx { xeCJK } }
                      42 \cs_new_protected_nopar:Npn \__xeCJK_info:nx { \msg_info:nxx { xeCJK } }
                      43 \cs_new_protected_nopar:Npn \__xeCJK_info:nxx { \msg_info:nxxx { xeCJK } }

```

(End definition for __xeCJK_msg_new:nn and others.)

\xeCJK_allow_break:

```

\xecjk_no_break:      44 \cs_new_protected_nopar:Npn \xeCJK_allow_break: { \tex_penalty:D \c_zero }
                      45 \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_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 46 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
                              47 {
                              48   \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
                              49   { \prg_return_true: } { \prg_return_false: }
                              50 }
                              51 \tl_const:Nx \c__xeCJK_package_ext_tl { \@pkgextension }

```

(End definition for \xeCJK_if_package_loaded:n.)

__xeCJK_at_end_preamble:n 在 \document 前后加上各种钩子。

```

\__xeCJK_after_preamble:n 52 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
\__xeCJK_after_end_preamble:n 53 \tl_new:N \g__xeCJK_after_preamble_hook_tl
                              54 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl
                              55 \cs_new_protected:Npn \__xeCJK_at_end_preamble:n #1
                              56 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
                              57 \cs_new_protected:Npn \__xeCJK_after_preamble:n #1
                              58 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
                              59 \cs_new_protected:Npn \__xeCJK_after_end_preamble:n #1
                              60 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
                              61 \xeCJK_if_package_loaded:nTF { etoolbox }
                              62 {
                              63   \AtEndPreamble { \g__xeCJK_at_end_preamble_hook_tl }
                              64   \AfterPreamble { \g__xeCJK_after_preamble_hook_tl }
                              65   \AfterEndPreamble { \g__xeCJK_after_end_preamble_hook_tl }
                              66 }
                              67 {
                              68   \AtBeginDocument { \g__xeCJK_after_preamble_hook_tl }
                              69   \cs_new_protected_nopar:Npn \__xeCJK_document_left_hook:
                              70   { \group_end: \g__xeCJK_at_end_preamble_hook_tl \group_begin: }
                              71   \cs_new_protected_nopar:Npn \__xeCJK_document_right_hook:
                              72   { \scan_stop: \g__xeCJK_after_end_preamble_hook_tl \tex_ignorespaces:D }

```

```

73 \cs_gset_nopar:Npx \document
74 {
75 \__xeCJK_document_left_hook:
76 \exp_not:o { \document }
77 \__xeCJK_document_right_hook:
78 }
79 }
(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。

```

80 \cs_new_nopar:Npn \xeCJK_reverse:nnn #1#2#3
81 { \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 82 \cs_new_protected_nopar:Npn \xeCJK_tl_remove_outer_braces:N #1
83 { \tl_set:Nx #1 { \exp_args:NV \xeCJK_tl_remove_outer_braces:n #1 } }
84 \cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
85 { \__xeCJK_tl_remove_outer_braces:w #1 \q_stop }
86 \cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \q_stop
87 {
88 \bool_if:nTF { \tl_if_single_p:n {#1} && \tl_if_head_is_group_p:n {#1} }
89 { \__xeCJK_tl_remove_outer_braces:w #1 \q_stop }
90 { \tl_trim_spaces:n {#1} }
91 }
(End definition for \xeCJK_tl_remove_outer_braces:N and \xeCJK_tl_remove_outer_braces:n.)

```

`\xeCJK_cs_clear:N` 让控制序列的意义为空。

```

\xeCJK_cs_gclear:N 92 \cs_new_eq:NN \xeCJK_cs_clear:N \tl_clear:N
93 \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。

```

94 \cs_new_protected:Npn \xeCJK_swap_cs:NN #1#2
95 {
96 \cs_set_eq:NN \__xeCJK_swap_cs_aux:w #1
97 \cs_set_eq:NN #1 #2
98 \cs_set_eq:NN #2 \__xeCJK_swap_cs_aux:w
99 \cs_undefine:N \__xeCJK_swap_cs_aux:w
100 }
(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。

```

101 \cs_new_protected_nopar:Npn \xeCJK_font_gset_to_current:c #1
102 {
103 \exp_after:wN \cs_gset_eq:NN
104 \cs:w #1 \exp_after:wN \cs_end: \tex_the:D \tex_font:D
105 }
(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`

```

106 \prg_new_conditional:Npnn \xeCJK_glyph_if_exist:N #1 { p , T , F , TF }
107 {
108 \etex_iffontchar:D \tex_font:D `#1 \exp_stop_f:
109 \prg_return_true: \else: \prg_return_false: \fi:
110 }
(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。

```

111 \tl_const:Nn \c_xeCJK_space_skip_tl
112 {
113   \int_compare:nNnTF \g__xeCJK_spacefactor_int = \c_one_thousand
114   {
115     \skip_if_eq:nNnTF \tex_spaceskip:D \c_zero_skip
116     {
117       \tex_fontdimen:D \c_two \tex_font:D
118       plus \tex_fontdimen:D \c_three \tex_font:D
119       minus \tex_fontdimen:D \c_four \tex_font:D
120     }
121     { \tex_spaceskip:D }
122   }
123   {
124     \skip_if_eq:nNnTF \tex_spaceskip:D \c_zero_skip
125     {
126       \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
127       { \__xeCJK_space_skip_scale:nnn { \tex_fontdimen:D \c_two \tex_font:D } }
128       {
129         \skip_if_eq:nNnTF \tex_xspaceskip:D \c_zero_skip
130         {
131           \__xeCJK_space_skip_scale:nnn
132           {
133             \tex_fontdimen:D \c_two \tex_font:D +
134             \tex_fontdimen:D \c_seven \tex_font:D
135           }
136         }
137         { \tex_xspaceskip:D \use_none:nn }
138       }
139       { \tex_fontdimen:D \c_three \tex_font:D }
140       { \tex_fontdimen:D \c_four \tex_font:D }
141     }
142     {
143       \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
144       { \__xeCJK_space_skip_scale:nnn { \tex_spaceskip:D } }
145       {
146         \skip_if_eq:nNnTF \tex_xspaceskip:D \c_zero_skip
147         {
148           \__xeCJK_space_skip_scale:nnn
149           { \tex_spaceskip:D + \tex_fontdimen:D \c_seven \tex_font:D }
150         }
151         { \tex_xspaceskip:D \use_none:nn }
152       }
153       { \etex_gluestretch:D \tex_spaceskip:D }
154       { \etex_glueshrink:D \tex_spaceskip:D }
155     }
156   }
157 }
158 \cs_new_nopar:Npn \__xeCJK_space_skip_scale:nnn #1#2#3
159 {
160   \dim_eval:n {#1}
161   plus \fp_eval:n { \g__xeCJK_spacefactor_int / 1000 } #2
162   minus
163   \int_div_truncate:nn
164   { 1000 * \tex_number:D #3 } { \g__xeCJK_spacefactor_int } sp
165 }
166 \int_new:N \g__xeCJK_spacefactor_int
167 \int_gset_eq:NN \g__xeCJK_spacefactor_int \c_one_thousand
(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。

```

168 \cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2

```

```

169 {
170   \hbox_set:Nn \l__xeCJK_tmp_box { #1 \scan_stop: \exp_after:wN } \exp_after:wN
171   \skip_set:Nn \exp_after:wN #2 \exp_after:wN { \skip_use:N \tex_lastskip:D }
172 }
(End definition for \xeCJK_glue_to_skip:nN.)

```

`\xeCJK_if_blank_x:n` 判断是否为空或者仅含一个空格。

```

\xeCJK_if_blank_x:nTF
173 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
174 {
175   \if_case:w \pdfTeX_strcmp:D { } {#1} \exp_stop_f:
176   \prg_return_true:
177   \else:
178   \if_case:w \pdfTeX_strcmp:D { ~ } {#1} \exp_stop_f:
179   \prg_return_true: \else: \prg_return_false: \fi:
180   \fi:
181 }
(End definition for \xeCJK_if_blank_x: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
182 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
183 { \__xeCJK_int_until_do:wn \use_none:n { \reverse_if:N \if_int_compare:w #1#2 } }
184 \cs_new_protected:Npn \__xeCJK_int_until_do:wn \use_none:n #1
185 { #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.)

```

`\xeCJK_peek_catcode_ignore_spaces:NTF` 我们在里面设置了一个变量 `\l__xeCJK_peek_ignore_spaces_bool` 用于标识后面的空格是否被省略掉了。

```

186 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
187 {
188   \cs_set_eq:NN \l__peek_search_token #1 \scan_stop:
189   \tl_set:Nn \__xeCJK_peek_catcode_true:w { \group_align_safe_end: #2 }
190   \tl_set:Nn \__xeCJK_peek_catcode_false:w { \group_align_safe_end: #3 }
191   \bool_set_false:N \l__xeCJK_peek_ignore_spaces_bool
192   \group_align_safe_begin:
193   \peek_after:Nw \__xeCJK_peek_catcode_ignore_spaces_branches:w
194 }
195 \cs_new_protected_nopar:Npn \__xeCJK_peek_catcode_ignore_spaces_branches:w
196 {
197   \if_meaning:w \l__peek_token \c_space_token
198   \bool_set_true:N \l__xeCJK_peek_ignore_spaces_bool
199   \exp_after:wN \peek_after:Nw
200   \exp_after:wN \__xeCJK_peek_catcode_ignore_spaces_branches:w
201   \tex_romannumeral:D 0
202   \else:
203   \if_catcode:w
204   \exp_not:N \l__peek_token \exp_not:N \l__peek_search_token
205   \exp_after:wN \exp_after:wN
206   \exp_after:wN \__xeCJK_peek_catcode_true:w
207   \else:
208   \exp_after:wN \exp_after:wN
209   \exp_after:wN \__xeCJK_peek_catcode_false:w
210   \fi:
211   \fi:
212 }
213 \bool_new:N \l__xeCJK_peek_ignore_spaces_bool
(End definition for \xeCJK_peek_catcode_ignore_spaces:NTF.)

```

`\xeCJK_peek_after_ignore_spaces:nw` 与 `\@ifnextchar` 和 `\futurenonspacel` 类似,会省略掉后面的空格。

```

214 \cs_new_protected:Npn \xeCJK_peek_after_ignore_spaces:nw #1
215 {
216   \tl_set:Nn \__xeCJK_peek_after_do:w { \group_align_safe_end: #1 }
217   \group_align_safe_begin:

```

```

218 \peek_after:Nw \__xeCJK_peek_ignore_spaces_branches:w
219 }
220 \cs_new_protected_nopar:Npn \__xeCJK_peek_ignore_spaces_branches:w
221 {
222 \if_meaning:w \l_peek_token \c_space_token
223 \exp_after:wN \peek_after:Nw
224 \exp_after:wN \__xeCJK_peek_ignore_spaces_branches:w
225 \tex_romannumeral:D 0
226 \else:
227 \exp_after:wN \__xeCJK_peek_after_do:w
228 \fi:
229 }
(End definition for \xeCJK_peek_after_ignore_spaces:nw.)

```

`\xeCJK_token_value_class:N` 用于取得记号 #1 所在的 \XeTeX 字符类。#1 应为 `\catcode` 为 11 或 12 的显性或隐性记号。

```

230 \cs_new_nopar:Npn \xeCJK_token_value_class:N #1
231 { \XeTeXcharclass \xeCJK_token_value_charcode:N #1 }
(End definition for \xeCJK_token_value_class:N.)

```

`\xeCJK_token_value_charcode:N` 当记号 #1 的 `charcode` 大于等于 `0x10000` 时, \XeTeX 0.9999.0 版以前的 `\meaning` 的返回结果比较特殊⁴, 需要特别处理。同时使用较新版本中提供的原语设置 `mathcode`。

```

232 \cs_new_nopar:Npn \xeCJK_token_value_charcode:N #1
233 { \exp_after:wN \__xeCJK_token_value_charcode:w \token_to_meaning:N #1 \q_stop }
234 \str_if_eq_x:nnTF { \token_to_str:N \Umathcode } { \token_to_meaning:N \Umathcode }
235 {
236 \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3 \q_stop
237 { '#3 }
238 \cs_new_eq:NN \xeCJK_xetex_mathcode:w \Umathcode
239 }
240 {
241 \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
242 { \tl_if_empty:nTF {#4} { '#3 } { "20000 } }
243 \cs_new_eq:NN \xeCJK_xetex_mathcode:w \XeTeXmathcode
244 }
(End definition for \xeCJK_token_value_charcode:N.)

```

`\xeCJK_if_CJK_class_p:N` 判断字符 #1 是否为 CJK 字符类, 包括文字和标点符号。

```

\xeCJK_if_CJK_class:NNTF
245 \prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
246 {
247 \if_cs_exist:w \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 } \cs_end:
248 \prg_return_true: \else: \prg_return_false: \fi:
249 }
250 \cs_new_nopar:Npn \__xeCJK_CJK_class_tl:n #1
251 { c__xeCJK_CJK_class_ \int_eval:n {#1} _tl }
252 \cs_generate_variant:Nn \__xeCJK_CJK_class_tl:n { c }
(End definition for \xeCJK_if_CJK_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
253 \prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
254 {
255 \if_int_compare:w
256 \xeCJK_token_value_class:N #1 = \xeCJK_token_value_class:N #2 \exp_stop_f:
257 \prg_return_true: \else: \prg_return_false: \fi:
258 }
(End definition for \xeCJK_if_same_class:NN.)

```

⁴参见 <http://tug.org/pipermail/xetex/2013-January/023967.html> 和 <http://tex.stackexchange.com/a/64848>。

5.3 功能开关

`xeCJKActive` 事实上,将开启或关闭 XeTeX 的整个字符类机制。

```
259 \keys_define:nn { xecjk / options }
260 {
261   xecjkActive .choice: ,
262   xecjkActive / true .code:n = { \makexecjkActive } ,
263   xecjkActive / false .code:n = { \makexecjkInactive } ,
264   xecjkActive .default:n = { true }
265 }
```

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

`\makexecjkActive`

`\makexecjkInactive`

```
266 \NewDocumentCommand \makexecjkActive { } { \XeTeXinterchartokenstate = \c_one }
267 \NewDocumentCommand \makexecjkInactive { } { \XeTeXinterchartokenstate = \c_zero }
```

(End definition for \makexecjkActive and \makexecjkInactive.)

抑制 BOM。

```
268 \char_set_catcode_ignore:n { "FEFF }
```

5.4 字符类别设定

`\g__xecjk_class_seq` 分别用于记录在 xecjk 中使用的字符类别名称和新建的字符类别的编号。
`\g__xecjk_new_class_seq`

```
269 \seq_new:N \g__xecjk_class_seq
270 \seq_new:N \g__xecjk_new_class_seq
```

(End definition for \g__xecjk_class_seq and \g__xecjk_new_class_seq.)

`\xecjk_new_class:n`

新建一个字符类别。#1 是自定义名称。

```
271 \cs_new_protected_nopar:Npn \xecjk_new_class:n #1
272 {
273   \int_if_exist:cTF { \__xecjk_class_csname:n {#1} }
274   { \__xecjk_error:nx { class-already-defined } {#1} }
275   {
276     \exp_args:Nc \newXeTeXintercharclass { \__xecjk_class_csname:n {#1} }
277     \clist_new:c { g__xecjk_#1_range_clist }
278     \seq_gput_right:Nn \g__xecjk_class_seq {#1}
279     \seq_gput_right:Nv \g__xecjk_new_class_seq { \__xecjk_class_csname:n {#1} }
280   }
281 }
```

(End definition for \xecjk_new_class:n.)

`\xecjk_save_class:nn`

保存 XeTeX 预定义的字符类别。#1 是自定义名称,#2 是编号。

```
282 \cs_new_protected_nopar:Npn \xecjk_save_class:nn #1#2
283 {
284   \int_if_exist:cTF { \__xecjk_class_csname:n {#1} }
285   { \__xecjk_error:nx { class-already-defined } {#1} }
286   {
287     \int_const:cn { \__xecjk_class_csname:n {#1} } {#2}
288     \clist_new:c { g__xecjk_#1_range_clist }
289     \seq_gput_right:Nn \g__xecjk_class_seq {#1}
290   }
291 }
```

(End definition for \xecjk_save_class:nn.)

`__xecjk_class_csname:n`

字符类名称对应的控制序列名字。

```
292 \cs_new_nopar:Npn \__xecjk_class_csname:n #1 { c__xecjk_#1_class_int }
293 \cs_new_eq:cN { \__xecjk_class_csname:n { Others } } \l__xecjk_tmpa_int
294 \__xecjk_msg_new:nn { class-already-defined }
295 {
296   XeTeX~character~class~`#1'~has~been~already~defined.\\\
297   Please~take~another~name. \\
298 }
```

(End definition for _xeCJK_class_csname:n.)

xeCJK 需要以下字符类别用于字符输出。其中 Default、CJK、FullLeft、FullRight、Boundary 为 XeTeX 中预定义的类别,xeCJK 新增加了 HalfLeft、HalfRight、NormalSpace 和 IVS。其中异体字选择符 (Ideographic Variation Selectors)⁵ 需要 XeTeX 0.9999.0 以上的版本⁶和相关字体的支持。

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
IVS	异体字选择符	“回字有四样写法”

Default 这五类是 XeTeX 预定义的类别。

```

    CJK      299 \xeCJK\_save\_class:nn { Default } { \c\_zero }
FullLeft    300 \xeCJK\_save\_class:nn { CJK } { \c\_one }
FullRight   301 \xeCJK\_save\_class:nn { FullLeft } { \c\_two }
Boundary    302 \xeCJK\_save\_class:nn { FullRight } { \c\_three }
            303 \xeCJK\_save\_class:nn { Boundary } { \c\_two\_hundred\_fifty\_five }

```

(End definition for Default and others.)

HalfLeft 新增西文半角左/右标点、前后原始间距的符号和异体字选择符类。

```

HalfLeft    304 \xeCJK\_new\_class:n { HalfLeft }
HalfRight   305 \xeCJK\_new\_class:n { HalfRight }
NormalSpace 306 \xeCJK\_new\_class:n { NormalSpace }
IVS         307 \xeCJK\_new\_class:n { IVS }

```

(End definition for HalfLeft and others.)

```

\_c\_xeCJK\_HalfLeft\_chars\_clist  xeCJK 设置的基本字符类的默认范围。
\_c\_xeCJK\_HalfRight\_chars\_clist 308 \clist\_const:Nn \c\_xeCJK\_HalfLeft\_chars\_clist
\_c\_xeCJK\_NormalSpace\_chars\_clist 309 { "28 , "2D , "5B , "60 , "7B }
\_c\_xeCJK\_FullLeft\_chars\_clist 310 \clist\_const:Nn \c\_xeCJK\_HalfRight\_chars\_clist
\_c\_xeCJK\_FullRight\_chars\_clist 311 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
\_c\_xeCJK\_CJK\_chars\_clist 312 \clist\_const:Nn \c\_xeCJK\_NormalSpace\_chars\_clist { "2F }
\_c\_xeCJK\_IVS\_chars\_clist 313 \clist\_const:Nn \c\_xeCJK\_FullLeft\_chars\_clist
314 {
315     "2018 , "201C , "3008 , "300A , "300C , "300E , "3010 , "3012 , "3014 ,
316     "3016 , "3018 , "301A , "301D , "3036 , "FE59 , "FE5B , "FE5D , "FE5F ,
317     "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 , "FF3B , "FF5B ,
318     "FFE0 , "FFE1 , "FFE5 , "FFE6
319 }
320 \clist\_const:Nn \c\_xeCJK\_FullRight\_chars\_clist
321 {
322     "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2027 , "2500 ,
323     "3001 , "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 ,
324     "3015 , "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 ,
325     "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C ,
326     "309D , "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 ,
327     "30E5 , "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE ,
328     "FE50 , "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C ,
329     "FE5E , "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B ,
330     "FF1F , "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 ,
331     "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E ,
332     "FF9F

```

⁵<http://www.unicode.org/reports/tr37/>

⁶<http://tug.org/pipermail/xetex/2013-March/024118.html>

```

333     }
334     \clist_const:Nn \c__xeCJK_CJK_chars_clist
335     {

```

- Hangul Jamo (谚文字母)

```

336         "1100 -> "11FF ,

```
- CJK Radicals Supplement (中日韩部首补充)

```

337         "2E80 -> "2EFF ,

```
- Kangxi Radicals (康熙字典部首)

```

338         "2F00 -> "2FDF ,

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

```

339         "2FF0 -> "2FFF ,

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

```

340         "3000 -> "303F ,

```
- Hiragana (日文平假名)

```

341         "3040 -> "309F ,

```
- Katakana (日文片假名)

```

342         "30A0 -> "30FF ,

```
- Bopomofo (注音字母)

```

343         "3100 -> "312F ,

```
- Hangul Compatibility Jamo (谚文兼容字母)

```

344         "3130 -> "318F ,

```
- Kanbun (象形字注释标志)

```

345         "3190 -> "319F ,

```
- Bopomofo Extended (注音字母扩展)

```

346         "31A0 -> "31BF ,

```
- CJK Strokes (中日韩笔画)

```

347         "31C0 -> "31EF ,

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

```

348         "31F0 -> "31FF ,

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

```

349         "3200 -> "32FF ,

```
- CJK Compatibility (中日韩兼容)

```

350         "3300 -> "33FF ,

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

```

351         "3400 -> "4DBF ,

```
- Yijing Hexagrams Symbols (易经六十四卦符号)

```

352         "4DC0 -> "4DFF ,

```
- CJK Unified Ideographs (中日韩统一表意文字)

```

353         "4E00 -> "9FFF ,

```


- Yi Syllables (彝文音节)
354 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
355 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
356 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
357 "AC00 -> "D7AF ,
- Hangul Jamo Extended-B (谚文扩展 B)
358 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
359 "F900 -> "FAFF ,
- Vertical Forms (竖排形式)
360 "FE10 -> "FE1F ,
- CJK Compatibility Forms (中日韩兼容形式)
361 "FE30 -> "FE4F ,
- Halfwidth and Fullwidth Forms (半角及全角形式)
362 "FF00 -> "FFEF ,
- Kana Supplement (日文假名增补)
363 "1B000 -> "1B0FF ,
- Enclosed Ideographic Supplement (带圈表意文字增补)
364 "1F200 -> "1F2FF ,
- CJK Unified Ideographs Extension-B (中日韩统一表意文字扩展 B)
365 "20000 -> "2A6DF ,
- CJK Unified Ideographs Extension-C (中日韩统一表意文字扩展 C)
366 "2A700 -> "2B73F ,
- CJK Unified Ideographs Extension-D (中日韩统一表意文字扩展 D)
367 "2B740 -> "2B81F ,
- CJK Compatibility Ideographs Supplement (中日韩兼容表意文字增补)
368 "2F800 -> "2FA1F
369 }
370 \clist_const:Nn \c__xeCJK_IVS_chars_clist
371 {
- COMBINING KATAKANA-HIRAGANA VOICED SOUND MARK
372 "3099 ,
- COMBINING KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK
373 "309A ,
- Variation Selectors (异体字选择符)
374 "FE00 -> "FE0F ,
- Variation Selectors Supplement (异体字选择符增补)
375 "E0100 -> "E01EF
376 }
(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_non_CJK_class_seq
\g__xeCJK_CJK_class_seq
377 \seq_new:N \g__xeCJK_base_class_seq
378 \seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq
379 \seq_new:N \g__xeCJK_non_CJK_class_seq
380 \seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq
381 { Default , HalfLeft , HalfRight , NormalSpace , Boundary }
382 \seq_new:N \g__xeCJK_CJK_class_seq
383 \cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
384 {
385   \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
386   \tl_const:cn { \__xeCJK_CJK_class_tl:c { \__xeCJK_class_csname:n {#1} } } {#1}
387 }
388 \clist_map_function:nN { CJK , FullLeft , FullRight , IVS } \__xeCJK_save_CJK_class:n
(End definition for \g__xeCJK_base_class_seq, \g__xeCJK_non_CJK_class_seq, and \g__xeCJK_CJK_class_seq.)

\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。
389 \cs_new_nopar:Npn \xeCJK_class_num:n #1 { \use:c { \__xeCJK_class_csname:n {#1} } }
(End definition for \xeCJK_class_num:n.)

\XeCJKDeclareCharClass
390 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
391 {
392   \xeCJK_declare_char_class:nx {#2} {#3}
393   \IfBooleanT {#1} { \xeCJKResetPunctClass }
394 }
(End definition for \xeCJKDeclareCharClass. This function is documented on page 9.)

\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
\__xeCJK_set_char_class_aux:Nnw 类似 "1100 -> 11FF 起止范围的使用方式。
395 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
396 {
397   \clist_set:Nn \l__xeCJK_tmpa_clist {#2}
398   \clist_gconcat:ccN
399   { \g__xeCJK_#1_range_clist } { \g__xeCJK_#1_range_clist } \l__xeCJK_tmpa_clist
400   \clist_map_inline:Nn \l__xeCJK_tmpa_clist
401   {
402     \str_if_eq:nnF {##1} { -> }
403     {
404       \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
405       { \xeCJK_class_num:n {#1} }
406     }
407   }
408 }
409 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw
410 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
411 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
412 \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_catcode:nnn 设置字符类别的同时,设置所有 CJK 字符的 \catcode 为 12。XeTeX 的配置文件 unicode-letters.
tex 将所有 CJK 表意文字的 \catcode 设置为 11。我们在作出改变的目的是方便 CheckSingle 的
判断。事实上,CheckSingle 的判断也可以不依赖 \catcode,而改为判断 \XeTeXcharclass。但是
判读一个隐性记号(\l_peek_token)的 \XeTeXcharclass 比直接判断它的 \catcode 要低效得多。
413 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class_catcode:nnn #1#2#3
414 {
415   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
416   \int_set:Nn \l__xeCJK_tmppc_int {#3}
417   \int_compare:nNnTF
418   { \use:c { \__xeCJK_class_csname:n { CJK } } } = \l__xeCJK_tmppc_int
419   { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \char_set_catcode_other:n }
```

```

420 { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \use_none:n }
421 \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
422 {
423   \__xeCJK_set_char_catcode:n { \l__xeCJK_tmpa_int }
424   \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
425   \int_incr:N \l__xeCJK_tmpa_int
426 }
427 }
(End definition for \__xeCJK_set_char_class_catcode:nnn.)

```

`__xeCJK_set_char_class:nnn` 只设置字符类别,而不改变 `\catcode`。

```

428 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class:nnn #1#2#3
429 {
430   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
431   \int_set:Nn \l__xeCJK_tmpc_int {#3}
432   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
433   {
434     \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
435     \int_incr:N \l__xeCJK_tmpa_int
436   }
437 }
(End definition for \__xeCJK_set_char_class:nnn.)

```

`__xeCJK_check_num_range:nnNN`

```

438 \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
439 {
440   \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
441   {
442     \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
443     \int_set_eq:NN #3 #4
444   }
445   {
446     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
447     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
448   }
449 }
(End definition for \__xeCJK_check_num_range:nnNN.)

```

`\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。在宏包内部使用时,设置 CJK 字符的 `\catcode`,而在文档中使用时不设置。

```

450 \cs_new_eq:NN \xeCJK_set_char_class:nnn \__xeCJK_set_char_class_catcode:nnn
451 \AtEndOfPackage
452 { \cs_set_eq:NN \xeCJK_set_char_class:nnn \__xeCJK_set_char_class:nnn }
(End definition for \xeCJK_set_char_class:nnn.)

```

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

```

453 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
454 {
455   \int_set:Nn \l__xeCJK_tmpa_int { \xeCJK_class_num:n {#2} }
456   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
457   { \XeTeXcharclass ##1 = \l__xeCJK_tmpa_int }
458 }
(End definition for \__xeCJK_set_char_class_eq:nn.)

```

`\normalspacedchars` 声明前后不加间距的字符。

```

459 \NewDocumentCommand \normalspacedchars { m }
460 {
461   \tl_map_inline:nn {#1}
462   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
463 }
(End definition for \normalspacedchars. This function is documented on page 9.)

```

\xeCJKResetPunctClass 用于重置标点符号所属的字符类。

```
464 \NewDocumentCommand \xeCJKResetPunctClass { }
465 {
466   \xeCJK_declare_char_class:nV { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
467   \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
468   \xeCJK_declare_char_class:nV { FullLeft } \c__xeCJK_FullLeft_chars_clist
469   \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
470 }
```

(End definition for \xeCJKResetPunctClass. This function is documented on page 9.)

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

```
471 \NewDocumentCommand \xeCJKResetCharClass { }
472 {
473   \xeCJK_declare_char_class:nV { CJK } \c__xeCJK_CJK_chars_clist
474   \xeCJK_declare_char_class:nV { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
475   \xeCJK_declare_char_class:nV { IVS } \c__xeCJK_IVS_chars_clist
476   \xeCJKResetPunctClass
477 }
```

(End definition for \xeCJKResetCharClass. This function is documented on page 9.)

设置字符类别。

```
478 \xeCJKResetCharClass
```

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

```
479 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
480 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }
481 \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。

```
482 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2
483 { \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。

```
484 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2
485 { \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。

```
486 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
487 {
488   \xeCJK_inter_class_toks:nnx {#1} {#2}
489   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
490 }
491 \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。

```
492 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
493 {
494   \xeCJK_inter_class_toks:nnx {#1} {#2}
495   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
496 }
497 \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。

```

498 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
499 {
500   \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
501   \tl_if_empty:NF \l__xeCJK_tmpa_tl
502   { \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_tl } }
503 }
(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。

```

504 \cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
505 {
506   \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
507   \tl_if_empty:NF \l__xeCJK_tmpa_tl
508   {
509     \tl_replace_all:Nnn \l__xeCJK_tmpa_tl {#3} {#4}
510     \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_tl }
511   }
512 }
(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。

```

513 \cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
514 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
515 \cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
516 { \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	IVS
Default		✓	✓	✓				✓	✓
CJK	✓	✓	✓	✓	✓	✓	✓	✓	
FullLeft	✓	✓	✓	✓	✓	✓	✓	✓	✓
FullRight	✓	✓	✓	✓	✓	✓	✓	✓	✓
HalfLeft		✓	✓	✓					✓
HalfRight		✓	✓	✓				✓	✓
NormalSpace		✓	✓	✓				✓	✓
Boundary	✓	✓	✓	✓	✓		✓		✓
IVS	✓	✓	✓	✓	✓	✓	✓	✓	✓

\xeCJK_class_group_begin:
 \xeCJK_class_group_end:

```

517 \cs_new_eq:NN \xeCJK_class_group_begin: \c_group_begin_token
518 \cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token
(End definition for \xeCJK_class_group_begin: and \xeCJK_class_group_end:.)
IVS 字符类与 CJK 字符类基本相同,只是从 CJK 转移到 IVS 时,不加入任何内容。
519 \AtEndOfPackage
520 {
521   \seq_map_inline:Nn \g__xeCJK_class_seq
522   {
523     \str_if_eq:nnTF {#1} { IVS }
524     { \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } { CJK } }
525     {
526       \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } {#1}
527       \str_if_eq:nnF {#1} { CJK }
528       { \xeCJK_copy_inter_class_toks:nnnn {#1} { IVS } {#1} { CJK } }
529     }
530   }
531 }

```

```

532 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
533 {
534   \xeCJK_inter_class_toks:nnn {#1} { CJK }
535   {
536     \xeCJK_class_group_begin:
537     \xeCJK_select_font:
538     \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
539     \xeCJK_clear_Boundary_and_CJK_toks:
540     \CJKsymbol
541   }
542   \xeCJK_inter_class_toks:nnn { CJK } {#1} { \xeCJK_class_group_end: }
543 }
544 \clist_map_inline:nn { Default , HalfLeft , NormalSpace }
545 {
546   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
547   {
548     \bool_if:nTF
549     {
550       \l__xeCJK_xecglue_bool &&
551       \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
552       \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
553     }
554     {
555       \tex_unskip:D
556       \bool_if:nTF
557       {
558         \xeCJK_if_last_node_p:n { CJK } ||
559         \xeCJK_if_last_node_p:n { CJK-space }
560       }
561       { \CJKecglue } { ~ }
562     }
563     {
564       \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
565       { \xeCJK_if_last_node:nT { CJK-space } { \__xeCJK_space_or_xecglue: } }
566     }
567   }
568   \str_if_eq:nnF {#1} { NormalSpace }
569   { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
570 }
571 \clist_map_inline:nn { Default , HalfRight , NormalSpace }
572 {
573   \xeCJK_inter_class_toks:nnn {#1} { Boundary }
574   {
575     \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
576     \peek_meaning_remove:NTF \tex_italiccorrection:D
577     { \tex_italiccorrection:D { \xeCJK_make_node:n { default } } }
578     {
579       \token_if_space:NTF \l_peek_token
580       { { \xeCJK_make_node:n { default-space } } }
581       { { \xeCJK_make_node:n { default } } }
582     }
583   }
584   \str_if_eq:nnF {#1} { NormalSpace }
585   { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
586 }
587 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
588 {
589   \xeCJK_check_for_glue:
590   \xeCJK_class_group_begin:
591   \xeCJK_clear_Boundary_and_CJK_toks:
592   \xeCJK_select_font:
593   \CJKsymbol

```


594 }

\xeCJK_check_for_glue:

```
595 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
596 {
597   \bool_if:nTF
598     { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
599     { \CJKglue }
600     {
601       \bool_if:nTF
602         {
603           \xeCJK_if_last_node_p:n { default } ||
604           \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
605         }
606         { \CJKecglue }
607         {
608           \bool_if:nT
609             {
610               \l__xeCJK_xecglue_bool &&
611               \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
612               ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
613                 \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_ecglue_skip )
614             }
615             {
616               \tex_unskip:D
617               \bool_if:nTF
618                 {
619                   \xeCJK_if_last_node_p:n { default-space } ||
620                   \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
621                   \xeCJK_if_last_node_p:n { default }
622                 }
623                 { \CJKecglue }
624                 {
625                   \bool_if:nTF
626                     {
627                       \xeCJK_if_last_node_p:n { CJK } ||
628                       \xeCJK_if_last_node_p:n { CJK-space }
629                     }
630                     { \bool_if:NTF \l__xeCJK_reserve_space_bool { ~ } { \CJKglue } }
631                     { ~ }
632                 }
633             }
634         }
635     }
636 }
```

(End definition for \xeCJK_check_for_glue:.)

\xeCJK_if_last_node_p:n

\xeCJK_if_last_node:nTF

```
637 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
638 {
639   \if_dim:w \use:c { c__xeCJK_#1_node_dim } = \tex_lastkern:D
640   \prg_return_true: \else: \prg_return_false: \fi:
641 }
```

(End definition for \xeCJK_if_last_node:n.)

\xeCJK_def_node:nn 用于判断插入的各种 kern。

\xeCJK_make_node:n

```
642 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
643 {
644   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
645     { \dim_gset:cn } { \dim_const:cn }
646     { c__xeCJK_#1_node_dim } {#2}
647 }
```

```

648 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
649 {
650   \tex_kern:D - \use:c { c__xeCJK_#1_node_dim }
651   \tex_kern:D \use:c { c__xeCJK_#1_node_dim }
652 }
653 \xeCJK_def_node:nn { CJK } { 11 sp }
654 \xeCJK_def_node:nn { CJK-space } { 12 sp }
655 \xeCJK_def_node:nn { default } { 13 sp }
656 \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。

```

657 \keys_define:nn { xeCJK / options }
658 {
659   CJKglue .code:n =
660   {
661     \cs_set_protected_nopar:Npn \CJKglue {#1}
662     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
663   }
664 }
665 \skip_new:N \l__xeCJK_ccglue_skip
(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。

```

666 \keys_define:nn { xeCJK / options }
667 {
668   CJKecglue .code:n =
669   {
670     \cs_set_protected_nopar:Npn \CJKecglue {#1}
671     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
672   } ,
673   xCJKecglue .choice: ,
674   xCJKecglue / true .code:n =
675   {
676     \bool_set_true:N \l__xeCJK_xecglue_bool
677     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
678   } ,
679   xCJKecglue / false .code:n =
680   {
681     \bool_set_false:N \l__xeCJK_xecglue_bool
682     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \c_space_tl
683   } ,
684   xCJKecglue / unknown .code:n =
685   {
686     \bool_set_true:N \l__xeCJK_xecglue_bool
687     \cs_set_protected_nopar:Npn \CJKecglue {#1}
688     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
689     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
690   } ,
691   xCJKecglue .default:n = { true }
692 }
693 \skip_new:N \l__xeCJK_ecglue_skip
694 \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。

```

695 \keys_define:nn { xeCJK / options }
696 {
697   CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
698   space .meta:n = { CJKspace = true } ,
699   nospace .meta:n = { CJKspace = false }
700 }

```

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

```

701 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
702 {
703   \xeCJK_class_group_end:
704   { \xeCJK_make_node:n { CJK } }
705   \xeCJK_ignore_spaces:w
706 }

```

\xeCJK_ignore_spaces:w

```

707 \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
708 {
709   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
710   {
711     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
712     { \__xeCJK_space_or_xecglue: } { \CJKecglue }
713   }
714   {
715     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
716     {
717       \tex_unkern:D \tex_unkern:D
718       { \xeCJK_make_node:n { CJK-space } }
719       \bool_if:nT
720       {
721         \token_if_macro_p:N \l_peek_token ||
722         ( \l__xeCJK_reserve_space_bool &&
723           \token_if_other_p:N \l_peek_token )
724       }
725       { \__xeCJK_space_or_xecglue: }
726     }
727     { \cs_if_exist:NF \l_peek_token \exp_not:N }
728   }
729 }

```

(End definition for \xeCJK_ignore_spaces:w.)

```

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

```

\xeCJK_CJK_and_CJK:N

```

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

```

(End definition for \xeCJK_CJK_and_CJK:N.)

```

732 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
733 { \xeCJK_FullLeft_and_CJK: \CJKsymbol }
734 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
735 { \xeCJK_FullRight_and_CJK: \CJKsymbol }
736 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
737 {
738   \clist_map_inline:nn { FullLeft , FullRight }
739   {
740     \xeCJK_inter_class_toks:nnx {#1} {##1}
741     { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
742     \xeCJK_inter_class_toks:nnc {##1} {#1} { xeCJK_##1_and_Default: }
743   }
744 }
745 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
746 { \xeCJK_Boundary_and_FullLeft:N }
747 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
748 { \xeCJK_Boundary_and_FullRight:N }

```

\xeCJK_FullRight_and_Boundary:

```

749 \xeCJK_app_inter_class_toks:nnn { FullLeft } { Boundary } { \tex_ignorespaces:D }
750 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
751 { \xeCJK_FullRight_and_Boundary: }

```

(End definition for \xeCJK_FullRight_and_Boundary:.)

\xeCJK_FullRight_and_Boundary:

```
752 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Boundary:
753 { \xeCJK_FullRight_and_Default: \tex_ignorespaces:D }
(End definition for \xeCJK_FullRight_and_Boundary:.)
754 \clist_map_inline:nn { CJK , FullLeft , FullRight }
755 {
756   \clist_map_inline:nn { FullLeft , FullRight }
757   { \xeCJK_inter_class_toks:nnc {#1} {##1} { \xeCJK_#1_and_##1:N } }
758 }
```

__xeCJK_punct_rule:NN 用于抹去标点符号的左/右空白。

```
759 \cs_new_protected_nopar:Npn \__xeCJK_punct_rule:NN #1#2
760 {
761   \tex_vrule:D
762   width - \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
763   depth \c_zero_dim
764   height \c_zero_dim \scan_stop:
765 }
(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。

```
766 \cs_new_protected_nopar:Npn \__xeCJK_punct_glue:NN #1#2
767 {
768   \__xeCJK_punct_hskip:n
769   {
770     \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
771     minus \dim_eval:n { ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} ) / \c_two }
772   }
773 }
774 \cs_new_eq:NN \__xeCJK_punct_hskip:n \skip_horizontal:n
(End definition for \__xeCJK_punct_glue:NN.)
```

__xeCJK_punct_kern:NN 相邻两个标点之间的间距。

```
775 \cs_new_protected_nopar:Npn \__xeCJK_punct_kern:NN #1#2
776 { \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。

```
777 \tl_new:N \g__xeCJK_last_punct_tl
(End definition for \g__xeCJK_last_punct_tl.)
```

\xeCJK_FullLeft_and_CJK:

```
778 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_CJK:
779 {
780   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
781   {
782     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
783     \xeCJK_no_break:
784     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
785   }
786   { \xeCJK_no_break: }
787 }
(End definition for \xeCJK_FullLeft_and_CJK:.)
```

\xeCJK_FullLeft_and_Default:

```
788 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_Default:
789 {
790   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
791   {
792     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
793     \xeCJK_class_group_end: \xeCJK_no_break:

```

```

794     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
795   }
796   { \xeCJK_class_group_end: \xeCJK_no_break: \__xeCJK_zero_glue: }
797 }
798 \cs_new_protected_nopar:Npn \__xeCJK_zero_glue:
799 { \skip_horizontal:N \c_zero_skip }
(End definition for \xeCJK_FullLeft_and_Default:.)

```

\xeCJK_FullRight_and_CJK:

```

800 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_CJK:
801 {
802   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
803   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
804   \CJKglue
805 }
(End definition for \xeCJK_FullRight_and_CJK:.)

```

\xeCJK_FullRight_and_Default:

```

806 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Default:
807 {
808   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
809   \xeCJK_class_group_end:
810   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
811 }
(End definition for \xeCJK_FullRight_and_Default:.)

```

\xeCJK_Default_and_FullLeft:nN

```

812 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullLeft:nN #1#2
813 {
814   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#2}
815   \__xeCJK_Default_and_FullLeft_glue:N {#2}
816   \xeCJK_class_group_begin:
817   \xeCJK_select_font:
818   \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
819   \xeCJK_clear_Boundary_and_CJK_toks:
820   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
821   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2}
822   \CJKpunctsymbol {#2}
823 }
824 \cs_new_protected_nopar:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
825 { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_Default_and_FullLeft:nN.)

```

\xeCJK_CJK_and_FullLeft:N

```

826 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
827 {
828   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
829   \__xeCJK_CJK_and_FullLeft_glue:N {#1}
830   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
831   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
832   \CJKpunctsymbol {#1}
833 }
834 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
835 { \CJKglue \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_CJK_and_FullLeft:N.)

```

\xeCJK_Boundary_and_FullLeft:N

```

836 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullLeft:N #1
837 {
838   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
839   \__xeCJK_Boundary_and_FullLeft_glue:N {#1}
840   \xeCJK_class_group_begin:

```

```

841 \xeCJK_select_font:
842 \xeCJK_clear_Boundary_and_CJK_toks:
843 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
844 \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
845 \CJKpunctsymbol {#1}
846 }
847 \cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
848 {
849   \int_case:nnn { \etex_lastnodetype:D }
850   {
851     { \c_one } { \__xeCJK_zero_glue: }
852     { \c_minus_one } { }
853     { \c_eleven } { }
854     { \c_thirteen } { }
855   }
856   { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
857 }
(End definition for \xeCJK_Boundary_and_FullLeft:N.)

```

\xeCJK_Default_and_FullRight:nN

```

858 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullRight:nN #1#2
859 {
860   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#2}
861   \__xeCJK_Default_and_FullRight_glue:N {#2}
862   \xeCJK_class_group_begin:
863   \xeCJK_select_font:
864   \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
865   \xeCJK_clear_Boundary_and_CJK_toks:
866   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
867   \__xeCJK_punct_if_middle:NT {#2}
868   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2} }
869   \xeCJK_FullRight_symbol:N {#2}
870 }
(End definition for \xeCJK_Default_and_FullRight:nN.)

```

\xeCJK_Boundary_and_FullRight:N

```

871 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullRight:N #1
872 {
873   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
874   \__xeCJK_Default_and_FullRight_glue:N {#1}
875   \xeCJK_class_group_begin:
876   \xeCJK_select_font:
877   \xeCJK_clear_Boundary_and_CJK_toks:
878   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
879   \__xeCJK_punct_if_middle:NT {#1}
880   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
881   \xeCJK_FullRight_symbol:N {#1}
882 }
(End definition for \xeCJK_Boundary_and_FullRight:N.)

```

\xeCJK_CJK_and_FullRight:N

```

883 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
884 {
885   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
886   \__xeCJK_CJK_and_FullRight_glue:N {#1}
887   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
888   \__xeCJK_punct_if_middle:NT {#1}
889   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
890   \xeCJK_FullRight_symbol:N {#1}
891 }
(End definition for \xeCJK_CJK_and_FullRight:N.)

```



```

\__xeCJK_CJK_and_FullRight_glue:N
\__xeCJK_Default_and_FullRight_glue:N
892 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
893 {
894   \__xeCJK_punct_if_long:NTF {#1}
895     { \CJKglue }
896     {
897       \__xeCJK_punct_if_middle:NTF {#1}
898       { \xeCJK_no_break: \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1} }
899       { \xeCJK_no_break: \__xeCJK_zero_glue: }
900     }
901   }
902 \cs_new_eq:NN \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_CJK_and_FullRight_glue:N
(End definition for \__xeCJK_CJK_and_FullRight_glue:N and \__xeCJK_Default_and_FullRight_glue:N.)

```

```

\xeCJK_FullLeft_and_FullLeft:N
903 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
904 {
905   \xeCJK_no_break:
906   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
907   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
908   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
909   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
910   \CJKpunctsymbol {#1}
911 }
(End definition for \xeCJK_FullLeft_and_FullLeft:N.)

```

```

\xeCJK_FullLeft_and_FullRight:N
912 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
913 {
914   \xeCJK_no_break:
915   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
916   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
917   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
918   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
919   \xeCJK_no_break:
920   \xeCJK_FullRight_symbol:N {#1}
921 }
(End definition for \xeCJK_FullLeft_and_FullRight:N.)

```

```

\xeCJK_FullRight_and_FullLeft:N
922 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
923 {
924   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
925   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
926   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
927   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
928   \__xeCJK_punct_break:
929   \CJKpunctsymbol {#1}
930 }
(End definition for \xeCJK_FullRight_and_FullLeft:N.)

```

```

\xeCJK_FullRight_and_FullRight:N
931 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
932 {
933   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
934   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
935   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
936   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
937   \xeCJK_no_break:
938   \xeCJK_FullRight_symbol:N {#1}
939 }
(End definition for \xeCJK_FullRight_and_FullRight:N.)

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```
940 \keys_define:nn { xeCJK / options }
941 {
942   CheckFullRight .choice: ,
943   CheckFullRight / true .code:n =
944   {
945     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
946     {
947       \cs_set_eq:NN \__xeCJK_save_FullRight_check: \xeCJK_FullRight_and_Boundary:
948       \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N \xeCJK_FullRight_symbol:N
949       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
950       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \xeCJK_check_FullRight_symbol:Nw
951     }
952   } ,
953   CheckFullRight / false .code:n =
954   {
955     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
956     {
957       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \__xeCJK_save_FullRight_check:
958       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \__xeCJK_save_FullRight_symbol:N
959     }
960   } ,
961   CheckFullRight .default:n = { true }
962 }
```

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

\xeCJK_FullRight_symbol:N

```
963 \cs_new_nopar:Npn \xeCJK_FullRight_symbol:N { \CJKpunctsymbol }
```

(End definition for \xeCJK_FullRight_symbol:N.)

\xeCJK_check_FullRight:

```
964 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight:
965 {
966   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
967   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
968   \tl_case:Non \l_peek_token { \l__xeCJK_no_break_cs_case_tl } { }
969   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
970   \xeCJK_class_group_end:
971 }
972 \cs_generate_variant:Nn \tl_case:Nnn { No }
```

(End definition for \xeCJK_check_FullRight:.)

\xeCJK_check_FullRight_symbol:Nw

```
973 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight_symbol:Nw #1
974 { \xeCJK_peek_after_ignore_spaces:nw { \__xeCJK_save_FullRight_symbol:N {#1} } }
```

(End definition for \xeCJK_check_FullRight_symbol:Nw.)

\xeCJK_cs_case_keys_define:nNnn

```
975 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNnn #1#2#3#4#5
976 {
977   \tl_new:N #2
978   \seq_new:N #3
979   \keys_define:nn { xeCJK / options }
980   {
981     #1 .code:n =
982     {
983       \seq_set_split:Nnn #3 { } {##1}
984       \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
985     } ,
986     #1+ .code:n =
```

```

987     {
988       \tl_map_inline:nn {##1}
989       { \seq_if_in:NnF #2 {###1} { \seq_put_right:Nn #3 {###1} } }
990       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
991     } ,
992     #1- .code:n =
993     {
994       \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {###1} }
995       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
996     }
997   }
998 }
999 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
1000 {
1001   \tl_clear:N #1
1002   \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
1003   #4
1004 }
(End definition for \xeCJK_cs_case_keys_define:nNNnn.)

```

NoBreakCS 设置不能在全角右标点之后断行的控制序列。

```

1005 \xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
1006 \l__xeCJK_no_break_cs_case_tl \l__xeCJK_no_break_cs_seq { \xeCJK_no_break: } { }
(End definition for NoBreakCS. This function is documented on page 5.)

```

\xeCJKnobreak 为保险起见,我们在这里用了一个循环。

```

1007 \NewDocumentCommand \xeCJKnobreak { }
1008 {
1009   \int_zero:N \l__xeCJK_tmpa_int
1010   \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
1011   {
1012     \int_if_even:nTF \l__xeCJK_tmpa_int
1013     {
1014       \int_incr:N \l__xeCJK_tmpa_int
1015       \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
1016     }
1017     { \skip_add:Nn \l__xeCJK_tmp_skip \tex_lastskip:D }
1018     \tex_unskip:D
1019   }
1020   \xeCJK_no_break:
1021   \int_if_even:nF \l__xeCJK_tmpa_int
1022   { \skip_horizontal:N \l__xeCJK_tmp_skip }
1023 }
(End definition for \xeCJKnobreak. This function is documented on page 11.)

```

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```

1024 \keys_define:nn { xeCJK / options }
1025 {
1026   CheckSingle .choice: ,
1027   CheckSingle / true .code:n =
1028   {
1029     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1030     {
1031       \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
1032       \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1033     }
1034   } ,
1035   CheckSingle / false .code:n =
1036   {
1037     \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw

```

```

1038     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
1039   } ,
1040   CheckSingle      .default:n = { true } ,
1041   CJKchecksingle   .meta:n = { CheckSingle = true }
1042 }

```

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

\xeCJK_check_single:Nw

```

1043 \cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
1044 {
1045   \peek_catcode:NTF \c_catcode_other_token
1046   { \xeCJK_check_single:NNw #1 }
1047   {
1048     \bool_if:nTF
1049     {
1050       \xeCJK_if_blank_x_p:n { \token_get_arg_spec:N \l_peek_token } &&
1051       \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
1052       \exp_after:wN \token_if_other_p:N \l_peek_token
1053     }
1054     { \exp_after:wN \xeCJK_check_single:NNw \exp_after:wN #1 }
1055     { \__xeCJK_check_single_save:N #1 }
1056   }
1057 }

```

(End definition for \xeCJK_check_single:Nw.)

\xeCJK_check_single:NNw 使用 \group_align_safe_begin: 和 \group_align_safe_end: 是为了防止在表格里面报错。

```

1058 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
1059 {
1060   \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_other_token
1061   {
1062     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1063     { \__xeCJK_check_single_space:NN #1#2 }
1064     { \__xeCJK_check_single_save:N #1 #2 }
1065   }
1066   {
1067     \group_align_safe_begin:
1068     \token_if_cs:NTF \l_peek_token
1069     {
1070       \group_align_safe_end:
1071       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1072       { \xeCJK_check_single_cs:NNn #1#2 { ~ } }
1073       { \xeCJK_check_single_cs:NNn #1#2 { } }
1074     }
1075     {
1076       \group_align_safe_end:
1077       \bool_if:nTF
1078       {
1079         \l__xeCJK_plain_equation_bool &&
1080         \token_if_math_toggle_p:N \l_peek_token
1081       }
1082       {
1083         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1084         { \xeCJK_check_single_equation:NNnNw #1 #2 { ~ } }
1085         { \xeCJK_check_single_equation:NNnNw #1 #2 { } }
1086       }
1087       {
1088         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1089         { \__xeCJK_check_single_save:N #1 #2 ~ }
1090         { \__xeCJK_check_single_save:N #1 #2 }
1091       }
1092     }
1093   }
1094 }

```

(End definition for \xeCJK_check_single:NNw.)

_xeCJK_check_single_space:NN

```
1095 \cs_new_protected_nopar:Npn \_xeCJK_check_single_space:NN #1#2
1096 {
1097   \xeCJK_if_CJK_class:NTF #2
1098   {
1099     \xeCJK_if_CJK_class:NTF \l_peek_token
1100     { \_xeCJK_check_single_save:N #1 #2 }
1101     { \_xeCJK_check_single_save:N #1 #2 ~ }
1102   }
1103   { \_xeCJK_check_single_save:N #1 #2 ~ }
1104 }
(End definition for \_xeCJK_check_single_space:NN.)
```

\xeCJK_check_single_equation:NNnNw

```
1105 \cs_new_protected_nopar:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
1106 {
1107   \peek_catcode:NTF \c_math_toggle_token
1108   {
1109     \xeCJK_no_break: \_xeCJK_check_single_save:N #1
1110     \xeCJK_no_break: #2 #4
1111   }
1112   { \_xeCJK_check_single_save:N #1 #2#3#4 }
1113 }
(End definition for \xeCJK_check_single_equation:NNnNw.)
```

\xeCJK_check_single_cs:NNn

```
1114 \cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
1115 {
1116   \tl_case:Non \l_peek_token
1117   { \l__xeCJK_check_single_cs_case_tl }
1118   { \use_iii:nnn }
1119   { \xeCJK_check_single_env:nnNn }
1120   {
1121     \xeCJK_no_break: \_xeCJK_check_single_save:N #1
1122     \xeCJK_no_break: #2
1123   }
1124   { \_xeCJK_check_single_save:N #1 #2#3 }
1125 }
1126 \tl_new:N \l__xeCJK_check_single_cs_case_tl
(End definition for \xeCJK_check_single_cs:NNn.)
```

\xeCJK_check_single_env:nnNn

```
1127 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
1128 {
1129   \str_case:x:non {#4}
1130   { \l__xeCJK_inline_env_case_tl }
1131   { \use_i:nn }
1132   {#1} {#2}
1133   \scan_stop: #3 {#4}
1134 }
1135 \cs_generate_variant:Nn \str_case:x:nnn { no }
(End definition for \xeCJK_check_single_env:nnNn.)
```

NewLineCS

```
1136 \xeCJK_cs_case_keys_define:nnNnn { NewLineCS }
1137 \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq { \use_ii:nnn }
1138 {
1139   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1140   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1141 }
```

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

EnvCS

```

1142 \xeCJK_cs_case_keys_define:nNNnn { EnvCS }
1143 \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq { \use:n }
1144 {
1145   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1146   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1147 }

```

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

InlineEnv

```

1148 \keys_define:nn { xeCJK / options }
1149 {
1150   InlineEnv .code:n =
1151   {
1152     \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
1153     \__xeCJK_update_inline_env_case_tl:
1154   } ,
1155   InlineEnv+ .code:n =
1156   {
1157     \clist_map_inline:nn {#1}
1158     {
1159       \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
1160       { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
1161     }
1162     \__xeCJK_update_inline_env_case_tl:
1163   } ,
1164   InlineEnv- .code:n =
1165   {
1166     \clist_map_inline:nn {#1}
1167     { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
1168     \__xeCJK_update_inline_env_case_tl:
1169   }
1170 }
1171 \seq_new:N \l__xeCJK_inline_env_seq

```

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

__xeCJK_update_inline_env_case_tl:

```

1172 \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
1173 {
1174   \tl_clear:N \l__xeCJK_inline_env_case_tl
1175   \seq_map_inline:Nn \l__xeCJK_inline_env_seq
1176   { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { \use_ii:nn } } }
1177 }
1178 \tl_new:N \l__xeCJK_inline_env_case_tl

```

(End definition for __xeCJK_update_inline_env_case_tl:.)

PlainEquation

```

1179 \keys_define:nn { xeCJK / options }
1180 { PlainEquation .bool_set:N = \l__xeCJK_plain_equation_bool }

```

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

5.9 增加 CJK 子分区

\g__xeCJK_CJK_sub_class_seq

```

1181 \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。

```
1182 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
1183 { s > { \TrimSpaces } m > { \TrimSpaces } m }
1184 {
1185   \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
1186   \IfBooleanT {#1} { \xeCJKResetPunctClass }
1187 }
1188 \@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`

```
1189 \bool_new:N \l__xeCJK_sub_cancel_bool
1190 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
1191 {
1192   \bool_if:NF \l__xeCJK_sub_cancel_bool
1193   {
1194     \bool_set_true:N \l__xeCJK_sub_cancel_bool
1195     \__xeCJK_sub_restore_or_cancel:x {#2}
1196     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1197   }
1198 }
1199 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
1200 {
1201   \bool_if:NT \l__xeCJK_sub_cancel_bool
1202   {
1203     \bool_set_false:N \l__xeCJK_sub_cancel_bool
1204     \__xeCJK_sub_restore_or_cancel:x {#2}
1205     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1206   }
1207 }
(End definition for \xeCJKCancelSubCJKBlock and \xeCJKRestoreSubCJKBlock. These functions are documented on page 9.)
```

`__xeCJK_sub_restore_or_cancel:n`

```
1208 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
1209 {
1210   \clist_map_inline:nn {#1}
1211   {
1212     \int_if_exist:CTF { \__xeCJK_class_csname:n { CJK/##1 } }
1213     {
1214       \xeCJK_declare_char_class:nx
1215       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
1216       { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
1217     }
1218     { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
1219   }
1220 }
1221 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
1222 \__xeCJK_msg_new:nn { SubBlock-undefined }
1223 {
1224   The~CJK~sub~block~`#1'~is~undefined.\\\
1225   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
1226   to~declare~it.
1227 }
(End definition for \__xeCJK_sub_restore_or_cancel:n.)
```

`\xeCJK_declare_sub_char_class:nnn`

```
1228 \cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
1229 {
1230   \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
1231   {
1232     \xeCJK_new_class:n { #1/#2 }
1233     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
1234   }
1235 }
```

```

1234 \xeCJK_new_sub_key:n {#2}
1235 }
1236 \xeCJK_declare_char_class:nn { #1/#2 } {#3}
1237 }
1238 \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

```

1239 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
1240 {
1241   \seq_map_inline:Nn \g__xeCJK_base_class_seq
1242   {
1243     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
1244     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
1245     \str_if_eq:nnTF {##1} { CJK }
1246     {
1247       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
1248       { \__xeCJK_switch_font:nn {#1} {#2} }
1249     }
1250     {
1251       \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
1252       { \CJKsymbol }
1253       { \__xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
1254     }
1255   }
1256   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
1257   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
1258   {
1259     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
1260     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
1261     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
1262     { \__xeCJK_switch_font:nn {#2} {##1} }
1263     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
1264     { \__xeCJK_switch_font:nn {##1} {#2} }
1265   }
1266   \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
1267   \__xeCJK_save_CJK_class:n { #1/#2 }
1268   \clist_map_inline:nn { CJK , FullLeft , FullRight }
1269   {
1270     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
1271     { \__xeCJK_switch_font:nn {#2} {#1} }
1272   }
1273 }
(End definition for \__xeCJK_set_sub_class_toks:nn.)

```

5.10 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。

```

1274 \__xeCJK_msg_new:nn { XeTeX-too-old }
1275 {
1276   \token_to_str:N \XeTeXglyphbounds \ is~not~defined.\
1277   CJK~punctuation~kerning~will~not~be~available.\
1278   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
1279 }
1280 \cs_if_exist:NF \XeTeXglyphbounds
1281 {
1282   \__xeCJK_error:n { XeTeX-too-old }
1283   \__xeCJK_after_end_preamble:n { \xeCJKsetup { PunctStyle = plain } }
1284 }

```


\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```
1285 \NewDocumentCommand \xeCJKsetwidth { m m }
1286 { \tl_map_inline:xn {#1} { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#2} } }
1287 \cs_generate_variant:Nn \tl_map_inline:nn { x }
(End definition for \xeCJKsetwidth. This function is documented on page 9.)
```

\xeCJKsetkern 手动设置相邻标点的距离。

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

```
1290 \tl_const:Nn \c__xeCJK_left_tl { left }
1291 \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

```
1292 \keys_define:nn { xeCJK / options }
1293 {
1294   AllowBreakBetweenPuncts .choice: ,
1295   AllowBreakBetweenPuncts / true .code:n =
1296     { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_allow_break: } ,
1297   AllowBreakBetweenPuncts / false .code:n =
1298     { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_no_break: } ,
1299   AllowBreakBetweenPuncts .default:n = { true } ,
1300   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
1301   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
1302   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
1303   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
1304   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
1305   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
1306   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
1307   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
1308   MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
1309   PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl
1310 }
```

(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

```
1311 \clist_new:N \g__xeCJK_special_punct_clist
1312 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
1313 \cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
1314 \cs_new_nopar:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
1315 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1316 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
1317 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1318 {
1319   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
1320   { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
1321   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
1322   \tl_map_inline:xn {#2}
1323   {
1324     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1325     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1326   }
1327 }
1328 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1329 {
1330   \tl_map_inline:xn {#2}
1331   {
1332     \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
1333     {
```

```

1334         \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1335         \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1336     }
1337 }
1338 }
1339 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1340 {
1341     \tl_map_inline:xn {#2}
1342     {
1343         \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1344         \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1345     }
1346 }
(End definition for \g__xeCJK_special_punct_clist and others.)

```

```

\__xeCJK_punct_if_right_p:N
\__xeCJK_punct_if_long_p:N 1347 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
\__xeCJK_punct_if_middle_p:N 1348 {
    \__xeCJK_punct_if_mixed_width_p:N 1349 \if_int_compare:w \xeCJK_token_value_class:N #1 = \xeCJK_class_num:n { FullRight }
    \__xeCJK_punct_if_right:NTF 1350 \prg_return_true: \else: \prg_return_false: \fi:
    \__xeCJK_punct_if_long:NTF 1351 }
    \__xeCJK_punct_if_middle:NTF 1352 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
    \__xeCJK_punct_if_mixed_width:NTF 1353 {
        \exp_args:Nc 1354 \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
        \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end: 1355
        \prg_return_true: \else: \prg_return_false: \fi: 1356
    } 1357
} 1358
1359 }
1360 }
(End definition for \__xeCJK_punct_if_right:N and others.)

```

```

\__xeCJK_punct_dim_csname:nn 一些用于记录的辅助函数。
    \__xeCJK_punct_dim_csname:nnn 1361 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
    \__xeCJK_punct_dim_csname:nnnn 1362 { g__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
    \__xeCJK_use_punct_dim:nnn 1363 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn
    \__xeCJK_use_punct_dim:nn 1364 { \__xeCJK_punct_dim_csname:nnnn { \l_xeCJK_punct_style_tl } }
    \__xeCJK_gset_punct_dim:nnx 1365 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnnn #1#2#3#4
    \__xeCJK_gset_punct_dim:nnnx 1366 { g__\l_xeCJK_current_font_tl/#1/#2/#3/#4/tl }
    \__xeCJK_gset_punct_dim:nnnnx 1367 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
    1368 { \tl_use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
    1369 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
    1370 { \tl_use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
    1371 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnx #1#2#3
    1372 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } {#3} }
    1373 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnx #1#2#3#4
    1374 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } {#4} }
    1375 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnnx #1#2#3#4#5
    1376 { \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。

```

1377 \DeclareObjectType { xeCJK / punctuation } { \c_three }
1378 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_three }
1379 {
1380     enabled-global-setting : boolean = true ,
1381     fixed-punct-width : length = \c_max_dim ,
1382     fixed-punct-ratio : real = \c_one_fp ,
1383     mixed-punct-width : length = \KeyValue { fixed-punct-width } ,
1384     mixed-punct-ratio : real = \KeyValue { fixed-punct-ratio } ,
1385     middle-punct-width : length = \KeyValue { fixed-punct-width } ,
1386     middle-punct-ratio : real = \KeyValue { fixed-punct-ratio } ,

```

```

1387     fixed-margin-width      : length = \c_max_dim ,
1388     fixed-margin-ratio      : real   = \c_one_fp ,
1389     mixed-margin-width      : length = \KeyValue { fixed-margin-width } ,
1390     mixed-margin-ratio      : real   = \KeyValue { fixed-margin-ratio } ,
1391     middle-margin-width     : length = \KeyValue { fixed-margin-width } ,
1392     middle-margin-ratio     : real   = \KeyValue { fixed-margin-ratio } ,
1393     add-min-bound-to-margin : boolean = false ,
1394     optimize-margin         : boolean = false ,
1395     margin-minimum          : length = \c_zero_dim ,
1396     enabled-kerning         : boolean = true ,
1397     min-bound-to-kerning    : boolean = false ,
1398     kerning-total-width     : length = \c_max_dim ,
1399     kerning-total-ratio     : real   = 0.75 ,
1400     optimize-kerning        : boolean = false ,
1401     same-align-margin       : length = \c_max_dim ,
1402     same-align-ratio        : real   = \c_zero_fp ,
1403     different-align-margin  : length = \c_max_dim ,
1404     different-align-ratio   : real   = \c_zero_fp ,
1405     kerning-margin-width    : length = \c_max_dim ,
1406     kerning-margin-ratio    : real   = \c_one_fp ,
1407     kerning-margin-minimum  : length = \c_zero_dim
1408 }

1409 \DeclareTemplateCode { xeCJK / punctuation } { basic } { \c_three }
1410 {
1411     enabled-global-setting = \l__xeCJK_enabled_global_setting_bool ,
1412     fixed-punct-width     = \l__xeCJK_fixed_punct_width_dim ,
1413     fixed-punct-ratio     = \l__xeCJK_fixed_punct_ratio_fp ,
1414     mixed-punct-width     = \l__xeCJK_mixed_punct_width_dim ,
1415     mixed-punct-ratio     = \l__xeCJK_mixed_punct_ratio_fp ,
1416     middle-punct-width    = \l__xeCJK_middle_punct_width_dim ,
1417     middle-punct-ratio    = \l__xeCJK_middle_punct_ratio_fp ,
1418     fixed-margin-width    = \l__xeCJK_fixed_margin_width_dim ,
1419     fixed-margin-ratio    = \l__xeCJK_fixed_margin_ratio_fp ,
1420     mixed-margin-width    = \l__xeCJK_mixed_margin_width_dim ,
1421     mixed-margin-ratio    = \l__xeCJK_mixed_margin_ratio_fp ,
1422     middle-margin-width   = \l__xeCJK_middle_margin_width_dim ,
1423     middle-margin-ratio   = \l__xeCJK_middle_margin_ratio_fp ,
1424     add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
1425     optimize-margin       = \l__xeCJK_optimize_margin_bool ,
1426     margin-minimum        = \l__xeCJK_margin_minimum_dim ,
1427     enabled-kerning       = \l__xeCJK_enabled_kerning_bool ,
1428     min-bound-to-kerning  = \l__xeCJK_min_bound_to_kerning_bool ,
1429     kerning-total-width   = \l__xeCJK_kerning_total_width_dim ,
1430     kerning-total-ratio   = \l__xeCJK_kerning_total_ratio_fp ,
1431     optimize-kerning      = \l__xeCJK_optimize_kerning_bool ,
1432     same-align-margin     = \l__xeCJK_same_align_margin_dim ,
1433     same-align-ratio      = \l__xeCJK_same_align_ratio_fp ,
1434     different-align-margin = \l__xeCJK_different_align_margin_dim ,
1435     different-align-ratio = \l__xeCJK_different_align_ratio_fp ,
1436     kerning-margin-width  = \l__xeCJK_kerning_margin_width_dim ,
1437     kerning-margin-ratio  = \l__xeCJK_kerning_margin_ratio_fp ,
1438     kerning-margin-minimum = \l__xeCJK_kerning_margin_minimum_dim
1439 }
1440 {
1441     \AssignTemplateKeys
1442     \tl_if_blank:nTF {#3}
1443     { \xeCJK_punct_margin_process:NN {#1} {#2} }
1444     { \xeCJK_punct_kerning_process:NN {#2} {#3} }
1445 }

```

\xeCJK_punct_margin_process:NN

1446 \cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2

```

1447 {
1448   \tl_set:Nx \l__xeCJK_tmpa_tl { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
1449   \tl_set:Nx \l__xeCJK_tmpb_tl
1450   {
1451     \__xeCJK_use_punct_dim:nnn { bound }
1452     { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl } {#2}
1453   }
1454   \dim_set:Nn \l__xeCJK_tmp_dim
1455   {
1456     \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1457     {
1458       \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
1459       { \use_none:n }
1460       {
1461         \xeCJK_if_blank_x:nTF \g__xeCJK_punct_width_tl
1462         { \use:n }
1463         { \g__xeCJK_punct_width_tl \use_none:n }
1464       }
1465     }
1466     { \use:n }
1467     {
1468       \__xeCJK_punct_if_middle:NTF {#2}
1469       { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
1470       {
1471         \__xeCJK_punct_if_mixed_width:NTF {#2}
1472         { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
1473         { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }
1474       }
1475     }
1476   }
1477   \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2}
1478   {
1479     \dim_max:nn
1480     { \l__xeCJK_margin_minimum_dim }
1481     {
1482       \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1483       {
1484         \__xeCJK_punct_if_middle:NTF {#2}
1485         {
1486           (
1487             \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1488           ) / \c_two
1489         }
1490         {
1491           \bool_if:NTF \l__xeCJK_optimize_margin_bool
1492           { \dim_max:nn { \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl } }
1493           { \use:n }
1494           {
1495             \l__xeCJK_tmp_dim - ( \l__xeCJK_tmpb_tl )
1496             - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1497           }
1498         }
1499       }
1500     }
1501     \bool_if:NTF \l__xeCJK_optimize_margin_bool
1502     { \dim_min:nn { \l__xeCJK_tmpa_tl } }
1503     { \use:n }
1504     {
1505       \__xeCJK_punct_if_middle:NTF {#2}
1506       {
1507         \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
1508         { \l__xeCJK_middle_margin_width_dim }

```

```

1509         {
1510             \fp_use:N \l__xeCJK_middle_margin_ratio_fp
1511             \etex_dimexpr:D
1512                 ( \l__xeCJK_tmpa_tl + \l__xeCJK_tmpl_tl ) / \c_two
1513             \scan_stop:
1514         }
1515     }
1516     {
1517         \__xeCJK_punct_if_mixed_width:NTF {#2}
1518         { \__xeCJK_margin_width_or_ratio:n { mixed } }
1519         { \__xeCJK_margin_width_or_ratio:n { fixed } }
1520     }
1521 }
1522 }
1523 }
1524 }
1525 }
(End definition for \xeCJK_punct_margin_process:NN.)

```

__xeCJK_punct_width_or_ratio:nN

```

1526 \cs_new_nopar:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
1527 {
1528     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
1529     { \use:c { l__xeCJK_#1_punct_width_dim } }
1530     {
1531         \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } = \c_zero_fp
1532         { \c_max_dim }
1533         {
1534             \fp_use:c { l__xeCJK_#1_punct_ratio_fp }
1535             \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
1536         }
1537     }
1538 }
(End definition for \__xeCJK_punct_width_or_ratio:nN.)

```

__xeCJK_margin_width_or_ratio:n

```

1539 \cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
1540 {
1541     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
1542     { \use:c { l__xeCJK_#1_margin_width_dim } }
1543     {
1544         \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
1545         \etex_dimexpr:D \l__xeCJK_tmpa_tl \scan_stop:
1546     }
1547     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
1548     { + \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpl_tl }
1549 }
(End definition for \__xeCJK_margin_width_or_ratio:n.)

```

\xeCJK_punct_kerning_process:NN

```

1550 \cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
1551 {
1552     \__xeCJK_gset_punct_dim:nnn { kern } {#1} {#2}
1553     {
1554         \dim_eval:n
1555         {
1556             \bool_if:nTF
1557             {
1558                 \l__xeCJK_enabled_global_setting_bool &&
1559                 \tl_if_exist_p:c { g__xeCJK_punct/kern/#1/#2/tl }
1560             }
1561             { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }

```

```

1562         {
1563             \bool_if:NTF \l__xeCJK_enabled_kerning_bool
1564             { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
1565             { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
1566         }
1567     - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1568     - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} )
1569 }
1570 }
1571 }

```

(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。

```

1572 \cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
1573 {
1574     \dim_eval:n
1575     {
1576         \__xeCJK_use_punct_dim:nnn
1577         { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
1578         { \c__xeCJK_right_tl } {#1} +
1579         \__xeCJK_use_punct_dim:nnn
1580         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1581         { \c__xeCJK_left_tl } {#2}
1582     }
1583 }

```

(End definition for __xeCJK_original_kerning_margin:NN.)

_xeCJK_calc_kerning_margin:NN

```

1584 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:NN #1#2
1585 {
1586     \dim_max:nn
1587     { \l__xeCJK_kerning_margin_minimum_dim }
1588     {
1589         \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
1590         { \__xeCJK_punct_min_bound:NN {#1} {#2} }
1591         {
1592             \bool_if:NTF \l__xeCJK_optimize_kerning_bool
1593             { \dim_max:nn { \__xeCJK_punct_min_bound:NN {#1} {#2} } }
1594             { \use:n }
1595             {
1596                 \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
1597                 { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
1598                 {
1599                     \fp_compare:nNnTF \l__xeCJK_kerning_total_ratio_fp = \c_zero_fp
1600                     {
1601                         \xeCJK_if_same_class:NNTF {#1} {#2}
1602                         { \__xeCJK_kerning_width_or_ratio:nNN { same } }
1603                         { \__xeCJK_kerning_width_or_ratio:nNN { different } }
1604                     }
1605                     {
1606                         \__xeCJK_calc_kerning_margin:nNN
1607                         {
1608                             \fp_use:N \l__xeCJK_kerning_total_ratio_fp
1609                             \etex_dimexpr:D
1610                             \__xeCJK_use_punct_dim:nn { width } {#1} +
1611                             \__xeCJK_use_punct_dim:nn { width } {#2}
1612                             \scan_stop:
1613                         }
1614                     }
1615                 }
1616                 {#1} {#2}
1617             }

```

```

1618     }
1619   }
1620 }
(End definition for \__xeCJK_calc_kerning_margin:NN.)

```

_xeCJK_kerning_width_or_ratio:nnN

```

1621 \cs_new_nopar:Npn \__xeCJK_kerning_width_or_ratio:nnN #1#2#3
1622 {
1623   \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
1624     { \use:c { l__xeCJK_#1_align_margin_dim } \use_none:nn }
1625     {
1626       \fp_compare:nNnTF { \use:c { l__xeCJK_#1_align_ratio_fp } } = \c_zero_fp
1627         { \use:n }
1628         { \fp_use:c { l__xeCJK_#1_align_ratio_fp } \use_ii:nn }
1629     }
1630   {
1631     \dim_compare:nNnTF { l__xeCJK_kerning_margin_width_dim } < \c_max_dim
1632       { \l__xeCJK_kerning_margin_width_dim \use_none:n }
1633       { \fp_use:N { l__xeCJK_kerning_margin_ratio_fp } \use:n }
1634   }
1635   { \etex_dimexpr:D \__xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
1636 }
(End definition for \__xeCJK_kerning_width_or_ratio:nnN.)

```

_xeCJK_punct_min_bound:NN

```

1637 \cs_new_nopar:Npn \__xeCJK_punct_min_bound:NN #1#2
1638 {
1639   \dim_max:nn
1640   {
1641     \dim_min:nn
1642     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} }
1643     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
1644   }
1645   {
1646     \dim_min:nn
1647     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} }
1648     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
1649   }
1650 }
(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。

```

1651 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:nnN #1#2#3
1652 {
1653   \dim_eval:n
1654   {
1655     (#1)
1656     - ( \__xeCJK_use_punct_dim:nnn
1657         { \__xeCJK_punct_if_right:NNTF {#2} { bound } { glue } }
1658         { \c__xeCJK_left_tl } {#2} )
1659     - ( \__xeCJK_use_punct_dim:nnn
1660         { \__xeCJK_punct_if_right:NNTF {#3} { glue } { bound } }
1661         { \c__xeCJK_right_tl } {#3} )
1662     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1663     - ( \__xeCJK_use_punct_dim:nn { dimen } {#3} )
1664   }
1665 }
(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。

```

1666 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
1667 {

```



```

1668 \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }
1669 {
1670   \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1671   {
1672     \__xeCJK_gset_punct_dim:nnx { glue } {#1} {#2} { \c_zero_dim }
1673     \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
1674     \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
1675   }
1676   {
1677     { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
1678     \__xeCJK_punct_if_long:NTF {#2}
1679     { \__xeCJK_gset_punct_dim:nnx { glue } {#1} {#2} { \c_zero_dim } }
1680     {
1681       \UseInstance { xeCJK / punctuation }
1682       { \l_xeCJK_punct_style_tl } {#1} {#2} { }
1683     }
1684   }
1685 }
1686 }

```

(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。

```

1687 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
1688 {
1689   \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_left_tl {#1}
1690   { \xeCJK_glyph_bounds:NN \c_one {#1} }
1691   \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_right_tl {#1}
1692   { \xeCJK_glyph_bounds:NN \c_three {#1} }
1693   \dim_set:Nn \l__xeCJK_tmp_dim
1694   {
1695     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} ) +
1696     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1697   }
1698   \__xeCJK_gset_punct_dim:nnx { width } {#1}
1699   { \dim_use:N \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
1700   \__xeCJK_gset_punct_dim:nnx { dimen } {#1}
1701   { \dim_eval:n { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim } }
1702   \__xeCJK_punct_if_long:NT {#1}
1703   {
1704     \seq_map_inline:Nn \g__xeCJK_punct_style_seq
1705     {
1706       \__xeCJK_gset_punct_dim:nnnnx {##1} { kern } {#1} {#1}
1707       {
1708         \bool_if:nTF
1709         {
1710           \str_if_eq_p:nn {#1} { ^^^^2025 } ||
1711           \str_if_eq_p:nn {#1} { ^^^^2026 }
1712         }
1713         { \c_zero_dim }
1714         { - \dim_use:N \l__xeCJK_tmp_dim }
1715       }
1716     }
1717   }
1718 }
1719 \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。

```

1720 \cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
1721 {
1722   \dim_use:N \XeTeXglyphbounds #1 ~

```

```

1723 \XeTeXcharglyph \xeCJK_token_value_charcode:N #2 \exp_stop_f:
1724 }
(End definition for \xeCJK_glyph_bounds:NN.)

```

`\xeCJK_get_punct_kerning:NN` 标点挤压。

```

1725 \cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
1726 {
1727   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
1728   {
1729     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1730     { \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#2} { \c_zero_dim } }
1731     {
1732       \UseInstance { xeCJK / punctuation }
1733       { \l_xeCJK_punct_style_tl } { } {#1} {#2}
1734     }
1735   }
1736 }
1737 \cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
(End definition for \xeCJK_get_punct_kerning:NN.)

```

PunctStyle

```

1738 \keys_define:nn { xeCJK / options }
1739 {
1740   PunctStyle .choice: ,
1741   PunctStyle .default:n = { quanjiao } ,
1742   PunctStyle / halfwidth .meta:n = { PunctStyle = banjiao } ,
1743   PunctStyle / fullwidth .meta:n = { PunctStyle = quanjiao } ,
1744   PunctStyle / mixedwidth .meta:n = { PunctStyle = kaiming } ,
1745   PunctStyle / marginkerning .meta:n = { PunctStyle = hangmobanjiangiao } ,
1746   PunctStyle / plain .code:n =
1747   { \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl } ,
1748   PunctStyle / unknown .code:n =
1749   {
1750     \IfInstanceExistTF { xeCJK / punctuation } { \l_keys_value_tl }
1751     { \tl_set:Nx \l_xeCJK_punct_style_tl { \l_keys_value_tl } }
1752     { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
1753   }
1754 }
1755 \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
1756 \__xeCJK_msg_new:nn { punct-style-unknown }
1757 {
1758   Punctuation~style~"#1"~is~unknown. \\\
1759   The~available~styles~are~listed~as~follow.\\\
1760   "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
1761 }
(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。

```

1762 \NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
1763 {
1764   \IfInstanceExistTF { xeCJK / punctuation } {#1}
1765   { \__xeCJK_warning:nx { punct-style-already-defined } {#1} }
1766   { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
1767   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
1768 }
1769 \seq_new:N \g__xeCJK_punct_style_seq
1770 \__xeCJK_msg_new:nn { punct-style-already-defined }
1771 {
1772   Punctuation~style~"#1"~is~already~defined!. \\\
1773   The~existing~style~of~"#1"~will~be~overwritten.\\
1774 }
1775 \@onlypreamble \xeCJKDeclarePunctStyle

```

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

\xeCJKEditPunctStyle 对已有的标点处理风格进行修改。

```
1776 \NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
1777 {
1778   \IfInstanceExistTF { xeCJK / punctuation } {#1}
1779   { \EditInstance { xeCJK / punctuation } {#1} {#2} }
1780   { \__xeCJK_error:nx { punct-style-unknown } {#1} }
1781 }
1782 \@onlypreamble \xeCJKEditPunctStyle
```

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

默认设置即为全角格式。

```
1783 \xeCJKDeclarePunctStyle { quanjiao } { }
1784 \xeCJKDeclarePunctStyle { hangmobanjiang } { enabled-kerning = false }
1785 \xeCJKDeclarePunctStyle { banjiao }
1786 {
1787   fixed-punct-ratio    = 0.5 ,
1788   optimize-margin      = true ,
1789   kerning-total-ratio  = 0.5 ,
1790   optimize-kerning     = true
1791 }
1792 \xeCJKDeclarePunctStyle { kaiming }
1793 {
1794   fixed-punct-ratio    = 0.5 ,
1795   mixed-punct-ratio    = 0.8 ,
1796   optimize-margin      = true ,
1797   kerning-total-ratio  = 0.5 ,
1798   optimize-kerning     = true
1799 }
1800 \xeCJKDeclarePunctStyle { CCT }
1801 {
1802   fixed-punct-ratio    = 0.7 ,
1803   optimize-margin      = true ,
1804   kerning-total-ratio  = 0.6 ,
1805   optimize-kerning     = true
1806 }
```

5.11 后备字体

AutoFallBack 后备字体的宏包选项声明。

```
1807 \keys_define:nn { xeCJK / options }
1808 {
1809   AutoFallBack .choice: ,
1810   AutoFallBack / true .code:n =
1811   {
1812     \cs_if_eq:NNF \CJKsymbol \xeCJK_fallback_test_glyph:N
1813     {
1814       \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1815       \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
1816     }
1817   } ,
1818   AutoFallBack / false .code:n =
1819   {
1820     \cs_if_eq:NNT \CJKsymbol \xeCJK_fallback_test_glyph:N
1821     { \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N }
1822   } ,
1823   AutoFallBack .default:n = { true } ,
1824   fallback .meta:n = { AutoFallBack = true }
1825 }
```

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

\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。

```
1826 \cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
1827 {
1828   \xeCJK_glyph_if_exist:NTF {#1}
1829   { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1830   { { \xeCJK_fallback_loop:Nn {#1} { \l_xeCJK_family_tl/FallBack } } }
1831 }
```

(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。

```
1832 \cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
1833 {
1834   \xeCJK_family_if_exist:XTF {#2}
1835   {
1836     \xeCJK_select_font:x {#2}
1837     \xeCJK_glyph_if_exist:NTF {#1}
1838     { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1839     { \xeCJK_fallback_loop:Nn {#1} { #2/FallBack } }
1840   }
1841   {
1842     \xeCJK_family_if_exist:XT { \l_xeCJK_family_tl/FallBack }
1843     { \__xeCJK_warning:nxx { fallback } {#1} {#2} }
1844     \__xeCJK_fallback_save_CJKsymbol:N {#1}
1845   }
1846 }
1847 \__xeCJK_msg_new:nn { fallback }
1848 {
1849   CJKfamily~`#2'~
1850   ( \prop_get:No \g__xeCJK_family_font_name_prop {#2} )~
1851   does~not~contain~glyph~`#1'~(U+\int_to_hexadecimal:n {`#1}).\\
1852 }
```

```
1853 \bool_new:N \l__xeCJK_fallback_first_bool
1854 \cs_generate_variant:Nn \prop_get:Nn { No }
```

(End definition for \xeCJK_fallback_loop:Nn.)

\setCJKfallbackfamilyfont

```
1855 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{ } m }
1856 {
1857   \clist_set:Nx \l__xeCJK_tmpa_clist {#3}
1858   \int_compare:nNtF { \clist_count:N \l__xeCJK_tmpa_clist } > \c_one
1859   { \xeCJK_set_family_fallback:xxN {#1} {#2} \l__xeCJK_tmpa_clist }
1860   { \xeCJK_set_family:xxx { #1/FallBack } {#2} {#3} }
1861 }
```

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

\xeCJK_set_family_fallback:nnN

```
1862 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnN #1#2#3
1863 {
1864   \group_begin:
1865   \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
1866   \prop_get:NVNF \g__xeCJK_family_font_name_prop
1867     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
1868     { \tl_clear:N \l__xeCJK_font_name_tl }
1869   \clist_map_inline:Nn #3
1870   {
1871     \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
1872     \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
1873     \clist_put_left:cn
1874       { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist } {#2}
1875     \xeCJK_set_family:Vvv \l__xeCJK_fallback_family_tl
1876       { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist }
```

```

1877         { l__xeCJK_ \l__xeCJK_fallback_family_tl _font_name_tl }
1878     }
1879     \group_end:
1880 }
1881 \cs_generate_variant:Nn \xeCJK_set_family_fallback:nnN { xx }
(End definition for \xeCJK_set_family_fallback:nnN.)

```

5.12 CJK 字体族声明方式

```

1882 \bool_new:N \g__xeCJK_auto_fake_bold_bool
1883 \bool_new:N \g__xeCJK_auto_fake_slant_bool
1884 \fp_new:N \g__xeCJK_embolden_factor_fp
1885 \fp_new:N \g__xeCJK_slant_factor_fp

```

AutoFakeBold
AutoFakeSlant
EmboldenFactor
SlantFactor

伪粗体和伪斜体的宏包选项声明。

```

1886 \keys_define:nn { xeCJK / options }
1887 {
1888     AutoFakeBold .choices:nn = { true , false }
1889     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,
1890     AutoFakeBold / unknown .code:n =
1891     {
1892         \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
1893         \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1894     } ,
1895     AutoFakeBold .default:n = { true } ,
1896     AutoFakeSlant .choices:nn = { true , false }
1897     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
1898     AutoFakeSlant / unknown .code:n =
1899     {
1900         \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
1901         \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
1902     } ,
1903     AutoFakeSlant .default:n = { true } ,
1904     EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
1905     SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
1906     BoldFont .meta:n = { AutoFakeBold = true } ,
1907     boldfont .meta:n = { AutoFakeBold = true } ,
1908     SlantFont .meta:n = { AutoFakeSlant = true } ,
1909     slantfont .meta:n = { AutoFakeSlant = true }
1910 }

```

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

\xeCJK_new_sub_key:n
\g__xeCJK_sub_key_seq

用于定义 CJK 子区字体和备用字体的选项。

```

1911 \seq_new:N \g__xeCJK_sub_key_seq
1912 \cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
1913 {
1914     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
1915     \keys_define:nn { xeCJK / features }
1916     {
1917         #1 .code:n =
1918         {
1919             \tl_if_blank:nTF {##1}
1920             {
1921                 \bool_set_false:c { l__xeCJK_copy_#1_bool }
1922                 \bool_set_false:c { l__xeCJK_add_#1_bool }
1923                 \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
1924             }
1925             {
1926                 \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
1927                 \str_if_eq:nnTF {##1} { * }
1928                 {
1929                     \bool_set_true:c { l__xeCJK_copy_#1_bool }
1930                     \bool_set_false:c { l__xeCJK_add_#1_bool }

```

```

1931         }
1932     {
1933         \bool_set_false:c { l__xeCJK_copy_#1_bool }
1934         \bool_set_true:c { l__xeCJK_add_#1_bool }
1935         \__xeCJK_get_sub_features:nn {#1} {##1}
1936     }
1937 }
1938 } ,
1939 #1 .default:n = { }
1940 }
1941 }
(End definition for \xeCJK_new_sub_key:n and \g__xeCJK_sub_key_seq.)

```

```

\__xeCJK_get_sub_features:nn
\__xeCJK_get_sub_features:wn
1942 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
1943 {
1944     \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
1945     \clist_clear:N \l__xeCJK_sub_font_options_clist
1946     \tl_if_head_eq_meaning:VNTF \l__xeCJK_tmpa_tl [ % ]
1947     {
1948         \exp_after:wN \__xeCJK_get_sub_features:wn \l__xeCJK_tmpa_tl \c_empty_tl
1949         \tl_if_blank:VT \l__xeCJK_sub_font_name_tl
1950         {
1951             \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl
1952             \clist_clear:N \l__xeCJK_sub_font_options_clist
1953         }
1954     }
1955     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl }
1956     \tl_if_blank:VTF \l__xeCJK_sub_font_name_tl
1957     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
1958     { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
1959     \clist_set_eq:cN { l__xeCJK_#1_font_options_clist } \l__xeCJK_sub_font_options_clist
1960     \tl_set_eq:cN { l__xeCJK_#1_font_name_tl } \l__xeCJK_sub_font_name_tl
1961 }
1962 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:wn [#1] #2
1963 {
1964     \clist_set:Nn \l__xeCJK_sub_font_options_clist {#1}
1965     \str_if_eq:nnTF {#2} { \c_empty_tl }
1966     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \c_empty_tl }
1967     { \tl_set:Nn \l__xeCJK_sub_font_name_tl {#2} }
1968 }
1969 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
1970 \cs_generate_variant:Nn \tl_if_head_eq_meaning:nNTF { V }
1971 \cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
(End definition for \__xeCJK_get_sub_features:nn and \__xeCJK_get_sub_features:wn.)

```

FallBack

```

1972 \xeCJK_new_sub_key:n { FallBack }
(End definition for FallBack. This function is documented on page 7.)

```

```

\g__xeCJK_features_id_prop 调用字体的属性声明,同 fontspec 宏包。
    BoldFont
    ItalicFont
    BoldItalicFont
    SlantedFont
    BoldSlantedFont
    BoldFeatures
    ItalicFeatures
    BoldItalicFeatures
    SlantedFeatures
    BoldSlantedFeatures
1973 \prop_new:N \g__xeCJK_features_id_prop
1974 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
1975 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
1976 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
1977 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
1978 \prop_put:Nnn \g__xeCJK_features_id_prop { bfs1 } { BoldSlanted }
1979 \prop_map_inline:Nn \g__xeCJK_features_id_prop
1980 {
1981     \keys_define:nn { xeCJK / features }
1982     {
1983         #2Font .tl_set:c = { l__xeCJK_font_name_#1_tl } ,

```

```

1984         #2Features .clist_set:c = { l__xeCJK_font_feat_#1_clist }
1985     }
1986 }
(End definition for \g__xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

```

1987 \keys_define:nn { xeCJK / features }
1988 {
1989     AutoFakeBold .choice: ,
1990     AutoFakeBold / false .code:n =
1991         { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
1992     AutoFakeBold / unknown .code:n =
1993         {
1994             \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
1995             \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1996         } ,
1997     AutoFakeBold .default:n = { \g__xeCJK_embolden_factor_fp } ,
1998     AutoFakeSlant .choice: ,
1999     AutoFakeSlant / false .code:n =
2000         { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
2001     AutoFakeSlant / unknown .code:n =
2002         {
2003             \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
2004             \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
2005         } ,
2006     AutoFakeSlant .default:n = { \g__xeCJK_slant_factor_fp }
2007 }
(End definition for AutoFakeBold and AutoFakeSlant.)

```

__xeCJK_set_family_initial:

```

2008 \cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
2009 {
2010     \int_gincr:N \g__xeCJK_family_int
2011     \prop_clear:N \l__xeCJK_add_font_prop
2012     \prop_map_inline:Nn \g__xeCJK_features_id_prop
2013     {
2014         \tl_clear:c { l__xeCJK_font_name_##1_tl }
2015         \clist_clear:c { l__xeCJK_font_feat_##1_clist }
2016     }
2017     \clist_clear:N \l__xeCJK_fontspec_options_clist
2018     \seq_clear:N \l__xeCJK_sub_key_seq
2019     \seq_map_inline:Nn \g__xeCJK_sub_key_seq
2020     {
2021         \bool_set_false:c { l__xeCJK_copy_##1_bool }
2022         \bool_set_false:c { l__xeCJK_add_##1_bool }
2023     }
2024     \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
2025     \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
2026     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
2027     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
2028 }
2029 \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。

```

2030 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
2031 {
2032     \group_begin:
2033     \__xeCJK_set_family_initial:
2034     \tl_set:Nn \l__xeCJK_family_name_tl {#1}
2035     \clist_set:Nn \l__xeCJK_font_options_clist {#2}
2036     \tl_set:Nn \l__xeCJK_font_name_tl {#3}

```

```

2037 \clist_concat:NNN \l__xeCJK_font_features_clist
2038 \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
2039 \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_features_clist
2040 \keys_set_known:nVN { xeCJK / features }
2041 \l__xeCJK_font_features_clist \l__xeCJK_fontspec_options_clist
2042 \__xeCJK_parse_features:
2043 \__xeCJK_pass_features:
2044 \__xeCJK_check_family:V \l__xeCJK_family_name_tl
2045 \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
2046 \__xeCJK_save_family_info:
2047 \__xeCJK_set_sub_block_family:
2048 \group_end:
2049 }
2050 \cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , xxx , Vvv }
(End definition for \xeCJK_set_family:nnn.)

```

__xeCJK_remove_duplicate_keys:N

```

2051 \cs_new_protected_nopar:Npn \__xeCJK_remove_duplicate_keys:N #1
2052 {
2053   \prop_clear:N \l__xeCJK_tmp_prop
2054   \keyval_parse:NNV \__xeCJK_prop_put_aux:n \__xeCJK_prop_put_aux:nn #1
2055   \clist_clear:N #1
2056   \prop_map_inline:Nn \l__xeCJK_tmp_prop
2057   {
2058     \tl_if_blank:nTF {##2}
2059     { \clist_put_right:Nn #1 {##1} }
2060     { \clist_put_right:Nn #1 { ##1 = {##2} } }
2061   }
2062 }
2063 \cs_generate_variant:Nn \keyval_parse:NNn { NNV }
2064 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
2065 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} { } }
2066 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
2067 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} {#2} }
(End definition for \__xeCJK_remove_duplicate_keys:N.)

```

__xeCJK_gset_family_cs:x

```

2068 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_cs:x #1
2069 {
2070   \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n {#1} }
2071   {
2072     \group_begin:
2073     \exp_not:n { \cs_set_eq:NN \__xeCJK_update_family:n \use_none:n }
2074     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
2075     { \exp_not:V \l__xeCJK_fontspec_options_clist }
2076     { \exp_not:V \l__xeCJK_font_name_tl }
2077     \__xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
2078     \group_end:
2079   }
2080 }
(End definition for \__xeCJK_gset_family_cs:x.)

```

__xeCJK_check_family:n

```

2081 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
2082 {
2083   \prop_get:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmpa_tl
2084   {
2085     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl
2086     { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
2087     \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmpa_tl }
2088   }
2089 }

```



```

2090 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
2091 \__xeCJK_msg_new:nn { CJKfamily-redef }
2092 { 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
2093 \cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
2094 { \prop_put:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2095 \cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
2096 { \prop_put_if_new:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2097 \cs_generate_variant:Nn \__xeCJK_add_font:nn { nx , nV , nv }
2098 \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
2099 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
2100 {
2101   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2102   { FakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
2103 }
2104 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
2105 {
2106   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2107   { FakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
2108 }
(End definition for \__xeCJK_add_fake_bold:n and \__xeCJK_add_fake_slant:n.)

```

```

\__xeCJK_parse_features:
\__xeCJK_parse_features:nn
2109 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:
2110 { \prop_map_function:NN \g__xeCJK_features_id_prop \__xeCJK_parse_features:nn }
2111 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:nn #1#2
2112 {
2113   \__xeCJK_if_font_select:nTF {#1}
2114   {
2115     \__xeCJK_add_font:nv {#2} { l__xeCJK_font_name_#1_tl }
2116     \__xeCJK_if_it_or_sl:nTF {#1}
2117     {
2118       \__xeCJK_if_font_select:nF { bf #1 }
2119       { \__xeCJK_add_font_if_new:nv {#2} { l__xeCJK_font_name_#1_tl } }
2120     }
2121     {
2122       \str_if_eq:nnT {#1} { bf }
2123       {
2124         \__xeCJK_if_font_select:nF { bfit }
2125         { \__xeCJK_add_font:nv { BoldItalic } \l__xeCJK_font_name_bf_tl }
2126         \__xeCJK_if_font_select:nF { bfsf }
2127         { \__xeCJK_add_font:nv { BoldSlanted } \l__xeCJK_font_name_bf_tl }
2128       }
2129     }
2130   }
2131   { \__xeCJK_set_fake_font:nn {#1} {#2} }
2132 }
(End definition for \__xeCJK_parse_features: and \__xeCJK_parse_features:nn.)

```

```

\__xeCJK_if_font_select_p:n
\__xeCJK_if_it_or_sl_p:n
\__xeCJK_if_font_select:nTF
\__xeCJK_if_it_or_sl:nTF
2133 \prg_new_conditional:Npnn \__xeCJK_if_font_select:n #1 { p , T , F , TF }
2134 {
2135   \tl_if_blank:vTF { l__xeCJK_font_name_#1_tl }
2136   { \prg_return_false: } { \prg_return_true: }
2137 }
2138 \prg_new_conditional:Npnn \__xeCJK_if_it_or_sl:n #1 { p , T , F , TF }
2139 {

```

```

2140 \if_predicate:w
2141 \bool_if_p:n { \str_if_eq_p:nn { it } {#1} || \str_if_eq_p:nn { sl } {#1} }
2142 \prg_return_true: \else: \prg_return_false: \fi:
2143 }
2144 \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

```

2145 \cs_new_protected_nopar:Npn \__xeCJK_set_fake_font:nn #1#2
2146 {
2147   \str_if_eq:nnTF {#1} { bf }
2148   { \bool_if:NT \l__xeCJK_auto_fake_bold_bool { \__xeCJK_add_fake_bold:n {#1} } }
2149   {
2150     \bool_if:NTF \l__xeCJK_auto_fake_slant_bool
2151     {
2152       \bool_if:nT
2153       { \__xeCJK_if_it_or_sl_p:n {#1}
2154         || ( \str_if_eq_p:nn {#1} { bfit }
2155           && ! ( \__xeCJK_if_font_select_p:n { it } ) )
2156         || ( \str_if_eq_p:nn {#1} { bfsl }
2157           && ! ( \__xeCJK_if_font_select_p:n { sl } ) )
2158       }
2159       { \__xeCJK_add_fake_slant:n {#1} }
2160     }
2161     { \__xeCJK_if_it_or_sl:nT {#1} { \__xeCJK_map_it_sl:nn {#1} {#2} } }
2162     \bool_if:nT
2163     { \l__xeCJK_auto_fake_bold_bool
2164       && ! ( \__xeCJK_if_it_or_sl_p:n {#1} )
2165       && ! ( \__xeCJK_if_font_select_p:n { bf } )
2166     }
2167     { \__xeCJK_add_fake_bold:n {#1} }
2168   }
2169   \__xeCJK_add_font_if_new:nn {#2} { * }
2170 }
(End definition for \__xeCJK_set_fake_font:nn.)

```

__xeCJK_map_it_sl:n

```

2171 \cs_new_protected_nopar:Npn \__xeCJK_map_it_sl:n #1#2
2172 {
2173   \__xeCJK_if_map_font_select:nT {#1}
2174   {
2175     \__xeCJK_add_font:nx {#2} { \__xeCJK_get_map_font:n {#1} }
2176     \__xeCJK_if_font_select:nF { bf #1 }
2177     { \__xeCJK_add_font_if_new:nx { Bold #2 } { \__xeCJK_get_map_font:n {#1} } }
2178   }
2179 }
(End definition for \__xeCJK_map_it_sl:n.)

```

__xeCJK_get_map_font:n

```

2180 \cs_new_nopar:Npn \__xeCJK_get_map_font:n #1
2181 { \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

```

2182 \prg_new_conditional:Npnn \__xeCJK_if_map_font_select:n #1 { p , T , F , TF }
2183 {
2184   \xeCJK_if_blank_x:nTF { \__xeCJK_get_map_font:n {#1} }
2185   { \prg_return_false: } { \prg_return_true: }
2186 }
(End definition for \__xeCJK_if_map_font_select:n.)

```

__xeCJK_pass_features:

```
2187 \cs_new_protected_nopar:Npn \__xeCJK_pass_features:
2188 {
2189   \prop_map_inline:Nn \g__xeCJK_features_id_prop
2190   {
2191     \clist_if_empty:cF { l__xeCJK_font_feat_##1_clist }
2192     {
2193       \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2194       { ##2Features = { \exp_not:v { l__xeCJK_font_feat_##1_clist } } }
2195     }
2196   }
2197   \prop_map_inline:Nn \l__xeCJK_add_font_prop
2198   { \clist_put_right:Nn \l__xeCJK_fontspec_options_clist { ##1 = {##2} } }
2199 }
(End definition for \__xeCJK_pass_features:.)
```

\g__xeCJK_family_name_prop

```
\g__xeCJK_family_font_name_prop 2200 \prop_new:N \g__xeCJK_family_name_prop
\g__xeCJK_family_font_options_prop 2201 \prop_new:N \g__xeCJK_family_font_name_prop
2202 \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:

```
2203 \cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
2204 {
2205   \group_begin:
2206   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
2207   { \clist_remove_all:Nn \l__xeCJK_font_options_clist {##1} }
2208   \prop_gput:NVV \g__xeCJK_family_font_name_prop
2209   \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
2210   \prop_gput:NVV \g__xeCJK_family_font_options_prop
2211   \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
2212   \group_end:
2213 }
(End definition for \__xeCJK_save_family_info:.)
```

__xeCJK_set_sub_block_family:

```
2214 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_block_family:
2215 {
2216   \seq_map_inline:Nn \l__xeCJK_sub_key_seq
2217   {
2218     \tl_set:Nx \l__xeCJK_sub_family_tl { \l__xeCJK_family_name_tl/##1 }
2219     \bool_if:cT { l__xeCJK_copy_##1_bool } { \__xeCJK_copy_sub_family:n {##1} }
2220     \bool_if:cT { l__xeCJK_add_##1_bool }
2221     {
2222       \xeCJK_set_family:Vvv \l__xeCJK_sub_family_tl
2223       { l__xeCJK_##1_font_options_clist } { l__xeCJK_##1_font_name_tl }
2224     }
2225   }
2226 }
2227 \cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
2228 {
2229   \__xeCJK_check_family:V \l__xeCJK_sub_family_tl
2230   \prop_get:NVNT \g__xeCJK_family_font_name_prop
2231   \l__xeCJK_family_name_tl \l__xeCJK_tmpa_tl
2232   {
2233     \prop_gput:NVV \g__xeCJK_family_font_name_prop
2234     \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl
2235   }
2236   \prop_get:NVNT \g__xeCJK_family_font_options_prop
2237   \l__xeCJK_family_name_tl \l__xeCJK_tmpa_clist
2238   {
```

```

2239         \clist_remove_all:Nn \l__xeCJK_tmpa_clist { #1 = * }
2240         \prop_gput:NVV \g__xeCJK_family_font_options_prop
2241         \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
2242     }
2243     \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_tl } }
2244     {
2245         \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }
2246         {
2247             \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
2248             { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_tmpa_tl
2249             {
2250                 \__xeCJK_gset_family_nfss_cs:xx
2251                 { \l__xeCJK_sub_family_tl } { \exp_not:N \l__xeCJK_tmpa_tl }
2252             }
2253         }
2254     }
2255 }
(End definition for \__xeCJK_set_sub_block_family:.)

```

__xeCJK_copy_family:nn

```

2256 \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
2257 {
2258     \xeCJK_family_if_exist:xT {#2}
2259     {
2260         \tl_map_inline:nn
2261         {
2262             \g__xeCJK_family_name_prop
2263             \g__xeCJK_family_font_name_prop
2264             \g__xeCJK_family_font_options_prop
2265         }
2266         {
2267             \prop_get:NnNT ##1 {#2} \l__xeCJK_tmpa_tl
2268             { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_tl }
2269         }
2270         \cs_gset_eq:cc
2271         { \__xeCJK_family_nfss_csname:n {#1} }
2272         { \__xeCJK_family_nfss_csname:n {#2} }
2273     }
2274 }
2275 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }
(End definition for \__xeCJK_copy_family:nn.)

```

5.13 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:
\xeCJK_select_font:x
2276 \tl_new:N \l_xeCJK_current_font_tl
2277 \tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
2278 \cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
2279 \cs_new_protected_nopar:Npn \xeCJK_select_font:
2280 {
2281     \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
2282     {
2283         \tl_set:Nx \l__xeCJK_current_coord_tl { \l_xeCJK_current_font_tl }
2284         \__xeCJK_family_use:x { \l_xeCJK_family_tl }
2285         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coord_tl }
2286     }
2287 }
2288 \cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
2289 {
2290     \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
2291     {
2292         \tl_set:Nx \l__xeCJK_current_coord_tl { \__xeCJK_font_csname:n {#1} }

```

```

2293         \__xeCJK_family_use:x {#1}
2294         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2295     }
2296 }
2297 \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。

```

2298 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
2299 {
2300     \str_if_eq:nnF {#1} {#2}
2301     {
2302         \__xeCJK_info:nxx { CJK-block } {#1} {#2}
2303         \str_if_eq:nnTF {#2} { CJK }
2304         { \xeCJK_select_font: }
2305         { \__xeCJK_block_select_font:n {#2} }
2306     }
2307 }
2308 \__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。

```

2309 \cs_new_protected_nopar:Npn \__xeCJK_block_select_font:n #1
2310 {
2311     \cs_if_exist_use:cF { \__xeCJK_font_csname:n { \l__xeCJK_family_tl/#1 } }
2312     {
2313         \tl_set:Nx \l__xeCJK_current_coor_tl
2314         { \__xeCJK_font_csname:n { \l__xeCJK_family_tl/#1 } }
2315         \xeCJK_family_if_exist:xF { \l__xeCJK_family_tl/#1 }
2316         {
2317             \__xeCJK_copy_family:xx { \l__xeCJK_family_tl/#1 }
2318             {
2319                 \cs_if_exist:cTF
2320                 { \__xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
2321                 { \CJKfamilydefault/#1 } { \l__xeCJK_family_tl }
2322             }
2323         }
2324         \__xeCJK_family_use:x { \l__xeCJK_family_tl/#1 }
2325         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2326     }
2327 }
(End definition for \__xeCJK_block_select_font:n.)

```

__xeCJK_family_csname:n

__xeCJK_family_nfss_csname:n

__xeCJK_family_use:x

__xeCJK_gset_family_nfss_cs:xx

```

2328 \cs_new_nopar:Npn \__xeCJK_family_csname:n #1 { xeCJK/family/#1 }
2329 \cs_new_nopar:Npn \__xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
2330 \cs_new_nopar:Npn \__xeCJK_family_use:x #1 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
2331 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_nfss_cs:xx #1#2
2332 {
2333     \prop_gput:Nxx \g__xeCJK_family_name_prop {#1} {#2}
2334     \cs_gset_protected_nopar:cpx { \__xeCJK_family_nfss_csname:n {#1} }
2335     {
2336         \tl_set:Nx \exp_not:N \f@encoding { \c__xeCJK_encoding_tl }
2337         \tl_set:Nx \exp_not:N \f@family {#2}
2338         \exp_not:N \selectfont
2339     }
2340 }
2341 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
(End definition for \__xeCJK_family_csname:n and others.)

```

`\xeCJK_family_if_exist:xTF`

```
2342 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
2343 {
2344   \cs_if_exist:cTF { \__xeCJK_family_nfss_csname:n {#1} }
2345   { \use_i:nn }
2346   { \cs_if_exist_use:cTF { \__xeCJK_family_csname:n {#1} } }
2347   { \prg_return_true: } { \prg_return_false: }
2348 }
(End definition for \xeCJK_family_if_exist:xTF.)
```

\CJKfamily 用于切换 CJK 字体族。

```
2349 \NewDocumentCommand \CJKfamily { t+ t- m }
2350 {
2351   \xeCJK_if_blank_x:nTF {#3}
2352   {
2353     \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
2354     \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
2355   }
2356   {
2357     \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
2358     {
2359       \xeCJK_family_if_exist:xTF {#3}
2360       {
2361         \tl_set:Nx \l_xeCJK_family_tl {#3}
2362         \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2363         \IfBooleanT {#1} { \__xeCJK_family_use:x {#3} }
2364       }
2365       { \__xeCJK_family_unknown_warning:x {#3} }
2366     }
2367   }
2368   \tex_ignorespaces:D
2369 }
(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。

```
2370 \tl_new:N \l_xeCJK_family_tl
(End definition for \l_xeCJK_family_tl.)
```

`__xeCJK_gobble_CJKfamily:`

```
2371 \cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
2372 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
2373 \DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }
(End definition for \__xeCJK_gobble_CJKfamily:.)
```

`\xeCJK_family_if_exist_use:x`

```
2374 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
2375 {
2376   \xeCJK_family_if_exist:xTF {#1}
2377   { \__xeCJK_family_use:x {#1} }
2378   { \__xeCJK_family_unknown_warning:x {#1} }
2379 }
(End definition for \xeCJK_family_if_exist_use:x.)
```

`_xeCJK_family_unknown_warning:n`

```
2380 \cs_new_protected_nopar:Npn \_xeCJK_family_unknown_warning:n #1
2381 {
2382   \prop_if_empty:NF \g__xeCJK_family_font_name_prop
2383   {
2384     \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
2385     {
2386       \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
2387       \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}

```

```

2388     }
2389   }
2390 }
2391 \cs_generate_variant:Nn \__xeCJK_family_unknown_warning:n { x }
2392 \seq_new:N \g__xeCJK_unknown_family_seq
2393 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
2394 {
2395   Unknown~CJK~family~'\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
2396   Try~to~use~'\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2397 }
2398 \cs_new_nopar:Npn \__xeCJK_msg_def_family_map:n #1
2399 {
2400   \str_case:x:nnn {#1}
2401   {
2402     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
2403     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
2404     \CJKttdefault { \token_to_str:N \setCJKmonofont }
2405   }
2406   { \token_to_str:N \setCJKfamilyfont {#1} }
2407   [...]{}
2408 }
2409 \cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1
2410 {
2411   \str_case:x:nnn {#1}
2412   {
2413     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
2414     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
2415     \CJKttdefault { \token_to_str:N \CJKttdefault }
2416   }
2417   {#1}
2418 }
(End definition for \__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 2419 \NewDocumentCommand \setCJKmainfont { O{} m }
\setCJKmonofont 2420 {
2421   \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2}
2422   \normalfont
2423 }
2424 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
2425 \NewDocumentCommand \setCJKsansfont { O{} m }
2426 {
2427   \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2}
2428   \normalfont
2429 }
2430 \NewDocumentCommand \setCJKmonofont { O{} m }
2431 {
2432   \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2}
2433   \normalfont
2434 }

```

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

```

2435 \@onlypreamble \setCJKmainfont
2436 \@onlypreamble \setCJKmathfont
2437 \@onlypreamble \setCJKsansfont
2438 \@onlypreamble \setCJKmonofont
2439 \@onlypreamble \setCJKromanfont

```

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

```

\newCJKfontfamily 2440 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
\CJKfontspec 2441 { \xeCJK_set_family:xxx {#1} {#2} {#3} }
2442 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
2443 {

```

```

2444 \tl_set:Nx \l__xeCJK_tmpa_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
2445 \cs_new_protected_nopar:Npx #2 { \CJKfamily { \l__xeCJK_tmpa_tl } }
2446 \xeCJK_set_family:xxx { \l__xeCJK_tmpa_tl } {#3} {#4}
2447 }
2448 \NewDocumentCommand \CJKfontspec { 0{ } m }
2449 {
2450 \xeCJK_fontspec:xx {#1} {#2}
2451 \tex_ignorespaces:D
2452 }

```

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

\xeCJK_fontspec:nn

```

2453 \cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
2454 {
2455 \tl_set:Nx \l__xeCJK_tmpa_tl { \tl_to_str:n { CJKfontspec/#1/#2/id } }
2456 \tl_if_exist:cF { \l__xeCJK_tmpa_tl }
2457 {
2458 \tl_set:Nx \l__xeCJK_tmppb_tl
2459 { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
2460 \tl_gset_eq:cN { \l__xeCJK_tmpa_tl } \l__xeCJK_tmppb_tl
2461 \xeCJK_set_family:Vnn \l__xeCJK_tmppb_tl {#1} {#2}
2462 }
2463 \exp_args:Nc \CJKfamily { \l__xeCJK_tmpa_tl }
2464 }
2465 \cs_generate_variant:Nn \xeCJK_fontspec:nn { VV , xx }

```

(End definition for \xeCJK_fontspec:nn.)

\defaultCJKfontfeatures

分别用于设置 CJK 字体的默认属性和增加当前 CJK 字体的属性。

\addCJKfontfeatures

```

2466 \clist_new:N \g__xeCJK_default_features_clist
2467 \NewDocumentCommand \defaultCJKfontfeatures { m }
2468 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
2469 \@onlypreamble \defaultCJKfontfeatures
2470 \NewDocumentCommand \addCJKfontfeatures { m }
2471 {
2472 \xeCJK_add_font_features:x {#1}
2473 \tex_ignorespaces:D
2474 }
2475 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

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

\xeCJK_add_font_features:n

```

2476 \cs_new_protected_nopar:Npn \xeCJK_add_font_features:n #1
2477 {
2478 \prop_get:NVNTF \g__xeCJK_family_font_name_prop \l_xeCJK_family_tl \l__xeCJK_tmpa_tl
2479 {
2480 \clist_set:Nn \l__xeCJK_tmpa_clist {#1}
2481 \seq_map_function:NN
2482 \g__xeCJK_CJK_sub_class_seq \__xeCJK_parse_sub_class_features:n
2483 \prop_get:NVNT \g__xeCJK_family_font_options_prop
2484 \l_xeCJK_family_tl \l__xeCJK_tmppb_clist
2485 {
2486 \clist_concat:NNN
2487 \l__xeCJK_tmpa_clist \l__xeCJK_tmppb_clist \l__xeCJK_tmpa_clist
2488 }
2489 \xeCJK_fontspec:VV \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_tl
2490 }
2491 { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
2492 }
2493 \cs_generate_variant:Nn \xeCJK_add_font_features:n { x }
2494 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
2495 {
2496 \token_to_str:N \addCJKfontfeature (s)~ignored.\!\!\!

```



```

2497     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
2498   }
(End definition for \xeCJK_add_font_features:n.)

```

_xeCJK_parse_sub_class_features:n

```

2499 \cs_new_protected_nopar:Npn \__xeCJK_parse_sub_class_features:n #1
2500 {
2501   \clist_if_in:NnT \l__xeCJK_tmpa_clist {#1}
2502   {
2503     \clist_remove_all:Nn \l__xeCJK_tmpa_clist {#1}
2504     \prop_get:NnNF \g__xeCJK_family_font_name_prop
2505       { \l__xeCJK_family_tl/#1 } \l__xeCJK_tmpb_tl
2506     {
2507       \prop_get:NnNF \g__xeCJK_family_font_name_prop
2508         { \CJKfamilydefault/#1 } \l__xeCJK_tmpb_tl
2509       { \tl_set_eq:NN \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
2510     }
2511     \clist_set:Nx \l__xeCJK_tmpa_clist
2512       { #1 = { [ \exp_not:V \l__xeCJK_tmpa_clist ] { \exp_not:V \l__xeCJK_tmpb_tl } } }
2513     \clist_map_break:
2514   }
2515 }
2516 \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。如果 \CJKfamilydefault 没有被更改,则在此时根据西文字体的情况更新 \CJKfamilydefault。如果 \CJKfamilydefault 对应的字体族没有定义,则使用 \CJKrmdefault 作为默认字体族。若 \CJKrmdefault 也没有定义,则使用在导言区设置的第一个 CJK 字体作为默认字体族。最后设置数学字体。

```

2517 \__xeCJK_at_end_preamble:n
2518 {
2519   \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
2520   \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
2521   {
2522     \tl_gset:Nx \CJKfamilydefault
2523     {
2524       \str_case:x:nnn { \familydefault }
2525       {
2526         { \rmdefault } { \exp_not:N \CJKrmdefault }
2527         { \sfdefault } { \exp_not:N \CJKsfdefault }
2528         { \ttdefault } { \exp_not:N \CJKttdefault }
2529       }
2530       { \CJKfamilydefault }
2531     }
2532   }
2533   \cs_undefine:N \__xeCJK_family_default_wrap:n
2534   \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
2535     { \__xeCJK_warning:nx { no-CJKfamily } { \CJKfamilydefault } }
2536     {
2537       \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2538       {
2539         \tl_set_eq:NN \l__xeCJK_tmpa_tl \CJKfamilydefault
2540         \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
2541         { \use:n }
2542         {
2543           \xeCJK_family_if_exist:xTF { \CJKrmdefault }
2544           { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
2545         }
2546         {
2547           \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
2548             { \prop_map_break:n { \tl_gset:Nn \CJKfamilydefault {#1} } }
2549         }
2550       }
2551       \__xeCJK_warning:nxx { CJKfamilydefault-undefined }

```

```

2551         { \l__xeCJK_tmpa_tl } { \CJKfamilydefault }
2552     }
2553     \CJKfamily { \CJKfamilydefault }
2554     \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
2555 }
2556 }
2557 \__xeCJK_msg_new:nn { no-CJKfamily }
2558 {
2559     It~seems~that~you~have~not~declare~a~CJKfamily.\\
2560     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
2561     '\__xeCJK_msg_def_family_map:n {#1}'\\
2562     in~the~preamble~to~declare~the~default~CJKfamily.\\
2563 }
2564 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
2565 {
2566     Undefined~CJK~default~family~'\__xeCJK_msg_family_map:n {#1}'~
2567     has~been~replaced~by~'\__xeCJK_msg_family_map:n {#2}'\\.\\
2568     Try~to~use~'\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2569 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```

2570 \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。

```

2571 \NewDocumentCommand \setCJKmathfont { O{} m }
2572 { \xeCJK_set_family:xxx { \c__xeCJK_math_tl } {#1} {#2} }
2573 \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。

```

2574 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
2575 {
2576     \xeCJK_family_if_exist:xF { \c__xeCJK_math_tl }
2577     { \tl_set:Nx \l__xeCJK_tmpa_tl { \c__xeCJK_math_tl } }
2578     {
2579         \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2580         { \tl_set:Nx \l__xeCJK_tmpa_tl { \CJKfamilydefault } }
2581         { \use_none:nnnn }
2582     }
2583     \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_tmpp_tl
2584     {
2585         \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmpp_tl }
2586         \DeclareSymbolFont \c__xeCJK_math_tl
2587         \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \mddefault \shapedefault
2588         \SetSymbolFont \c__xeCJK_math_tl { bold }
2589         \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \bfdefault \shapedefault
2590         \int_const:Nn \c__xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }
2591         \clist_concat:NNN
2592         \l__xeCJK_tmpa_clist \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
2593         \clist_concat:NNN
2594         \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_clist \c__xeCJK_FullRight_chars_clist
2595         \clist_map_inline:Nn \l__xeCJK_tmpa_clist
2596         {
2597             \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
2598             { \c_seven } { \c__xeCJK_math_fam_int }
2599         }
2600     }
2601 }

```

(End definition for \xeCJK_set_mathfont:.)

\xeCJK_gset_mathcode:nnnn

```
2602 \cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
2603 {
2604   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
2605   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
2606   {
2607     \tex_global:D \xeCJK_xetex_mathcode:w
2608     \l__xeCJK_tmpa_int = #3 ~ #4 \l__xeCJK_tmpa_int
2609     \int_incr:N \l__xeCJK_tmpa_int
2610   }
2611 }
(End definition for \xeCJK_gset_mathcode:nnnn.)
```

5.15 抄录环境中的间距调整

Verb 如果设置为 `env`, 则只在 \LaTeX 的抄录环境里使用 `\xeCJKVerbAddon`, 而不包括 `\verb`。对当前使用环境的判断基于在标准 \LaTeX 的环境定义里使用 `\begingroup` 和 `\endgroup` 来分组。

```
2612 \keys_define:nn { xecjk / options }
2613 {
2614   Verb .choice: ,
2615   Verb / true .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_zero } ,
2616   Verb / env .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_one } ,
2617   Verb / env+ .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_two } ,
2618   Verb / false .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_three }
2619 }
2620 \int_new:N \l__xeCJK_verb_case_int
2621 \__xeCJK_after_preamble:n
2622 {
2623   \tl_put_right:Nn \verbatim@font
2624   {
2625     \if_case:w \l__xeCJK_verb_case_int
2626     \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2627     \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2628     \or:
2629     \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
2630     {
2631       \__xeCJK_set_verb_exspace:
2632       \__xeCJK_verb_addon:
2633     }
2634     {
2635       \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2636       \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2637     }
2638     \or:
2639     \int_compare:nNnT \etex_currentgrouptype:D = \c_fourteen
2640     {
2641       \__xeCJK_set_verb_exspace:
2642       \__xeCJK_verb_addon:
2643     }
2644     \fi:
2645   }
2646 }
```

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

\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。

```
2647 \NewDocumentCommand \xeCJKVerbAddon { }
2648 {
2649   \int_compare:nNnF \etex_currentgrouplevel:D = \c_zero
2650   {
```

```

2651         \__xeCJK_set_verb_exspace:
2652         \__xeCJK_verb_addon:
2653     }
2654 }
2655 \cs_new_protected_nopar:Npn \__xeCJK_verb_addon:
2656 {
2657     \bool_if:NF \l__xeCJK_verb_addon_bool
2658     {
2659         \bool_set_true:N \l__xeCJK_verb_addon_bool
2660         \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
2661         \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
2662         \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
2663         \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
2664         \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
2665         \xeCJKsetup { xCJKecglue = false }
2666         \cs_set_protected_nopar:Npn \CJKglue
2667         { \skip_horizontal:N \g__xeCJK_verb_exspace_skip }
2668         \cs_set_protected_nopar:Npn \CJKecglue
2669         { \skip_horizontal:n { 0.5 \g__xeCJK_verb_exspace_skip } }
2670         \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
2671         \cs_set_eq:NN \xeCJK_ignore_spaces:w \CJKecglue
2672         \skip_set:Nn \tex_rightskip:D { \c_zero_dim plus 1 fil }
2673     }
2674 }
2675 \bool_new:N \l__xeCJK_verb_addon_bool
2676 \cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon

```

(End definition for \xeCJKVerbAddon. This function is documented on page 11.)

`__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。

```

2677 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
2678 {
2679     \tl_if_exist:cF { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
2680     {
2681         \group_begin:
2682         \xeCJK_select_font:
2683         \use:x
2684         {
2685             \group_end:
2686             \tl_gset:cx
2687             \exp_not:n { { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2688             {
2689                 \exp_not:N \dim_max:nn { \c_zero_dim }
2690                 {
2691                     \c_two \tex_fontdimen:D \c_two \tex_font:D -
2692                     \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00
2693                 }
2694             }
2695         }
2696     }
2697     \skip_gset:Nn \g__xeCJK_verb_exspace_skip
2698     { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2699 }
2700 \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。

```

2701 \cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
2702 { {
2703     \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
2704     { \xeCJK_set_visible_space_font: }
2705     \textvisiblespace
2706 } }
2707 \AtEndOfPackage
2708 {
2709     \tl_put_left:Nn \fontspec_visible_space: { \scan_stop: }
2710     \cs_gset_eq:NN \fontspec_visible_space_fallback: \xeCJK_visible_space_fallback:
2711 }
(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。

```

2712 \cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
2713 {
2714     \tl_set:Nx \l__xeCJK_current_coor_tl { xeCJK/space/\curr@fontshape/\f@size }
2715     \exp_after:wN \__xeCJK_set_visible_space_size:n
2716     \exp_after:wN { \dim_use:N \tex_fontdimen:D \c_two \tex_font:D }
2717     \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2718 }
2719 \cs_new_protected_nopar:Npn \__xeCJK_set_visible_space_size:n #1
2720 {
2721     \tl_set:Nx \f@encoding { \g_fontspec_encoding_tl }
2722     \tl_set:Nx \f@family { lmtt }
2723     \selectfont
2724     \dim_compare:nNf {#1} = { \tex_fontdimen:D \c_two \tex_font:D }
2725     {
2726         \fontsize
2727         {
2728             \dim_eval:n
2729             {
2730                 \f@size pt *
2731                 \dim_ratio:nn {#1} { \tex_fontdimen:D \c_two \tex_font:D }
2732             }
2733         }
2734         { \f@baselineskip }
2735         \selectfont
2736     }
2737 }
(End definition for \xeCJK_set_visible_space_font:.)

```

5.16 xeCJK 其它选项

LocalConfig 声明载入本地配置文件的选项。

```

2738 \keys_define:nn { xeCJK / options }
2739 {
2740     LocalConfig .choice: ,
2741     LocalConfig / false .code:n =
2742     { \bool_gset_false:N \g__xeCJK_config_bool } ,
2743     LocalConfig / true .code:n =
2744     {
2745         \bool_gset_true:N \g__xeCJK_config_bool
2746         \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
2747     } ,
2748     LocalConfig / unknown .code:n =
2749     {
2750         \bool_gset_true:N \g__xeCJK_config_bool
2751         \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }

```

```

2752     } ,
2753     LocalConfig .default:n = { true }
2754 }
2755 \tl_new:N \g__xeCJK_config_name_tl
2756 \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 2757 \keys_define:nn { xeCJK / options }
2758 {
2759     CJKnumber .bool_gset:N = \g__xeCJK_number_bool ,
2760     indentfirst .bool_gset:N = \g__xeCJK_indent_bool ,
2761     normalindentfirst .meta:n = { indentfirst = false }
2762 }
(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 选项进行修改,使其适用于 xeCJK。

```

silent 2763 \keys_define:nn { xeCJK / options }
2764 {
2765     quiet .code:n =
2766     {
2767         \msg_redirect_module:nnn { xeCJK } { warning } { info }
2768         \msg_redirect_module:nnn { xeCJK } { info } { none }
2769         \xeCJK_if_package_loaded:nF { fontspec }
2770         { \PassOptionsToPackage { quiet } { fontspec } }
2771     } ,
2772     silent .code:n =
2773     {
2774         \msg_redirect_module:nnn { xeCJK } { warning } { none }
2775         \msg_redirect_module:nnn { xeCJK } { info } { none }
2776         \xeCJK_if_package_loaded:nF { fontspec }
2777         { \PassOptionsToPackage { silent } { fontspec } }
2778     } ,
2779     unknown .code:n =
2780     {
2781         \xeCJK_if_package_loaded:nTF { fontspec }
2782         { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2783         { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2784     }
2785 }
2786 \__xeCJK_msg_new:nn { key-unknown }
2787 {
2788     Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~`#1'.\\
2789     The~key~`#1'~is~being~ignored.
2790 }
(End definition for quiet and silent.)

```

5.17 xeCJK 初始化设置

```

\CJKsymbol
\CJKpunctsymbol 2791 \cs_new_nopar:Npn \CJKsymbol #1 {#1}
2792 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}
(End definition for \CJKsymbol and \CJKpunctsymbol.)

```

xeCJK 宏包的初始化设置。

```

2793 \keys_set:nn { xeCJK / options }
2794 {
2795     CJKglue = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
2796     CJKecglue = { ~ } ,
2797     xCJKecglue = false ,
2798     CheckSingle = false ,

```

```

2799 PlainEquation = false ,
2800 CheckFullRight = false ,
2801 CJKspace = false ,
2802 CJKmath = false ,
2803 CJKnumber = false ,
2804 xeCJKactive = true ,
2805 LocalConfig = true ,
2806 indentfirst = true ,
2807 Verb = env ,
2808 EmboldenFactor = 4 ,
2809 SlantFactor = 0.167 ,
2810 PunctStyle = quanjiao ,
2811 NewLineCS = { \par \[ } ,
2812 EnvCS = { \begin \end } ,
2813 NoBreakCS = { \footnote \footnotemark \nobreak } ,
2814 KaiMingPunct = { ^^^^3002 ^^^^ff0e ^^^^ff1f ^^^^ff01 } ,
2815 LongPunct = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^2025 ^^^^2026 } ,
2816 MiddlePunct = { ^^^^2014 ^^^^2015 ^^^^2027 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
2817 AllowBreakBetweenPuncts = false
2818 }
2819 \defaultCJKfontfeatures { Script = CJK }

```

执行宏包选项,并载入 fontspec 宏包。

```

2820 \ProcessKeysOptions { xeCJK / options }
2821 \RequirePackage { fontspec } [ 2011/09/13 ]

```

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

```

2822 \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。

```

2823 \bool_if:NT \g__xeCJK_indent_bool { \RequirePackage { indentfirst } }

```

对不能通过 \xeCJKsetup 设置的选项给出警告。

```

2824 \keys_define:nn { xeCJK / options }
2825 {
2826   LocalConfig .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2827   CJKnumber .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2828   indentfirst .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } }
2829 }
2830 \__xeCJK_msg_new:nn { option-invalid }
2831 {
2832   The~`#1'~option~only~can~be~set~in~the~optional~argument~to~the\\
2833   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
2834   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
2835 }

```

`\CJKrmdefault`

`\CJKsfdefault`

`\CJKttdefault`

`\CJKfamilydefault`

```

2836 \tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
2837 \tl_if_exist:NF \CJKsfdefault { \tl_gset:Nn \CJKsfdefault { sf } }
2838 \tl_if_exist:NF \CJKttdefault { \tl_gset:Nn \CJKttdefault { tt } }
2839 \tl_new:N \l__xeCJK_family_default_init_tl
2840 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
2841 \tl_gset:Nx \l__xeCJK_family_default_init_tl
2842 {
2843   \exp_not:N \__xeCJK_family_default_wrap:n
2844   {
2845     \tl_if_exist:NTF \CJKfamilydefault
2846       { \exp_not:N \CJKfamilydefault }
2847       { \exp_not:N \CJKrmdefault }
2848   }
2849 }
2850 \tl_gset_eq:NN \CJKfamilydefault \l__xeCJK_family_default_init_tl

```

(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。

```
2851 \NewDocumentCommand \xeCJKsetup { +m }
2852 {
2853   \keys_set:nn { xeCJK / options } {#1}
2854   \tex_ignorespaces:D
2855 }
```

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

\xeCJKsetemboldenfactor

\xeCJKsetslantfactor

```
2856 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
2857 { \xeCJKsetup { EmboldenFactor = {#1} } }
2858 \NewDocumentCommand \xeCJKsetslantfactor { m }
2859 { \xeCJKsetup { SlantFactor = {#1} } }
```

(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor.)

\punctstyle

\xeCJKplainchr

```
2860 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2861 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }
```

(End definition for \punctstyle and \xeCJKplainchr.)

\CJKsetecglue

```
2862 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2863 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
```

(End definition for \CJKsetecglue.)

\CJKspace

\CJKnospace

```
2864 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
2865 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }
```

(End definition for \CJKspace and \CJKnospace.)

\xeCJKallowbreakbetweenpuncts

\xeCJKnobreakbetweenpuncts

```
2866 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
2867 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
2868 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
2869 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }
```

(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts.)

\xeCJKenablefallback

\xeCJKdisablefallback

```
2870 \NewDocumentCommand \xeCJKenablefallback { }
2871 { \xeCJKsetup { AutoFallBack = true } }
2872 \NewDocumentCommand \xeCJKdisablefallback { }
2873 { \xeCJKsetup { AutoFallBack = false } }
```

(End definition for \xeCJKenablefallback and \xeCJKdisablefallback.)

\xeCJKsetcharclass

```
2874 \NewDocumentCommand \xeCJKsetcharclass { m m m }
2875 {
2876   \xeCJK_set_char_class:nnn {#1} {#2} {#3}
2877   \xeCJKResetPunctClass
2878 }
```

(End definition for \xeCJKsetcharclass.)

5.18 兼容性修补

`\nobreakspace` 按照 xunicode 的设计, 使用 `\DeclareUTFcharacter` 定义符号命令时, 如果命令已经存在, 那么它会检查它是否是“健壮”的。如果是, 则什么也不做; 否则使用 `\DeclareTextCommand` 来重定义它, 实际意义与原来相同, 原来的定义保存在名为 `?-\cs` 的宏中。这里的主要问题是 xunicode 的健壮性检查对像 `\nobreakspace` 这种原本由 `\DeclareRobustCommand` 定义的命令返回结果是“假”, 因而重定义它。如果编码发生变化, 使用 `\nobreakspace` 时, NFSS 就会报错。按照 NFSS 的机制, 我们用 `\ProvideTextCommandDefault` 对未声明的编码给出默认定义即可避免报错。其实 xunicode 重定义 `\nobreakspace` 的方式并不恰当。因为在 $\text{\LaTeX} 2_{\epsilon}$ 中, `\nobreakspace` 原本并不是由 `\DeclareTextCommand` 或 `\DeclareTextSymbol` 来定义, 这样按照 xunicode 的实现方式, `\nobreakspace` 永远都不会输出 `\char"00A0`。要想让 `\nobreakspace` 在 `\UTFencname` 编码下面输出 `\char"00A0`, 除非在 xunicode 载入前用 `\DeclareTextCommandDefault` 重新定义它:

```
\DeclareTextCommandDefault\nobreakspace{\leavevmode\nobreak\ }
```

这样我们也就不需要再做下面的处理。另外, 如果作者使用名字 `?-\cs` 来保存原来的定义的话, 也不需要下面的处理。因为下面实际上是定义了一个名为 `?-\nobreakspace` 的宏作为未声明编码的默认定义。但是作者使用了 `?-\cs`, 不知道这是有意为之还是一个疏忽。

```
2879 \cs_if_exist:cT { ? - \token_to_str:N \nobreakspace }
2880 {
2881     \exp_args:NNv \ProvideTextCommandDefault \nobreakspace
2882     { ? - \token_to_str:N \nobreakspace }
2883 }
(End definition for \nobreakspace.)

2884 \__xeCJK_msg_new:nn { conflict-package }
2885 {
2886     The~\#1'~package~is~too~old. \\\
2887     Please~update~an~up~to~date~version~of~it\\
2888     using~your~TeX~package~manager~or~from~CTAN.
2889 }
2890 \xeCJK_if_package_loaded:nTF { realscripts }
2891 {
2892     \@ifpackagelater { realscripts } { 2010/10/10 } { }
2893     {
2894         \__xeCJK_error:nx { conflict-package }
2895         {
2896             \xeCJK_if_package_loaded:nTF { xltextra }
2897             { xltextra } { realscripts }
2898         }
2899     }
2900 }
2901 {
2902     \cs_new_eq:NN \__xeCJK_dim_max:nn \dim_max:nn
2903     \cs_new_eq:NN \__xeCJK_dim_min:nn \dim_min:nn
2904     \__xeCJK_at_end_preamble:n
2905     {
2906         \xeCJK_if_package_loaded:nT { realscripts }
2907         {
2908             \@ifpackagelater { realscripts } { 2010/10/10 } { }
2909             {
2910                 \cs_gset_eq:NN \dim_max:nn \__xeCJK_dim_max:nn
2911                 \cs_gset_eq:NN \dim_min:nn \__xeCJK_dim_min:nn
2912             }
2913         }
2914         \cs_undefine:N \__xeCJK_dim_max:nn
2915         \cs_undefine:N \__xeCJK_dim_min:nn
2916     }
2917 }
```

`\fontfamily` 修改 `\fontfamily`, 使主要 CJK 字体族能随西文主要字体更新。

```
2918 \RenewDocumentCommand \fontfamily { m }
2919 {
```

```

2920 \tl_set:Nx \f@family {#1}
2921 \__xeCJK_update_family:n
2922 {
2923   \str_case:nnn {#1}
2924   {
2925     { \rmdefault } { \CJKfamily { \CJKrmdefault } }
2926     { \sfdefault } { \CJKfamily { \CJKsfdefault } }
2927     { \ttdefault } { \CJKfamily { \CJKttdefault } }
2928     { \familydefault } { \CJKfamily { \CJKfamilydefault } }
2929   }
2930   { }
2931 }
2932 }
2933 \cs_new_eq:NN \__xeCJK_update_family:n \use:n
(End definition for \fontfamily.)

```

`\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 `\emph{强调}` 文本,第二个空格可以被忽略掉。如果使用 `\xeCJKecglue` 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。

```

2935 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
2936 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
2937 \tl_replace_once:Nnn \sw@slant { \fix@penalty } { \xeCJK@fix@penalty }
(End definition for \xeCJK@fix@penalty.)

```

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

```

2938 \cs_new_protected_nopar:Npn \xeCJK@italiccorr
2939 {
2940   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
2941   {
2942     \xeCJK_if_last_node:nTF { default }
2943     {
2944       \tex_unkern:D \tex_unkern:D \@@italiccorr
2945       { \xeCJK_make_node:n { default } }
2946     }
2947     {
2948       \xeCJK_if_last_node:nTF { CJK }
2949       {
2950         \tex_unkern:D \tex_unkern:D \@@italiccorr
2951         { \xeCJK_make_node:n { CJK } } \use:n
2952       }
2953       {
2954         \xeCJK_if_last_node:nTF { CJK-space }
2955         {
2956           \tex_unkern:D \tex_unkern:D \@@italiccorr
2957           { \xeCJK_make_node:n { CJK-space } } \use:n
2958         }
2959         { \@@italiccorr \use_none:n }
2960       }
2961     }
2962   }

```

`\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:`)。

```

2961 {
2962   \exp_after:wN \exp_after:wN \exp_after:wN
2963   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2964   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2965   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN

```

```

2966         \xeCJK_ignore_spaces:w
2967     }
2968 }
2969 }
2970 { \@@italiccorr }
2971 }
(End definition for \xeCJK@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。

```

2973 \__xeCJK_after_end_preamble:n
2974 {
2975     \int_compare:nNf
2976     { \c_three + \seq_count:N \g__xeCJK_new_class_seq } = \xe@alloc@intercharclass
2977     {
2978         \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
2979         {
2980             \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
2981             { \__xeCJK_set_others_toks:n {#1} }
2982         }
2983     }
2984 }
2985 \cs_new_protected_nopar:Npn \__xeCJK_set_others_toks:n #1
2986 {
2987     \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
2988     \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
2989     {
2990         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2991         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2992         \xeCJK_app_inter_class_toks:nnx {##1} { Others }
2993         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
2994         \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
2995         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
2996         \xeCJK_if_blank_x:nT
2997         { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2998         {
2999             \xeCJK_copy_inter_class_toks:nnnn
3000             { Others } { Boundary } { Default } { Boundary }
3001         }
3002         \xeCJK_if_blank_x:nT
3003         { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
3004         {
3005             \xeCJK_copy_inter_class_toks:nnnn
3006             { Boundary } { Others } { Boundary } { Default }
3007         }
3008     }
3009 }
(End definition for \__xeCJK_set_others_toks:n.)

```

_xeCJK_group_begin: 用于保护下面歧义宽度标点的分组。

```

\_xeCJK_group_end: 3010 \cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
3011 \cs_new_eq:NN \__xeCJK_group_end: \group_end:

```

(End definition for _xeCJK_group_begin: and _xeCJK_group_end:.)

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。

```

3012 \__xeCJK_after_preamble:n
3013 {
3014     \tl_map_inline:nn
3015     {
3016         \textellipsis \textemdash \textperiodcentered \textcentereddot
3017         \textquoteleft \textquoteright \textquotedblleft \textquotedblright
3018     }
3019     {

```

```

3020 \cs_gset_nopar:Npx #1
3021 { \__xeCJK_group_begin: \makexeCJKinactive \exp_not:o {#1} \__xeCJK_group_end: }
3022 }
3023 \tl_put_left:Nn \tipaencoding { \makexeCJKinactive }
3024 \cs_new_eq:NN \__xeCJK_aux_r:n \r
3025 \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \__xeCJK_aux_r:n {#1} } }
3026 \xeCJK_if_package_loaded:nT { pifont }
3027 {
3028 \RenewDocumentCommand \Pifont { m }
3029 { \makexeCJKinactive \usefont { U } {#1} { m } { n } }
3030 }
3031 }

```

简单处理与 hyperref 宏包的兼容问题。

```

3032 \__xeCJK_after_end_preamble:n
3033 {
3034 \xeCJK_if_package_loaded:nT { hyperref }
3035 {
3036 \pdfstringdefDisableCommands
3037 {
3038 \__xeCJK_gobble_CJKfamily:
3039 \xeCJK_cs_clear:N \makexeCJKinactive
3040 \xeCJK_cs_clear:N \__xeCJK_group_begin:
3041 \xeCJK_cs_clear:N \__xeCJK_group_end:
3042 }
3043 }
3044 }

```

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

```

3045 \__xeCJK_after_end_preamble:n
3046 {
3047 \bool_if:nT
3048 { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
3049 { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
3050 }

```

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。

```

3051 \tl_set:cn { ver@CJK.sty } { 9999/99/99 }
3052 \tl_set_eq:cc { ver@CJKulem.sty } { ver@CJK.sty }
3053 \tl_set_eq:cc { ver@CJKpunct.sty } { ver@CJK.sty }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

3054 \cs_if_exist:NF \CJK@ifundefined
3055 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
3056 \NewDocumentCommand \xeCJKcaption { o m }
3057 {
3058 \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
3059 \use:x
3060 {
3061 \char_set_catcode_letter:n { 64 }
3062 \file_input:n { #2.cpx }
3063 \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
3064 }
3065 \XeTeXdefaultencoding "UTF-8"
3066 }

```

(End definition for \xeCJKcaption.)

由于 xeCJK 禁止 CJKulem 的载入,因此当使用 ctex 宏包的 fntef 选项时,就会出现 \normalem 没有定义的问题。此时改用 xeCJKfntef 以便载入 ulem。

```

3067 \cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
3068 { \RequirePackage { xeCJKfntef } }
3069 {
3070 \__xeCJK_at_end_preamble:n
3071 {

```

```

3072 \xeCJK_if_package_loaded:nF { xeCJKfntef }
3073 {
3074 \xeCJK_if_package_loaded:nTF { CJKfntef }
3075 { \RequirePackage { xeCJKfntef } }
3076 {
3077 \xeCJK_if_package_loaded:nT { ulem }
3078 { \RequirePackage { xeCJKfntef } }
3079 }
3080 }
3081 }
3082 }

```

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。

```

3083 \bool_if:NT \g__xeCJK_number_bool
3084 {
3085 \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
3086 \cs_set_protected:Npn \CJKaddEncHook #1#2
3087 { \cs_set:cpn { __xeCJK_enc_#1_hook: } {#2} }
3088 \cs_set_protected_nopar:Npn \Unicode #1#2
3089 { \tex_char:D \etex_numexpr:D (#1) * \c_two_hundred_fifty_six + (#2) \scan_stop: }
3090 \RequirePackage { CJKnumb }
3091 \cs_if_exist_use:c { __xeCJK_enc_ \CJK@UnicodeEnc _hook: }
3092 \tl_set:Nn \CJK@tenthousand { \Unicode { 78 } { 7 } }
3093 \tl_set:Nn \CJK@hundredmillion { \Unicode { 78 } { 191 } }
3094 }

```

最后引入本地配置文件。

```

3095 \bool_if:NT \g__xeCJK_config_bool
3096 {
3097 \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
3098 \@onefilewithoptions
3099 { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
3100 }
3101 </package>

```

5.19 xeCJKfntef

```

3102 <*fntef>

```

xeCJKfntef 不需要 CJKulem 宏包的支持,因此当使用 CJKfntef 时,需要另行载入 ulem。

```

3103 \PassOptionsToPackage { normalem } { ulem }
3104 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
3105 \ProcessOptions \scan_stop:
3106 \@ifpackageloaded { xeCJK }
3107 { }
3108 { \RequirePackage { xeCJK } }
3109 \RequirePackage { ulem }
3110 \RequirePackage { CJKfntef }
3111 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

```

\xeCJK_hook_for_ulem:

```

3112 \cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
3113 {
3114 \bool_if:NF \l__xeCJK_ulem_hook_used_bool
3115 {
3116 \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
3117 \xeCJKsetup { CheckFullRight = false }
3118 \bool_if:NTF \l__xeCJK_ulem_skip_punct_bool
3119 { \cs_set_eq:NN \__xeCJK_ulem_leader_type: \UL@leadtype }
3120 {
3121 \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
3122 \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:

```

```

3123     }
3124     \__xeCJK_ulem_initial:
3125     \xeCJK_glue_to_skip:nN
3126     {
3127         \cs_set_eq:NN \ \tex_space:D
3128         \cs_set_eq:NN \penalty \tex_penalty:D
3129         \cs_set_eq:NN \hskip \skip_horizontal:N
3130         \CJKglue
3131     } \l__xeCJK_ccglue_skip
3132     \xeCJK_glue_to_skip:nN
3133     {
3134         \cs_set_eq:NN \ \tex_space:D
3135         \cs_set_eq:NN \penalty \tex_penalty:D
3136         \cs_set_eq:NN \hskip \skip_horizontal:N
3137         \CJKecglue
3138     } \l__xeCJK_ecglue_skip
3139     \cs_set_protected_nopar:Npn \CJKglue
3140     { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
3141     \cs_set_protected_nopar:Npn \CJKecglue
3142     { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
3143 }
3144 }
3145 \bool_new:N \l__xeCJK_ulem_hook_used_bool
(End definition for \xeCJK_hook_for_ulem:.)

```

\CJK@UL 修改 CJKfntef 中的 \CJK@UL 和 \CJK@@UL 以适应下面的修改。

```

\CJK@@UL 3147 \cs_set_eq:NN \CJK@UL \CJK@@UL
3148 \tl_replace_once:Nnn \CJK@UL { \ULon }
3149 { \bool_set_true:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3150 \tl_replace_once:Nnn \CJK@@UL { \ULon }
3151 { \bool_set_false:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3152 \bool_new:N \l__xeCJK_ulem_skip_punct_bool
(End definition for \CJK@UL and \CJK@@UL.)

```

```

\__xeCJK_ulem_skip_punct_begin:
\__xeCJK_ulem_skip_punct_end: 3154 \cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_begin:
3155 { \xeCJK_cs_clear:N \UL@leadtype }
3156 \cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_end:
3157 { \cs_set_eq:NN \UL@leadtype \__xeCJK_ulem_leader_type: }
3158 \xeCJK_cs_clear:N \__xeCJK_ulem_leader_type:
(End definition for \__xeCJK_ulem_skip_punct_begin: and \__xeCJK_ulem_skip_punct_end:.)

```

__xeCJK_ulem_initial: 这里的设置是为了在下划线状态下,下划线可以自动跳过全角标点符号和正确的在它们前/后断行,并且与行首行末对齐。

```

3159 \cs_new_protected_nopar:Npn \__xeCJK_ulem_initial:
3160 {
3161     \xeCJK_swap_cs:NN
3162     \__xeCJK_Default_and_FullLeft_glue:N \__xeCJK_ulem_Default_and_FullLeft_glue:N
3163     \xeCJK_swap_cs:NN
3164     \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N
3165     \xeCJK_swap_cs:NN
3166     \__xeCJK_CJK_and_FullLeft_glue:N \__xeCJK_ulem_CJK_and_FullLeft_glue:N
3167     \xeCJK_swap_cs:NN
3168     \__xeCJK_CJK_and_FullRight_glue:N \__xeCJK_ulem_CJK_and_FullRight_glue:N
3169     \xeCJK_swap_cs:NN
3170     \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
3171     \xeCJK_swap_cs:NN \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
3172     \xeCJK_swap_cs:NN \xeCJK_FullLeft_and_Default: \__xeCJK_ulem_FullLeft_and_Default:
3173     \xeCJK_swap_cs:NN \xeCJK_FullLeft_and_CJK: \__xeCJK_ulem_FullLeft_and_CJK:
3174     \xeCJK_swap_cs:NN \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
3175     \xeCJK_swap_cs:NN \xeCJK_FullRight_and_CJK: \__xeCJK_ulem_FullRight_and_CJK:
3176     \xeCJK_swap_cs:NN \xeCJK_CJK_and_CJK:N \__xeCJK_ulem_CJK_and_CJK:N

```

```

3177 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3178 {
3179   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3180   {
3181     \str_if_eq:nnTF {##1} {####1}
3182     {
3183       \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
3184       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3185       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
3186       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3187     }
3188     {
3189       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
3190       { \__xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
3191     }
3192   }
3193 }
3194 }
(End definition for \__xeCJK_ulem_initial:.)

```

`\xeCJK_if_ulem_patch:TF` 在下划线状态下, `ulem` 宏包在数学模式或者盒子中使用 `\UL@hrest` 恢复 `_` 等的定义, 此时不需要使用 `\UL@stop` 和 `\UL@start` 来断开下划线而产生断点。

```

3195 \cs_new_nopar:Npn \xeCJK_if_ulem_patch:TF
3196 {
3197   \if_meaning:w \_ \LA@space
3198   \exp_after:wN \use_ii:nn
3199   \else:
3200     \exp_after:wN \use_i:nn
3201   \fi:
3202 }
(End definition for \xeCJK_if_ulem_patch:TF.)

```

`__xeCJK_ulem_CJK_and_CJK:N`

```

3203 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_CJK:N
3204 {
3205   \xeCJK_if_ulem_patch:TF
3206   {
3207     \xeCJK_class_group_end:
3208     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
3209     \__xeCJK_ulem_class_group_begin:
3210     \CJKsymbol
3211   }
3212   { \__xeCJK_ulem_CJK_and_CJK:N }
3213 }
(End definition for \__xeCJK_ulem_CJK_and_CJK:N.)

```

`_xeCJK_ulem_class_group_begin:`

```

3214 \cs_new_protected_nopar:Npn \_xeCJK_ulem_class_group_begin:
3215 {
3216   \xeCJK_class_group_begin:
3217   \xeCJK_clear_Boundary_and_CJK_toks:
3218   \xeCJK_select_font:
3219 }
(End definition for \_xeCJK_ulem_class_group_begin:.)

```

`__xeCJK_ulem_between_CJK_blocks:nnN`

```

3220 \cs_new_protected_nopar:Npn \__xeCJK_ulem_between_CJK_blocks:nnN #1#2
3221 {
3222   \xeCJK_if_ulem_patch:TF
3223   {
3224     \xeCJK_class_group_end:
3225     \UL@stop \__xeCJK_ulem_ccglue: \UL@start

```

```

3226         \xeCJK_class_group_begin:
3227         \xeCJK_clear_Boundary_and_CJK_toks:
3228         \__xeCJK_switch_font:nn {#1} {#2}
3229         \CJKsymbol
3230     }
3231     {
3232         \skip_horizontal:N \l__xeCJK_ccglue_skip
3233         \__xeCJK_switch_font:nn {#1} {#2}
3234         \CJKsymbol
3235     }
3236 }
(End definition for \__xeCJK_ulem_between_CJK_blocks:nnN.)

```

_xeCJK_ulem_Default_and_FullLeft_glue:N

```

3237 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
3238 {
3239     \xeCJK_if_ulem_patch:TF
3240     {
3241         \UL@stop
3242         \__xeCJK_ulem_skip_punct_begin:
3243         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3244         \UL@start
3245     }
3246     { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
3247 }
(End definition for \__xeCJK_ulem_Default_and_FullLeft_glue:N.)

```

_xeCJK_ulem_Boundary_and_FullLeft_glue:N

```

3248 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
3249 {
3250     \xeCJK_if_ulem_patch:TF
3251     {
3252         \UL@stop
3253         \__xeCJK_ulem_skip_punct_begin:
3254         \int_compare:nNf \etex_lastnodetype:D = \c_one
3255         { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
3256         \UL@start
3257     }
3258     { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
3259 }
(End definition for \__xeCJK_ulem_Boundary_and_FullLeft_glue:N.)

```

_xeCJK_ulem_CJK_and_FullLeft_glue:N

```

3260 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
3261 {
3262     \xeCJK_if_ulem_patch:TF
3263     {
3264         \xeCJK_class_group_end:
3265         \UL@stop
3266         \__xeCJK_ulem_skip_punct_begin:
3267         \__xeCJK_ulem_ccglue:
3268         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3269         \UL@start
3270         \__xeCJK_ulem_class_group_begin:
3271     }
3272     { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
3273 }
(End definition for \__xeCJK_ulem_CJK_and_FullLeft_glue:N.)

```

_xeCJK_ulem_Default_and_FullRight_glue:N

```

3274 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
3275 {

```



```

3276 \xeCJK_if_ulem_patch:TF
3277 {
3278   \UL@stop
3279   \__xeCJK_ulem_skip_punct_begin:
3280   \__xeCJK_punct_if_long:NTF {#1}
3281   { \__xeCJK_ulem_ccglue: }
3282   {
3283     \__xeCJK_punct_if_middle:NT {#1}
3284     {
3285       \xeCJK_no_break:
3286       \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
3287     }
3288     \xeCJK_no_break:
3289   }
3290   \UL@start
3291 }
3292 { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
3293 }
(End definition for \__xeCJK_ulem_Default_and_FullRight_glue:N.)

```

_xeCJK_ulem_CJK_and_FullRight_glue:N

```

3294 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
3295 {
3296   \xeCJK_if_ulem_patch:TF
3297   {
3298     \xeCJK_class_group_end:
3299     \__xeCJK_ulem_Default_and_FullRight_glue:N {#1}
3300     \__xeCJK_ulem_class_group_begin:
3301   }
3302   { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
3303 }
(End definition for \__xeCJK_ulem_CJK_and_FullRight_glue:N.)

```

_xeCJK_ulem_FullLeft_and_Default:

```

3304 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_Default:
3305 {
3306   \xeCJK_if_ulem_patch:TF
3307   {
3308     \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3309     {
3310       \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3311       \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3312       \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
3313       \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3314     }
3315     { \xeCJK_class_group_end: \UL@stop }
3316     \__xeCJK_ulem_skip_punct_end:
3317     \xeCJK_no_break:
3318     \UL@start
3319   }
3320   { \__xeCJK_ulem_FullLeft_and_Default: }
3321 }
(End definition for \__xeCJK_ulem_FullLeft_and_Default:.)

```

_xeCJK_ulem_FullLeft_and_CJK:

```

3322 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_CJK:
3323 {
3324   \xeCJK_if_ulem_patch:TF
3325   {
3326     \__xeCJK_ulem_FullLeft_and_Default:
3327     \__xeCJK_ulem_class_group_begin:
3328   }

```

```

3329     { \__xeCJK_ulem_FullLeft_and_CJK: }
3330   }

```

(End definition for __xeCJK_ulem_FullLeft_and_CJK:.)

_xeCJK_ulem_FullRight_and_Default:

```

3331 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_Default:
3332 {
3333   \xeCJK_if_ulem_patch:TF
3334   {
3335     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3336     \xeCJK_class_group_end:
3337     \UL@stop
3338     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3339     \__xeCJK_ulem_skip_punct_end:
3340     \UL@start
3341   }
3342   { \__xeCJK_ulem_FullRight_and_Default: }
3343 }

```

(End definition for __xeCJK_ulem_FullRight_and_Default:.)

_xeCJK_ulem_FullRight_and_CJK:

```

3344 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_CJK:
3345 {
3346   \xeCJK_if_ulem_patch:TF
3347   {
3348     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3349     \xeCJK_class_group_end:
3350     \UL@stop
3351     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3352     \__xeCJK_ulem_ccglue:
3353     \__xeCJK_ulem_skip_punct_end:
3354     \UL@start
3355     \__xeCJK_ulem_class_group_begin:
3356   }
3357   { \__xeCJK_ulem_FullRight_and_CJK: }
3358 }

```

(End definition for __xeCJK_ulem_FullRight_and_CJK:.)

__xeCJK_ulem_punct_hskip:n

```

3359 \cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_hskip:n
3360 {
3361   \xeCJK_if_ulem_patch:TF
3362   { \__xeCJK_ulem_hskip:n }
3363   { \__xeCJK_ulem_punct_hskip:n }
3364 }

```

(End definition for __xeCJK_ulem_punct_hskip:n.)

__xeCJK_ulem_glue:n 在下划线状态下的分别代替 \CJKglue 等。

__xeCJK_ulem_ccglue:n 3365 \cs_new_protected_nopar:Npn __xeCJK_ulem_glue:n #1

```

3366 {
3367   \xeCJK_if_ulem_patch:TF
3368   { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
3369   { \skip_horizontal:n {#1} }
3370 }
3371 \cs_new_protected_nopar:Npn \__xeCJK_ulem_ccglue:
3372 { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders }
3373 \cs_new_protected_nopar:Npn \__xeCJK_ulem_hskip:n #1
3374 {
3375   \int_compare:nNnTF \tex_lastkern:D = \c_three
3376   { \skip_horizontal:n {#1} }
3377   { \skip_set:Nn \UL@skip {#1} \UL@leaders }
3378 }

```

(End definition for __xeCJK_ulem_glue:n, __xeCJK_ulem_ccglue:, and __xeCJK_ulem_hskip:n)

\CJKUnderdot 使用 xeCJK 时, CJKfntef 中的 \CJKUnderdot 和 \CJKUnderanysymbol 在汉字之间不能断行。因此需要我们在这里修改它们。

```

3379 \RenewDocumentCommand \CJKUnderdot { m }
3380 {
3381   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3382   \CJK@preUnderdot
3383   \__xeCJK_make_under_symbol:n { \CJK@underdotSkip }
3384   \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3385   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
3386   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3387   #1
3388   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3389   \cs_set_eq:NN \CJKsymbol \__xeCJK_save_under_dot_CJKsymbol:N
3390   \CJK@postUnderdot
3391   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3392   \tex_ignorespaces:D
3393 }
3394 \box_new:N \g__xeCJK_under_symbol_box
(End definition for \CJKUnderdot.)

```

\CJKUnderanysymbol

```

3395 \RenewDocumentCommand \CJKUnderanysymbol { m m m }
3396 {
3397   \group_begin:
3398   \hbox_set:Nn \CJK@underdotBox {#2}
3399   \__xeCJK_make_under_symbol:n {#1}
3400   \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3401   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
3402   #3
3403   \group_end:
3404   \tex_ignorespaces:D
3405 }
(End definition for \CJKUnderanysymbol.)

```

__xeCJK_make_under_symbol:n

```

3406 \cs_new_protected:Npn \__xeCJK_make_under_symbol:n #1
3407 {
3408   \hbox_set:Nn \l__xeCJK_tmp_box { ^^^^4e00 }
3409   \vbox_gset_to_ht:Nnn \g__xeCJK_under_symbol_box \c_zero_dim
3410   {
3411     \skip_vertical:n {#1}
3412     \hbox_to_zero:n
3413     {
3414       \tex_kern:D - \box_wd:N \l__xeCJK_tmp_box
3415       \tex_hss:D \box_use:N \CJK@underdotBox \tex_hss:D
3416     }
3417     \tex_vss:D
3418   }
3419 }
(End definition for \__xeCJK_make_under_symbol:n.)

```

__xeCJK_under_CJKsymbol:N

```

3420 \cs_new_protected_nopar:Npn \__xeCJK_under_CJKsymbol:N #1
3421 {
3422   \__xeCJK_save_under_dot_CJKsymbol:N {#1}
3423   \hbox_overlap_left:n { \box_use:N \g__xeCJK_under_symbol_box }
3424   { \xeCJK_make_node:n { CJK } }
3425   \xeCJK_ignore_spaces:w
3426 }
(End definition for \__xeCJK_under_CJKsymbol:N.)

```

CJKfilltwosides 修改 CJKfilltwosides 环境,使其不必重定义 \CJKsymbol。

```
3427 \RenewDocumentEnvironment { CJKfilltwosides } { m }
3428 {
3429   \dim_set:Nn \l__xeCJK_fill_width_dim {#1}
3430   \cs_set_eq:NN \CJKglue \tex_hfil:D
3431   \cs_set_eq:NN \ \ \__xeCJK_fill_newline:
3432   \mode_if_vertical:T { \tex_indent:D }
3433   \tex_vbox:D \c_group_begin_token
3434   \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
3435   \tex_ignorespaces:D
3436 }
3437 {
3438   \tex_unskip:D
3439   \c_group_end_token
3440   \c_group_end_token
3441 }
3442 \dim_new:N \l__xeCJK_fill_width_dim
(End definition for CJKfilltwosides.)
```

__xeCJK_fill_newline: CJKfilltwosides 环境里的断行命令 \。

```
3443 \cs_new_protected_nopar:Npn \__xeCJK_fill_newline:
3444 {
3445   \tex_unskip:D
3446   \c_group_end_token
3447   \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
3448   \tex_ignorespaces:D
3449 }
(End definition for \__xeCJK_fill_newline:.)
3450 \</fntef>
```

5.20 xeCJK.cfg

```
3451 \<*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。

```
3452
```

```
3453 \</config>
```

版本历史

v3.1.0

General: 使用 <code>xtemplate</code> 宏包的机制来组织标点符号的处理。	42
删除多余的 <code>default-icorr</code> 结点。	27
取消 <code>\cprotect</code> 的外部宏限制。	75
改用 <code>indentfirst</code> 宏包处理缩进的问题。	70
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。	66
放弃对 <code>\outer</code> 宏的特殊处理。	1
<code>_xeCJK_switch_font:n</code> : 改进定义,加快切换速度。	60
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑 <code>\spaceskip</code> 不为零的情况。	15
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。	68
<code>\xeCJK@fix@penalty</code> : 采用通过不修改原语 <code>\/</code> 的方式对修复倾斜校正。	73
<code>\xeCJK_fallback_loop:Nn</code> : 调整备用字体的循环方式。	51
<code>\xeCJK_glyph_if_exist:N</code> : 改进 <code>fontspec</code> 宏包中定义的 <code>\font_glyph_if_exist:NnTF</code> 。	14
<code>\xeCJK_hook_for_ulem::</code> : 简化对 <code>ulem</code> 宏包的兼容补丁。	76
<code>\xeCJK_visible_space_fallback::</code> : 调整 <code>fontspec</code> 的后备可视空格符号,以便于使用时对齐。	67
<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。	66

v3.1.1

General: 不再依赖 <code>xpatch</code> 宏包。	1
增加 <code>NewLineCS</code> 和 <code>EnvCS</code> 选项。	37
增加小宏包 <code>xeCJKfntef</code> ,用于处理下划线的问题。	76
对于与 <code>xltxtra</code> 的冲突给出错误警告。	72
<code>_xeCJK_check_single_space:NN</code> : <code>CheckSingle</code> 支持段末“汉字 + 汉字 + 空格 + 汉字/标点”的形式。	37
<code>_xeCJK_set_char_class_eq:n</code> : 交换参数的顺序。	23
<code>_xeCJK_set_verb_exspace::</code> : 调整间距的计算方法。	67
<code>\fontfamily</code> : 修改主要 CJK 字体族的自动更新方式。	72
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。	34
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。	38
<code>InlineEnv</code> : 改变行内环境的设置方式,从而使用 <code>\str_case:x:nnn</code> 代替原来的 <code>\clist_if_in:NnTF</code> 来判断是否是行内环境。	38
<code>\xeCJK_check_single:NNw</code> : 改进定义,减少使用 <code>peek</code> 函数的次数。	36

<code>\xeCJK_fallback_loop:Nn</code> : 修正备用字体的不正确警告。	51
<code>\xeCJK_hook_for_ulem::</code> : 完全处理下划线里的标点符号的有关问题。	76
<code>\xeCJK_peek_catcode_ignore_spaces:N</code> : 新增有省略空格标识的 <code>peek</code> 函数。	16
<code>\xeCJK_save_class:n</code> : 使用 <code>\xeCJK_save_class:n</code> 保存 <code>X₂TeX</code> 预定义的字符类别。	18
<code>\xeCJK_set_char_class:nnn</code> : 在文档中设置字符类别时不重复设置 <code>\catcode</code> 。	23
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。	35

v3.1.2

General: 修正重定义 <code>\CJKfamilydefault</code> 无效的问题,恢复容错能力。	64
解决在下划线状态下使用 <code>\makebox</code> 时的错误。	78
<code>_xeCJK_check_single_space:NN</code> : 使用 <code>\xeCJK_if_CJK_class:N</code> 来代替 <code>\int_case:nnn</code> 判断是否是 CJK 字符类。	37
<code>_xeCJK_family_unknown_warning:n</code> : 在没有定义任何 CJK 字体的情况下,不再重复给出字体没有定义的警告。	61
<code>\fontfamily</code> : 不将参数完全展开。	72
<code>\nobreakspace</code> : 修正非 <code>\UTFencname</code> 编码下面 <code>xunicode</code> 重定义的 <code>\nobreakspace</code> 会失效的问题。	72
<code>\xeCJK_token_value_charcode:N</code> : 考虑 <code>charcode</code> 大于等于 <code>0x10000</code> 的情况。	17

v3.2.0

General: 增加 <code>IVS</code> 字符类用于处理异体字选择符。	19
增加 <code>Verb</code> 选项。	66
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑到 <code>\spacefactor</code> 和 <code>\xspaceskip</code> 的情况。	15
<code>\l_xeCJK_family_tl</code> : 不将其初始化为 <code>\CJKfamilydefault</code> 。	61
<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。	62
<code>\xeCJK_Boundary_and_FullLeft:N</code> : 当全角左标点前面是 <code>hlist:none</code> 、 <code>glue</code> 和 <code>penalty</code> 等节点时,压缩其左空白。	31
<code>\xeCJK_FullLeft_and_Default::</code> : 修正 <code>xeCJK</code> 使西文在部分情况下无法断词的问题。	30

v3.2.1

General: 调整 <code>Verb</code> 选项: 在命令 <code>\verb</code> 里使用时,不破坏标点禁则,增加值 <code>env+</code> 。	66
---	----

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,而直立的数字表示对应项使用时所在的行号。

Symbols

<code>\@italiccorr</code>	2936, 2944, 2950, 2956, 2959, 2970
<code>\ifpackagelater</code>	19, 22, 2892, 2908
<code>\ifpackageloaded</code>	3106
<code>\@onefilewithoptions</code>	3098
<code>\@onlypreamble</code>	1188, 1775, 1782, 2435, 2436, 2437, 2438, 2439, 2469
<code>\@pgkextension</code>	51
<code>\[</code>	2811
<code>\</code>	5, 6, 7, 13, 14, 15, 16, 296, 297, 1224, 1276, 1277, 1758, 1759, 1760, 1772, 1773, 1851, 2395, 2496, 2559, 2560, 2561, 2562, 2567, 2788, 2832, 2833, 2886, 2887, 3431

<code>_xeCJK_Boundary_and_FullLeft_glue:N</code>	839, 847, 3170
<code>_xeCJK_CJK_and_FullLeft_glue:N</code>	829, 834, 3166
<code>_xeCJK_CJK_and_FullRight_glue:N</code>	886, 892, 892, 902, 3168
<code>_xeCJK_CJK_class_tl:c</code>	386
<code>_xeCJK_CJK_class_tl:n</code>	247, 250, 252
<code>_xeCJK_Default_and_FullLeft_glue:N</code>	815, 824, 3162
<code>_xeCJK_Default_and_FullRight_glue:N</code>	861, 874, 892, 902, 3164
<code>_xeCJK_add_fake_bold:n</code>	2099, 2099, 2148, 2167
<code>_xeCJK_add_fake_slant:n</code>	2099, 2104, 2159
<code>_xeCJK_add_font:nV</code>	2125, 2127
<code>_xeCJK_add_font:nn</code>	2093, 2093, 2097

_xeCJK_add_font:nv	2115	_xeCJK_if_it_or_sl:nTF	2116, 2133
_xeCJK_add_font:nx	2175	_xeCJK_if_it_or_sl_p:n	2133, 2153, 2164
_xeCJK_add_font_if_new:nn	2093, 2095, 2098, 2169	_xeCJK_if_map_font_select:n	2182
_xeCJK_add_font_if_new:nv	2119	_xeCJK_if_map_font_select:nT	2173
_xeCJK_add_font_if_new:nx	2177	_xeCJK_if_map_font_select:nTF	2182
_xeCJK_add_special_punct:nn	1301, 1304, 1307, 1311, 1328	_xeCJK_if_map_font_select_p:n	2182
_xeCJK_after_end_preamble:n	52, 59, 1283, 2973, 3032, 3045	_xeCJK_info:nx	36, 42
_xeCJK_after_preamble:n	52, 57, 2621, 3012	_xeCJK_info:nxx	36, 43, 2302
_xeCJK_at_end_preamble:n	52, 55, 2517, 2904, 3070	_xeCJK_int_until_do:wn	182, 183, 184, 185
_xeCJK_aux_r:n	3024, 3025	_xeCJK_kerning_width_or_ratio:nnN	1602, 1603, 1621, 1621
_xeCJK_block_select_font:n	2305, 2309, 2309	_xeCJK_make_under_symbol:n	3383, 3399, 3406, 3406
_xeCJK_calc_kerning_margin:NN	1564, 1584, 1584	_xeCJK_map_it_sl:n	2171
_xeCJK_calc_kerning_margin:nnN	1597, 1606, 1651, 1651	_xeCJK_map_it_sl:nn	2161, 2171
_xeCJK_check_family:V	2044, 2229	_xeCJK_margin_width_or_ratio:n	1518, 1519, 1539, 1539
_xeCJK_check_family:n	2081, 2081, 2090	_xeCJK_msg_def_family_map:n	2396, 2398, 2561, 2568
_xeCJK_check_num_range:nnNN	415, 430, 438, 438, 2604	_xeCJK_msg_family_map:n	2092, 2395, 2409, 2566, 2567
_xeCJK_check_single_save:N	1031, 1038, 1055, 1064, 1089, 1090, 1100, 1101, 1103, 1109, 1112, 1121, 1124	_xeCJK_msg_new:nn	36, 36, 294, 1222, 1274, 1756, 1770, 1847, 2091, 2308, 2393, 2494, 2557, 2564, 2786, 2830, 2884
_xeCJK_check_single_space:NN	1063, 1095, 1095	_xeCJK_original_kerning_margin:NN	1565, 1572, 1572, 1635
_xeCJK_class_csname:n	273, 276, 279, 284, 287, 292, 292, 293, 386, 389, 418, 1212, 1230, 2987	_xeCJK_parse_features:	2042, 2109, 2109
_xeCJK_clear_Boundary_and_CJK_toks:n	514, 515	_xeCJK_parse_features:nn	2109, 2110, 2111
_xeCJK_copy_family:nn	2256, 2256, 2275	_xeCJK_parse_sub_class_features:n	2482, 2499, 2499
_xeCJK_copy_family:xx	2317	_xeCJK_pass_features:	2043, 2187, 2187
_xeCJK_copy_sub_family:n	2219, 2227	_xeCJK_peek_after_do:w	216, 227
_xeCJK_dim_max:nn	2902, 2910, 2914	_xeCJK_peek_catcode_false:w	190, 209
_xeCJK_dim_min:nn	2903, 2911, 2915	_xeCJK_peek_catcode_ignore_spaces_branches:w	193, 195, 200
_xeCJK_document_left_hook:	69, 75	_xeCJK_peek_catcode_true:w	189, 206
_xeCJK_document_right_hook:	71, 77	_xeCJK_peek_ignore_spaces_branches:w	218, 220, 224
_xeCJK_error:n	36, 37, 1282	_xeCJK_prop_put_aux:n	2054, 2064
_xeCJK_error:nx	36, 38, 274, 285, 1218, 1752, 1780, 2782, 2894	_xeCJK_prop_put_aux:nn	2054, 2066
_xeCJK_fallback_save_CJKsymbol:N	1814, 1821, 1829, 1838, 1844	_xeCJK_punct_break:	928, 1296, 1298
_xeCJK_family_csname:n	2070, 2243, 2320, 2328, 2328, 2346	_xeCJK_punct_dim_csname:nn	1361, 1361, 1368, 1372
_xeCJK_family_default_wrap:n	2519, 2533, 2840, 2843	_xeCJK_punct_dim_csname:nnn	1361, 1363, 1370, 1374, 1668, 1727
_xeCJK_family_nfss_csname:n	2086, 2271, 2272, 2328, 2329, 2330, 2334, 2344	_xeCJK_punct_dim_csname:nnnn	1361, 1364, 1365, 1376
_xeCJK_family_unknown_warning:n	2380, 2380, 2391	_xeCJK_punct_glue:NN	766, 766, 784, 794, 803, 810, 825, 835, 856, 898, 969, 3243, 3255, 3268, 3286, 3313, 3338, 3351
_xeCJK_family_unknown_warning:x	2365, 2378	_xeCJK_punct_hskip:n	768, 774, 3171
_xeCJK_family_use:x	2284, 2293, 2324, 2328, 2330, 2363, 2377	_xeCJK_punct_if_long:NT	1702
_xeCJK_fill_newline:	3431, 3443, 3443	_xeCJK_punct_if_long:NFT	894, 1347, 1678, 3280
_xeCJK_font_csname:n	2277, 2278, 2290, 2292, 2311, 2314	_xeCJK_punct_if_long_p:N	1347
_xeCJK_get_map_font:n	2175, 2177, 2180, 2180, 2184	_xeCJK_punct_if_middle:NT	867, 879, 888, 3283
_xeCJK_get_sub_features:Vn	1872	_xeCJK_punct_if_middle:NFT	780, 790, 897, 1347, 1468, 1484, 1505, 3308
_xeCJK_get_sub_features:nn	1935, 1942, 1942, 1969	_xeCJK_punct_if_middle_p:N	1347
_xeCJK_get_sub_features:wn	1942, 1948, 1962	_xeCJK_punct_if_mixed_width:NFT	1347, 1471, 1517
_xeCJK_gobble_CJKfamily:	2371, 2371, 3038	_xeCJK_punct_if_mixed_width_p:N	1347
_xeCJK_gobble_CJKfamily:wn	2372, 2373	_xeCJK_punct_if_right:N	1347
_xeCJK_group_begin:	3010, 3010, 3021, 3040	_xeCJK_punct_if_right:NFT	1347, 1577, 1580, 1657, 1660
_xeCJK_group_end:	3010, 3011, 3021, 3041	_xeCJK_punct_if_right_p:N	1347
_xeCJK_gset_family_cs:x	2045, 2068, 2068	_xeCJK_punct_kern:NN	775, 775, 908, 917, 926, 935
_xeCJK_gset_family_nfss_cs:xx	2077, 2250, 2328, 2331	_xeCJK_punct_min_bound:NN	1590, 1593, 1637, 1637
_xeCJK_gset_punct_dim:nnnx	1361, 1375, 1706	_xeCJK_punct_rule:NN	759, 759, 782, 792, 802, 808, 821, 831, 844, 868, 880, 889, 967, 3311, 3335, 3348
_xeCJK_gset_punct_dim:nnx	1361, 1373, 1477, 1552, 1672, 1673, 1674, 1679, 1689, 1691, 1730	_xeCJK_punct_width_or_ratio:nN	1469, 1472, 1473, 1526, 1526
_xeCJK_gset_punct_dim:nnx	1361, 1371, 1698, 1700	_xeCJK_remove_duplicate_keys:N	2039, 2051, 2051
_xeCJK_if_font_select:n	2133	_xeCJK_save_CJK_class:n	383, 388, 1267
_xeCJK_if_font_select:nF	2118, 2124, 2126, 2176	_xeCJK_save_FullRight_check:	947, 957
_xeCJK_if_font_select:nTF	2113, 2133	_xeCJK_save_FullRight_symbol:N	948, 958, 974
_xeCJK_if_font_select_p:n	2133, 2155, 2157, 2165	_xeCJK_save_family_info:	2046, 2203, 2203
_xeCJK_if_it_or_sl:n	2138		
_xeCJK_if_it_or_sl:nT	2161		

\bool_set_true:N	198, 676, 686, 1194, 1994, 2003, 2659, 3116, 3149
\Boundary	299
\box_new:N	30, 3394
\box_use:N	3415, 3423
\box_wd:N	3414
C	
\c__xeCJK_CJK_chars_clist	308, 334, 473, 2592
\c__xeCJK_config_ext_tl	3097, 3099
\c__xeCJK_encoding_tl	2336, 2587, 2589, 2822, 2822
\c__xeCJK_FullLeft_chars_clist	308, 313, 468, 2592
\c__xeCJK_FullRight_chars_clist	308, 320, 469, 2594
\c__xeCJK_HalfLeft_chars_clist	308, 308, 466
\c__xeCJK_HalfRight_chars_clist	308, 310, 467
\c__xeCJK_IVS_chars_clist	308, 370, 475
\c__xeCJK_left_tl	784, 794, 814, 821, 825, 828, 831, 835, 838, 844, 856, 868, 880, 889, 906, 924, 1290, 1290, 1452, 1568, 1581, 1642, 1647, 1658, 1673, 1689, 1695, 3243, 3255, 3268, 3310, 3313
\c__xeCJK_math_family_tl	2585, 2587, 2589
\c__xeCJK_math_tl	2572, 2573, 2576, 2577, 2586, 2588, 2590
\c__xeCJK_NormalSpace_chars_clist	308, 312, 474
\c__xeCJK_package_ext_tl	48, 51
\c__xeCJK_punct_style_plain_tl	1670, 1729, 1747, 1755
\c__xeCJK_right_tl	782, 792, 802, 803, 808, 810, 860, 873, 885, 898, 915, 933, 966, 967, 969, 1290, 1291, 1452, 1567, 1578, 1643, 1648, 1661, 1674, 1691, 1696, 3286, 3311, 3335, 3338, 3348, 3351
\c_catcode_other_token	1045, 1060
\c_eleven	551, 611, 853, 1010
\c_empty_tl	1948, 1965, 1966
\c_four	119, 140, 2978
\c_fourteen	2629, 2639
\c_group_begin_token	517, 3433, 3434, 3447
\c_group_end_token	518, 3439, 3440, 3446
\c_math_toggle_token	709, 1107
\c_max_dim	1381, 1387, 1398, 1401, 1403, 1405, 1482, 1507, 1528, 1532, 1541, 1596, 1623, 1631
\c_minus_one	852
\c_one	266, 300, 851, 1690, 1858, 2459, 2616, 2978, 3254
\c_one_fp	1382, 1388, 1406
\c_one_thousand	113, 167
\c_seven	134, 149, 2598
\c_space_tl	682
\c_space_token	197, 222
\c_ten	604, 620
\c_ten_thousand	45
\c_thirteen	854
\c_three	118, 139, 302, 1377, 1378, 1409, 1692, 2618, 2976, 3375
\c_two	117, 127, 133, 301, 771, 1488, 1512, 2617, 2691, 2716, 2724, 2731
\c_two_hundred_fifty_five	303
\c_two_hundred_fifty_six	3089
\c_xeCJK_math_fam_int	2590, 2598
\c_xeCJK_space_skip_tl	111, 111, 552, 612
\c_zero	44, 267, 299, 2615, 2649, 2940
\c_zero_dim	763, 764, 1395, 1407, 1672, 1673, 1674, 1679, 1713, 1730, 2672, 2689, 2795, 3409
\c_zero_fp	1402, 1404, 1531, 1599, 1626
\c_zero_skip	115, 124, 129, 146, 799
\char_set_catcode:nn	3063
\char_set_catcode:ignore:n	268
\char_set_catcode_letter:n	3061
\char_set_catcode_other:n	419
\char_value_catcode:n	3063
\CheckFullRight	940
\CheckSingle	1024
\CJK	299
\CJK@eul	3147, 3147, 3150
\CJK@hundredmillion	3093
\CJK@ifundefined	3054, 3055
\CJK@postUnderdot	3390
\CJK@preUnderdot	3382
\CJK@tenthousand	3092
\CJK@ul	3147, 3147, 3148
\CJK@underdotBox	3398, 3415
\CJK@underdotSkip	3383
\CJK@UnicodeEnc	3085, 3091
\CJKaddEncHook	3086
\CJKecglue	561, 564, 569, 585, 606, 623, 666, 670, 677, 687, 689, 712, 2627, 2636, 2668, 2670, 2671, 3137, 3141
xCJKecglue	3
\CJKfamily	5, 2349, 2349, 2372, 2445, 2463, 2553, 2925, 2926, 2927, 2928
\CJKfamilydefault	6, 2320, 2321, 2508, 2520, 2522, 2530, 2535, 2537, 2539, 2540, 2544, 2548, 2551, 2553, 2579, 2580, 2836, 2845, 2846, 2850, 2928
\CJKfilltwosides	3427
\CJKfixedspacing	2676
\CJKfontspec	6, 2440, 2448
\CJKglue	599, 630, 657, 661, 731, 804, 835, 895, 2626, 2635, 2666, 3130, 3139, 3430
\CJKmath	2570
\CJKknospace	2864, 2865
\CJKknumber	2757
\CJKpunctsymbol	822, 832, 845, 910, 929, 963, 2791, 2792
\CJKrmdefault	6, 2402, 2413, 2421, 2526, 2540, 2543, 2544, 2836, 2836, 2847, 2925
\CJKsetecglue	2862, 2862, 2863
\CJKsfdefault	6, 2403, 2414, 2427, 2527, 2836, 2837, 2926
\CJKspace	695, 2864, 2864
\CJKsymbol	540, 593, 731, 733, 735, 1252, 1253, 1812, 1814, 1815, 1820, 1821, 2791, 2791, 3210, 3229, 3234, 3384, 3385, 3389, 3400, 3401
\CJKttdefault	6, 2404, 2415, 2432, 2528, 2836, 2838, 2927
\CJKunderanysymbol	3395, 3395
\CJKunderdot	3379, 3379
\clist_clear:c	2015
\clist_clear:N	1945, 1952, 2017, 2055
\clist_concat:NNN	2037, 2486, 2591, 2593
\clist_const:Nn	308, 310, 312, 313, 320, 334, 370
\clist_count:N	1858
\clist_gconcat:ccN	398
\clist_gconcat:NNN	411
\clist_gset:Nn	1312, 2468
\clist_if_empty:cF	2191
\clist_if_in:NnT	2501
\clist_map_break:	2513
\clist_map_function:nN	388
\clist_map_inline:cn	456
\clist_map_inline:Nn	400, 1315, 1352, 1869, 2595
\clist_map_inline:nn	532, 544, 571, 738, 754, 756, 1157, 1166, 1210, 1268
\clist_new:c	277, 288

<code>\clist_new:N</code>	34, 35, 1311, 2466	2635, 2636, 2670, 2671, 3055, 3119, 3127, 3128, 3129, 3134, 3135, 3136, 3147, 3157, 3385, 3389, 3401, 3430, 3431
<code>\clist_put_left:cn</code>	1873	
<code>\clist_put_left:cx</code>	2101, 2106	
<code>\clist_put_right:Nn</code>	2059, 2060, 2198	
<code>\clist_put_right:Nx</code>	2193	
<code>\clist_remove_all:Nn</code>	2207, 2239, 2503	
<code>\clist_set:Nn</code>	397, 1964, 2035, 2480	
<code>\clist_set:Nx</code>	1857, 2511	
<code>\clist_set_eq:cN</code>	1959	
<code>\cs:w</code>	104, 3049	
<code>\cs_end:</code>	104, 247, 1357, 3049	
<code>\cs_generate_variant:Nn</code>	252, 411, 412, 481, 491, 497, 972, 1135, 1221, 1238, 1287, 1719, 1737, 1854, 1881, 1969, 1970, 1971, 2050, 2063, 2090, 2097, 2098, 2144, 2275, 2341, 2391, 2465, 2493, 2516	
<code>\cs_gset_eq:cc</code>	2270	
<code>\cs_gset_eq:NN</code>	103, 2710, 2910, 2911, 3384, 3400	
<code>\cs_gset_nopar:Npx</code>	73, 3020	
<code>\cs_gset_protected_nopar:cpx</code>	2070, 2243, 2334	
<code>\cs_if_eq:NNF</code>	945, 1029, 1812	
<code>\cs_if_eq:NNT</code>	955, 1037, 1820	
<code>\cs_if_eq:NNTF</code>	3067	
<code>\cs_if_exist:cT</code>	2879	
<code>\cs_if_exist:cTF</code>	2319, 2344	
<code>\cs_if_exist:NF</code>	727, 1280, 3054	
<code>\cs_if_exist_p:N</code>	3048	
<code>\cs_if_exist_use:c</code>	3091	
<code>\cs_if_exist_use:cF</code>	2281, 2290, 2311, 2703	
<code>\cs_if_exist_use:cTF</code>	1458, 2346	
<code>\cs_if_free:NTF</code>	3055	
<code>\cs_new:Npn</code>	84, 86	
<code>\cs_new_eq:cN</code>	293	
<code>\cs_new_eq:NN</code>	92, 93, 238, 243, 450, 517, 518, 774, 902, 2297, 2424, 2475, 2676, 2840, 2863, 2902, 2903, 2933, 2935, 3010, 3011, 3024	
<code>\cs_new_nopar:Npn</code>	80, 158, 230, 232, 236, 241, 250, 292, 389, 482, 963, 1313, 1314, 1361, 1363, 1365, 1367, 1369, 1526, 1539, 1572, 1584, 1621, 1637, 1651, 1720, 2180, 2278, 2328, 2329, 2330, 2398, 2409, 2791, 2792, 3195	
<code>\cs_new_protected:Npn</code>	55, 57, 59, 94, 182, 184, 186, 214, 975, 999, 1172, 3406	
<code>\cs_new_protected_nopar:Npn</code>	36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 69, 71, 82, 101, 168, 195, 220, 271, 282, 383, 395, 413, 428, 438, 453, 479, 484, 486, 492, 498, 504, 513, 515, 595, 642, 648, 707, 731, 752, 759, 766, 775, 778, 788, 798, 800, 806, 812, 824, 826, 834, 836, 847, 858, 871, 883, 892, 903, 912, 922, 931, 964, 973, 1043, 1058, 1095, 1105, 1114, 1127, 1208, 1228, 1239, 1317, 1328, 1339, 1371, 1373, 1375, 1446, 1550, 1666, 1687, 1725, 1826, 1832, 1862, 1912, 1942, 1962, 2008, 2030, 2051, 2064, 2066, 2068, 2081, 2093, 2095, 2099, 2104, 2109, 2111, 2145, 2171, 2187, 2203, 2214, 2227, 2256, 2279, 2288, 2298, 2309, 2331, 2371, 2374, 2380, 2453, 2476, 2499, 2574, 2602, 2655, 2677, 2701, 2712, 2719, 2938, 2985, 3112, 3154, 3156, 3159, 3203, 3214, 3220, 3237, 3248, 3260, 3274, 3294, 3304, 3322, 3331, 3344, 3359, 3365, 3371, 3373, 3420, 3443	
<code>\cs_new_protected_nopar:Npx</code>	2445	
<code>\cs_set:cpn</code>	3087	
<code>\cs_set_eq:NN</code>	96, 97, 98, 188, 419, 420, 452, 677, 682, 689, 947, 948, 949, 950, 957, 958, 1031, 1032, 1038, 1296, 1298, 1814, 1815, 1821, 2073, 2372, 2519, 2626, 2627, 2635, 2636, 2670, 2671, 3055, 3119, 3127, 3128, 3129, 3134, 3135, 3136, 3147, 3157, 3385, 3389, 3401, 3430, 3431	
<code>\cs_set_nopar:Npn</code>	3025	
<code>\cs_set_protected:Npn</code>	3086	
<code>\cs_set_protected_nopar:Npn</code>	661, 670, 687, 2666, 2668, 3088, 3139, 3141	
<code>\cs_to_str:N</code>	2444	
<code>\cs_undefine:c</code>	1320, 1343, 2086	
<code>\cs_undefine:N</code>	99, 2533, 2914, 2915	
<code>\curr@fontshape</code>	2679, 2687, 2698, 2703, 2714	
<code>\CurrentOption</code>	3104	
D		
<code>\DeclareExpandableDocumentCommand</code>	2373	
<code>\DeclareInstance</code>	1767	
<code>\DeclareObjectType</code>	1377	
<code>\DeclareOption</code>	3104	
<code>\DeclareSymbolFont</code>	2586	
<code>\DeclareTemplateCode</code>	1409	
<code>\DeclareTemplateInterface</code>	1378	
<code>\Default</code>	299	
<code>\defaultCJKfontfeatures</code>	6, 2466, 2467, 2469, 2819	
<code>\dim_compare:nNnF</code>	2724	
<code>\dim_compare:nNnTF</code>	1482, 1507, 1528, 1541, 1596, 1623, 1631	
<code>\dim_const:cn</code>	645	
<code>\dim_eval:n</code>	160, 771, 1554, 1574, 1653, 1701, 2728	
<code>\dim_gset:cn</code>	645	
<code>\dim_if_exist:cTF</code>	644	
<code>\dim_max:nn</code>	1479, 1492, 1586, 1593, 1639, 2689, 2902, 2910	
<code>\dim_min:nn</code>	1492, 1502, 1548, 1641, 1646, 2903, 2911	
<code>\dim_new:N</code>	31, 3442	
<code>\dim_ratio:nn</code>	2731	
<code>\dim_set:Nn</code>	1454, 1693, 3429	
<code>\dim_use:N</code>	1699, 1714, 1722, 2692, 2716	
<code>\document</code>	73, 76	
E		
<code>xeCJKactive</code>	3	
<code>\EditInstance</code>	1779	
<code>\else:</code>	109, 177, 179, 202, 207, 226, 248, 257, 640, 1350, 1358, 2142, 3199	
<code>\EmboldenFactor</code>	1886	
<code>\end</code>	2812	
<code>\EnvCS</code>	1142	
<code>Verb</code>	5	
<code>\etex_currentgrouplevel:D</code>	2649	
<code>\etex_currentgrouptype:D</code>	2629, 2639	
<code>\etex_dimexpr:D</code>	1511, 1535, 1545, 1609, 1635	
<code>\etex_fontcharwd:D</code>	1699, 2692	
<code>\etex_glueshrink:D</code>	154	
<code>\etex_gluestretch:D</code>	153	
<code>\etex_iffontchar:D</code>	108	
<code>\etex_lastnodetype:D</code>	551, 604, 611, 620, 849, 1010, 3254	
<code>\etex_numexpr:D</code>	3089	
<code>NewLineCS</code>	3	
<code>NewLineCS+</code>	3	
<code>NewLineCS-</code>	3	
<code>\exp_after:wN</code>	103, 104, 170, 171, 185, 199, 200, 205, 206, 208, 209, 223, 224, 227, 233, 1052, 1054, 1948, 2715, 2716, 2962, 2963, 2964, 2965, 3049, 3198, 3200	
<code>\exp_args:Nc</code>	276, 1354, 2463	
<code>\exp_args:NNv</code>	2881	
<code>\exp_args:No</code>	1051	

\exp_args:NV	83	\g__xeCJK_family_name_prop	2085, 2200, 2200, 2247, 2262, 2333, 2583
\exp_not:c	741	\g__xeCJK_features_id_prop	1973, 1974, 1975, 1976, 1977, 1978, 1979, 2012, 2110, 2189
\exp_not:N	204, 727, 2077, 2247, 2248, 2251, 2336, 2337, 2338, 2526, 2527, 2528, 2689, 2843, 2847	\g__xeCJK_indent_bool	2760, 2823
\exp_not:n	489, 495, 2073, 2074, 2519, 2687	\g__xeCJK_last_punct_tl	777, 777, 780, 782, 784, 790, 792, 794, 802, 803, 808, 810, 820, 830, 843, 866, 878, 887, 907, 908, 909, 916, 917, 918, 925, 926, 927, 934, 935, 936, 966, 967, 969, 3308, 3310, 3311, 3313, 3335, 3338, 3348, 3351
\exp_not:o	76, 3021	\g__xeCJK_math_bool	2554, 2570
\exp_not:V	502, 510, 2075, 2076, 2512, 2846	\g__xeCJK_new_class_seq	269, 270, 279, 2976, 2980
\exp_not:v	2181, 2194	\g__xeCJK_non_CJK_class_seq	377, 379, 380, 736
\exp_stop_f:	108, 175, 178, 256, 1723	\g__xeCJK_number_bool	2759, 3083
F		\g__xeCJK_punct_style_seq	1704, 1760, 1766, 1769
\f@baselineskip	2734	\g__xeCJK_punct_width_tl	1309, 1461, 1463
\f@encoding	2336, 2721	\g__xeCJK_slant_factor_fp	1885, 1901, 1905, 2006, 2027
\f@family	2337, 2722, 2920	\g__xeCJK_spacefactor_int	113, 126, 143, 161, 164, 166, 167, 575
\f@series	2278	\g__xeCJK_special_punct_clist	1311, 1311, 1312, 1315, 1352
\f@shape	2278	\g__xeCJK_sub_key_seq	1911, 1911, 1914, 2019
\f@size	2278, 2679, 2687, 2698, 2703, 2714, 2730	\g__xeCJK_under_symbol_box	3394, 3409, 3423
\FallBack	1972	\g__xeCJK_unknown_family_seq	2384, 2386, 2392
\familydefault	2524, 2928	\g__xeCJK_verb_exspace_skip	2667, 2669, 2697, 2700
\fi:	109, 179, 180, 185, 210, 211, 228, 248, 257, 640, 1350, 1358, 2142, 2644, 3201	\g_fontspeg_encoding_tl	2721, 2822
\file_input:n	3062	\group_align_safe_begin:	192, 217, 1067
\fix@penalty	2935, 2937	\group_align_safe_end:	189, 190, 216, 1070, 1076
\fontfamily	2918, 2918	\group_begin:	70, 1864, 2032, 2072, 2205, 2681, 3010, 3397
\fontsize	2726	\group_end:	70, 1879, 2048, 2078, 2212, 2685, 3011, 3403
\fontspec_set_family:Nnn	2074	H	
\fontspec_visible_space:	2709	\HalfLeft	304
\fontspec_visible_space_fallback:	2710	\HalfRight	304
\footnote	2813	\hbox_overlap_left:n	3423
\footnotemark	2813	\hbox_set:Nn	170, 3398, 3408
\fp_compare:nNnTF	1531, 1599, 1626	\hbox_to_zero:n	3412
\fp_eval:n	161	CheckFullRight	5
\fp_gset:Nn	1893, 1901	CheckSingle	3
\fp_new:N	1884, 1885	\hskip	3129, 3136
\fp_set:Nn	1995, 2004	I	
\fp_set_eq:NN	2026, 2027	\icprotect	3048, 3049
\fp_use:c	1534, 1544, 1628	MiddlePunct	4
\fp_use:N	1510, 1608, 1633, 2102, 2107	MiddlePunct+	4
\FullLeft	299	MiddlePunct-	4
\FullRight	299	\if_case:w	175, 178, 2625
G		\if_catcode:w	203
\g__xeCJK_after_end_preamble_hook_tl	54, 60, 65, 72	\if_cs_exist:w	247, 1357
\g__xeCJK_after_preamble_hook_tl	53, 58, 64, 68	\if_dim:w	639
\g__xeCJK_at_end_preamble_hook_tl	52, 56, 63, 70	\if_int_compare:w	183, 255, 1349
\g__xeCJK_auto_fake_bold_bool	1882, 1889, 1892, 2024	\if_meaning:w	197, 222, 3197
\g__xeCJK_auto_fake_slant_bool	1883, 1897, 1900, 2025	\if_predicate:w	2140
\g__xeCJK_base_class_seq	377, 377, 378, 1241	\if_BooleanF	2353
\g__xeCJK_CJK_class_seq	377, 382, 385, 514, 2988	\if_BooleanT	393, 1186, 1196, 1205, 2363
\g__xeCJK_CJK_sub_class_seq	1181, 1181, 1257, 1266, 2206, 2482, 3177, 3179	\if_BooleanTF	2357
\g__xeCJK_class_seq	269, 269, 278, 289, 378, 521	\if_CTEX@fntef	3067
\g__xeCJK_config_bool	2742, 2745, 2750, 2756, 3095	\if_InstanceExistTF	1750, 1764, 1778
\g__xeCJK_config_name_tl	2746, 2751, 2755, 3099	\if_NoValueF	3058
\g__xeCJK_default_features_clist	2038, 2466, 2468	\if_NoValueTF	446, 447, 2444
\g__xeCJK_embolden_factor_fp	1884, 1893, 1904, 1997, 2026	\indentfirst	2757
\g__xeCJK_family_font_name_prop	1850, 1866, 2083, 2200, 2201, 2208, 2230, 2233, 2263, 2382, 2478, 2504, 2507, 2534, 2547	\InlineEnv	1148
\g__xeCJK_family_font_options_prop	2200, 2202, 2210, 2236, 2240, 2264, 2483	\int_case:nnn	849
\g__xeCJK_family_int	2010, 2029, 2459	\int_compare:nNnF	2649, 2975, 3254
		\int_compare:nNnT	2639

\int_compare:nNnTF	113, 126, 143, 417, 1858, 2629, 2940, 3375	\l__xeCJK_env_cs_seq	1143
\int_compare:p:nNn	551, 604, 611, 620	\l__xeCJK_fallback_family_tl	1865, 1867, 1871, 1872, 1874, 1875, 1876, 1877
\int_const:cn	287	\l__xeCJK_fallback_first_bool	1853
\int_const:Nn	2590	\l__xeCJK_family_default_init_tl	2520, 2839, 2841, 2850
\int_div_truncate:nn	163	\l__xeCJK_family_name_tl	1923, 2034, 2044, 2045, 2209, 2211, 2218, 2231, 2237, 2245, 2248
\int_eval:n	251, 2459	\l__xeCJK_fill_width_dim	3429, 3434, 3442, 3447
\int_gincr:N	2010	\l__xeCJK_fixed_margin_ratio_fp	1419
\int_gset_eq:NN	167, 575	\l__xeCJK_fixed_margin_width_dim	1418
\int_if_even:nF	1021	\l__xeCJK_fixed_punct_ratio_fp	1413
\int_if_even:nTF	1012	\l__xeCJK_fixed_punct_width_dim	1412
\int_if_exist:cF	1230	\l__xeCJK_font_features_clist	2037, 2039, 2041
\int_if_exist:cTF	273, 284, 1212	\l__xeCJK_font_name_bf_tl	2125, 2127
\int_incr:N	425, 435, 1014, 2609	\l__xeCJK_font_name_tl	1867, 1868, 1957, 1958, 2036, 2076, 2209
\int_max:nn	447	\l__xeCJK_font_options_clist	2035, 2038, 2207, 2211
\int_min:nn	446	\l__xeCJK_fontspec_family_tl	2074, 2077
\int_new:N	27, 28, 29, 166, 2029, 2620	\l__xeCJK_fontspec_options_clist	2017, 2041, 2075, 2193, 2198
\int_set:cn	2987	\l__xeCJK_inline_env_case_tl	1130, 1174, 1176, 1178
\int_set:Nn	416, 431, 442, 446, 447, 455	\l__xeCJK_inline_env_seq	1152, 1159, 1160, 1167, 1171, 1175
\int_set_eq:NN	443, 2615, 2616, 2617, 2618	\l__xeCJK_kerning_margin_minimum_dim	1438, 1587
\int_step_inline:nnnn	2978	\l__xeCJK_kerning_margin_ratio_fp	1437, 1633
\int_to_hexadecimal:n	1851	\l__xeCJK_kerning_margin_width_dim	1436, 1631, 1632
\int_while_do:nNnn	1010	\l__xeCJK_kerning_total_ratio_fp	1430, 1599, 1608
\int_zero:N	1009	\l__xeCJK_kerning_total_width_dim	1429, 1596, 1597
\ItalicFeatures	1973	\l__xeCJK_margin_minimum_dim	1426, 1480
\ItalicFont	1973	\l__xeCJK_middle_margin_ratio_fp	1423, 1510
\IVS	304	\l__xeCJK_middle_margin_width_dim	1422, 1507, 1508
J			
CJKecglue	3	\l__xeCJK_middle_punct_ratio_fp	1417
CJKglue	3	\l__xeCJK_middle_punct_width_dim	1416
CJKmath	3	\l__xeCJK_min_bound_to_kerning_bool	1428, 1589
CJKnumber	3	\l__xeCJK_mixed_margin_ratio_fp	1421
CJKspace	3	\l__xeCJK_mixed_margin_width_dim	1420
K			
\KaiMingPunct	1292	\l__xeCJK_mixed_punct_ratio_fp	1415
\keys_define:nn	259, 657, 666, 695, 940, 979, 1024, 1148, 1179, 1292, 1738, 1807, 1886, 1915, 1981, 1987, 2570, 2612, 2738, 2757, 2763, 2824	\l__xeCJK_mixed_punct_width_dim	1414
\keys_set:nn	2793, 2853	\l__xeCJK_new_line_cs_case_tl	1137, 1140, 1146
\keys_set_known:nVN	2040	\l__xeCJK_new_line_cs_seq	1137
\keyval_parse:NNn	2063	\l__xeCJK_no_break_cs_case_tl	968, 1006
\keyval_parse:NNV	2054	\l__xeCJK_no_break_cs_seq	1006
\KeyValue	1383, 1384, 1385, 1386, 1389, 1390, 1391, 1392	\l__xeCJK_optimize_kerning_bool	1431, 1592
L			
\l__keys_module_tl	2788	\l__xeCJK_optimize_margin_bool	1425, 1491, 1501
\l__peek_search_token	188, 204	\l__xeCJK_peek_ignore_spaces_bool	191, 198, 213, 711, 715, 1062, 1071, 1083, 1088
\l__xeCJK_add_font_prop	2011, 2094, 2096, 2197	\l__xeCJK_plain_equation_bool	1079, 1180
\l__xeCJK_add_min_bound_to_margin_bool	1424, 1547	\l__xeCJK_reserve_space_bool	630, 697, 722
\l__xeCJK_auto_fake_bold_bool	1991, 1994, 2024, 2148, 2163	\l__xeCJK_same_align_margin_dim	1432
\l__xeCJK_auto_fake_slant_bool	2000, 2003, 2025, 2150	\l__xeCJK_same_align_ratio_fp	1433
\l__xeCJK_ccglue_skip	662, 665, 3131, 3140, 3232, 3372	\l__xeCJK_slant_factor_fp	2004, 2027, 2107
\l__xeCJK_check_single_cs_case_tl	1117, 1126, 1139, 1145	\l__xeCJK_sub_cancel_bool	1189, 1192, 1194, 1201, 1203, 1215
\l__xeCJK_current_coord_tl	2283, 2285, 2292, 2294, 2313, 2325, 2714, 2717	\l__xeCJK_sub_family_tl	2218, 2222, 2229, 2234, 2241, 2243, 2251
\l__xeCJK_different_align_margin_dim	1434	\l__xeCJK_sub_font_name_tl	1949, 1951, 1955, 1956, 1957, 1958, 1960, 1966, 1967
\l__xeCJK_different_align_ratio_fp	1435	\l__xeCJK_sub_font_options_clist	1945, 1952, 1959, 1964
\l__xeCJK_ecglue_skip	613, 671, 688, 693, 3138, 3142	\l__xeCJK_sub_key_seq	1926, 2018, 2216
\l__xeCJK_embolden_factor_fp	1995, 2026, 2102	\l__xeCJK_tmp_box	30, 170, 3408, 3414
\l__xeCJK_enabled_global_setting_bool	1411, 1456, 1558	\l__xeCJK_tmp_dim	31, 1454, 1482, 1487, 1495, 1693, 1701, 1714
\l__xeCJK_enabled_kerning_bool	1427, 1563	\l__xeCJK_tmp_prop	33, 2053, 2056, 2065, 2067
\l__xeCJK_env_cs_case_tl	1140, 1143, 1146	\l__xeCJK_tmp_skip	32, 1015, 1017, 1022

\prop_gput:Nnn	2341	\setCJKsansfont	5, 2403, <u>2419</u> , 2425, 2437
\prop_gput:NnV	2268	\SetSymbolFont	2588
\prop_gput:NVV	2208, 2210, 2233, 2240	\sfdefault	2527, 2926
\prop_gput:Nxx	2333	\shapedefault	2587, 2589
\prop_if_empty:NF	2382	\silent	<u>2763</u>
\prop_if_empty:NTF	2534	\skip_add:Nn	1017
\prop_map_break:n	2548	\skip_gset:Nn	2697
\prop_map_function:NN	2110	\skip_horizontal:N	799, 1022, 2667, 3129, 3136, 3232
\prop_map_inline:Nn	1979, 2012, 2056, 2189, 2197, 2547	\skip_horizontal:n	774, 2669, 2795, 3369, 3376
\prop_new:N	33, 1973, 2200, 2201, 2202	\skip_if_eq:nnTF	115, 124, 129, 146
\prop_put:Nnn	1974, 1975, 1976, 1977, 1978, 2065, 2067, 2094	\skip_if_eq_p:nn	552, 612, 613
\prop_put_if_new:Nnn	2096	\skip_new:N	32, 665, 693, 2700
\ProvideTextCommandDefault	2881	\skip_set:Nn	171, 2672, 3377
\PunctStyle	<u>1738</u>	\skip_set_eq:NN	1015, 3372
\punctstyle	<u>2860</u> , <u>2860</u>	\skip_use:N	171
\PunctWidth	<u>1292</u>	\skip_vertical:n	3411
Q			
\q_stop	85, 86, 89, 233, 236, 241	\SlantedFeatures	<u>1973</u>
\quiet	<u>2763</u>	\SlantedFont	<u>1973</u>
R			
\r	3024, 3025	\SlantFactor	<u>1886</u>
\RenewDocumentCommand	2918, 3028, 3379, 3395	\SplitArgument	410
\RenewDocumentEnvironment	3427	\str_case:nnn	2923
\RequirePackage	21, 24, 2821, 2823, 3068, 3075, 3078, 3090, 3108, 3109, 3110	\str_case_x:nnn	1135, 2400, 2411, 2524
\reverse_if:N	183	\str_case_x:non	1129
\rmdefault	2526, 2925	\str_if_eq:nnF	402, 527, 568, 584, 2300
S			
\scan_stop:	72, 170, 188, 764, 1133, 1513, 1535, 1545, 1612, 1635, 2709, 3089, 3105	\str_if_eq:nnT	17, 2122
\selectfont	2338, 2723, 2735	\str_if_eq:nnTF	523, 1245, 1927, 1965, 2147, 2303, 3181
\seq_clear:N	2018	\str_if_eq_p:nn	1710, 1711, 2141, 2154, 2156
\seq_count:N	2976	\str_if_eq_x:nnTF	81, 234, 2540
\seq_gclear:c	1321	\sw@slant	2937
\seq_gput_right:cn	1325, 1335	T	
\seq_gput_right:Nn	278, 289, 385, 1266, 1766, 1914, 2386	\tex_baselineskip:D	2795
\seq_gput_right:Nv	279	\tex_char:D	3089
\seq_gremove_all:cn	1344	\tex_font:D	104, 108, 117, 118, 119, 127, 133, 134, 139, 140, 149, 1699, 2691, 2692, 2716, 2724, 2731
\seq_gset_eq:NN	378	\tex_fontdimen:D	117, 118, 119, 127, 133, 134, 139, 140, 149, 2691, 2716, 2724, 2731
\seq_gset_from_clist:Nn	380	\tex_global:D	2607
\seq_if_in:cnF	1332	\tex_hbox:D	3434, 3447
\seq_if_in:NnF	989, 1159, 2384, 2980	\tex_hfil:D	3430
\seq_map_function:NN	514, 2481	\tex_hss:D	3415
\seq_map_inline:cn	1319	\tex_iftrue:D	3067
\seq_map_inline:Nn	521, 736, 1002, 1175, 1241, 1257, 1704, 2019, 2206, 2216, 2988, 3177, 3179	\tex_ignorespaces:D	72, 749, 753, 2368, 2451, 2473, 2854, 3392, 3404, 3435, 3448
\seq_new:c	1316	\tex_indent:D	3432
\seq_new:N	269, 270, 377, 379, 382, 978, 1171, 1181, 1769, 1911, 2392	\tex_italiccorrection:D	576, 577
\seq_put_right:Nn	989, 1160, 1926	\tex_kern:D	650, 651, 776, 3414
\seq_remove_all:Nn	994, 1167	\tex_lastkern:D	639, 3375
\seq_set_from_clist:Nn	1152	\tex_lastskip:D	171, 552, 612, 613, 1015, 1017
\seq_set_split:Nnn	983	\tex_let:D	3049
\seq_use:Nnnn	1760	\tex_number:D	164
\setCJKfallbackfamilyfont	6, 7, 1855, 1855	\tex_penalty:D	44, 45, 3128, 3135
\setCJKfamilyfont	5, 2406, <u>2440</u> , 2440	\tex_rightskip:D	2672
\setCJKmainfont	5, 2402, <u>2419</u> , 2419, 2424, 2435	\tex_romannumeral:D	201, 225
\setCJKmathfont	6, 2436, <u>2571</u> , 2571	\tex_space:D	3127, 3134
\setCJKmonofont	5, 2404, <u>2419</u> , 2430, 2438	\tex_spacefactor:D	575
\setCJKromanfont	2424, 2439	\tex_spaceskip:D	115, 121, 124, 144, 149, 153, 154
		\tex_the:D	104, 483
		\tex_unkern:D	717, 2944, 2950, 2956
		\tex_unskip:D	555, 616, 1018, 3438, 3445
		\tex_vbox:D	3433
		\tex_vrule:D	761
		\tex_vss:D	3417

\xeCJK_add_font_features:n	2476, 2476, 2493	\xeCJK_FullRight_and_CJK:	735, 800, 800, 3175
\xeCJK_add_font_features:x	2472	\xeCJK_FullRight_and_Default:	753, 806, 806, 3174
\xeCJK_allow_break:	44, 44, 1296	\xeCJK_FullRight_and_FullLeft:N	922, 922
\xeCJK_app_inter_class_toks:nnn	492, 492, 497, 569, 749	\xeCJK_FullRight_and_FullRight:N	931, 931
\xeCJK_app_inter_class_toks:nnx	2992	\xeCJK_FullRight_symbol:N	869, 881, 890, 920, 938, 948, 950, 958, 963, 963
\xeCJK_Boundary_and_FullLeft:N	746, 836, 836	\xeCJK_get_inter_class_toks:nn	482, 482, 489, 495, 500, 506, 2993, 2995, 2997, 3003
\xeCJK_Boundary_and_FullRight:N	748, 871, 871	\xeCJK_get_punct_bounds:NN	814, 828, 838, 860, 873, 885, 906, 915, 924, 933, 966, 1666, 1666, 3310
\xeCJK_calc_punct_dimen:f	1677	\xeCJK_get_punct_kerning:NN	1725, 1725, 1737
\xeCJK_calc_punct_dimen:N	1687, 1687, 1719	\xeCJK_get_punct_kerning:oN	907, 916, 925, 934
\xeCJK_check_for_glue:	589, 595, 595, 2670	\xeCJK_glue_to_skip:nN	168, 168, 662, 671, 688, 3125, 3132
\xeCJK_check_FullRight:	945, 949, 955, 964, 964	\xeCJK_glyph_bounds:NN	1690, 1692, 1720, 1720
\xeCJK_check_FullRight_symbol:Nw	950, 973, 973	\xeCJK_glyph_if_exist:N	106
\xeCJK_check_single:NNw	1046, 1054, 1058, 1058	\xeCJK_glyph_if_exist:NTF	106, 1828, 1837
\xeCJK_check_single:Nw	1029, 1032, 1037, 1043, 1043	\xeCJK_glyph_if_exist_p:N	106
\xeCJK_check_single_cs:NNn	1072, 1073, 1114, 1114	\xeCJK_gset_mathcode:nnnn	2597, 2602, 2602
\xeCJK_check_single_env:nnNn	1119, 1127, 1127	\xeCJK_hook_for_ulem:	3111, 3112, 3112
\xeCJK_check_single_equation:NNnNw	1084, 1085, 1105, 1105	\xeCJK_if_blank_x:n	173
\xeCJK_CJK_and_CJK:N	730, 731, 731, 1029, 1031, 1032, 1037, 1038, 3176	\xeCJK_if_blank_x:nT	2996, 3002
\xeCJK_CJK_and_FullLeft:N	826, 826	\xeCJK_if_blank_x:nTF	173, 442, 1461, 2184, 2351
\xeCJK_CJK_and_FullRight:N	883, 883	\xeCJK_if_blank_x_p:n	173, 440, 1050
\xeCJK_class_group_begin:	517, 517, 536, 590, 816, 840, 862, 875, 3216, 3226	\xeCJK_if_CJK_class:N	245
\xeCJK_class_group_end:	517, 518, 542, 703, 793, 796, 809, 970, 3207, 3224, 3264, 3298, 3312, 3315, 3336, 3349	\xeCJK_if_CJK_class:NTF	245, 1097, 1099
\xeCJK_class_num:n	389, 389, 405, 455, 462, 480, 483, 1349	\xeCJK_if_CJK_class_p:N	245
\xeCJK_clear_Boundary_and_CJK_toks:	513, 513, 539, 591, 819, 842, 865, 877, 3217, 3227	\xeCJK_if_last_node:n	637
\xeCJK_clear_inter_class_toks:nn	484, 484, 516, 538, 818, 864	\xeCJK_if_last_node:nT	565
\xeCJK_copy_inter_class_toks:nnnn	498, 498, 524, 526, 528, 1243, 1244, 1256, 1259, 1260, 2990, 2991, 2999, 3005	\xeCJK_if_last_node:NTF	564, 637, 2942, 2948, 2954
\xeCJK_cs_case_keys_define:nNNnn	975, 975, 1005, 1136, 1142	\xeCJK_if_last_node_p:n	558, 559, 598, 603, 619, 621, 627, 628, 637
\xeCJK_cs_clear:N	92, 92, 3039, 3040, 3041, 3121, 3122, 3155, 3158	\xeCJK_if_package_loaded:n	46
\xeCJK_cs_gclear:N	92, 93	\xeCJK_if_package_loaded:nF	2769, 2776, 3072
\xeCJK_declare_char_class:nn	395, 395, 412, 1236	\xeCJK_if_package_loaded:nT	2906, 3026, 3034, 3077
\xeCJK_declare_char_class:nV	466, 467, 468, 469, 473, 474, 475	\xeCJK_if_package_loaded:nTF	46, 61, 2781, 2890, 2896, 3074
\xeCJK_declare_char_class:nx	392, 1214	\xeCJK_if_package_loaded_p:n	46, 3048
\xeCJK_declare_sub_char_class:nnn	1228, 1228, 1238	\xeCJK_if_same_class:NN	253
\xeCJK_declare_sub_char_class:nxx	1185	\xeCJK_if_same_class:NNTF	253, 1601
\xeCJK_def_node:nn	642, 642, 653, 654, 655, 656	\xeCJK_if_same_class_p:NN	253
\xeCJK_Default_and_FullLeft:nN	812, 812	\xeCJK_if_ulem_patch:TF	3195, 3195, 3205, 3222, 3239, 3250, 3262, 3276, 3296, 3306, 3324, 3333, 3346, 3361, 3367
\xeCJK_Default_and_FullRight:nN	858, 858	\xeCJK_ignore_spaces:w	705, 707, 707, 2671, 2966, 3425
\xeCJK_fallback_loop:Nn	1830, 1832, 1832, 1839	\xeCJK_int_until_do:nn	182, 182, 421, 432, 2605
\xeCJK_fallback_test_glyph:N	1812, 1815, 1820, 1826, 1826	\xeCJK_inter_class_toks:nnc	742, 757
\xeCJK_family_if_exist:x	2342	\xeCJK_inter_class_toks:nnn	479, 479, 481, 485, 534, 542, 546, 573, 587, 701, 730, 732, 734, 745, 747, 750, 3183, 3185, 3189
\xeCJK_family_if_exist:xF	2315, 2537	\xeCJK_inter_class_toks:nnx	488, 494, 502, 510, 740
\xeCJK_family_if_exist:xT	1842, 2245, 2258	\xeCJK_make_node:n	577, 580, 581, 642, 648, 704, 718, 2945, 2951, 2957, 3424
\xeCJK_family_if_exist:xTF	1834, 2342, 2359, 2376, 2543, 2576, 2579	\xeCJK_new_class:n	271, 271, 304, 305, 306, 307, 1232
\xeCJK_family_if_exist_use:x	2354, 2357, 2374, 2374	\xeCJK_new_sub_key:n	1234, 1911, 1912, 1972
\xeCJK_font_gset_to_current:c	101, 101, 2285, 2294, 2325, 2717	\xeCJK_no_break:	44, 45, 783, 786, 793, 796, 898, 899, 905, 914, 919, 937, 1006, 1020, 1109, 1110, 1121, 1122, 1298, 3285, 3288, 3312, 3317
\xeCJK_fontspec:nn	2453, 2453, 2465	\xeCJK_peek_after_ignore_spaces:nw	214, 214, 974
\xeCJK_fontspec:VV	2489	\xeCJK_peek_catcode_ignore_spaces:NTF	186, 186, 709, 1060
\xeCJK_fontspec:xx	2450	\xeCJK_pre_inter_class_toks:nnn	486, 486, 491, 585, 1247, 1261, 1263, 1270
\xeCJK_FullLeft_and_CJK:	733, 778, 778, 3173	\xeCJK_pre_inter_class_toks:nnx	2994
\xeCJK_FullLeft_and_Default:	788, 788, 3172	\xeCJK_punct_kerning_process:NN	1444, 1550, 1550
\xeCJK_FullLeft_and_FullLeft:N	903, 903	\xeCJK_punct_margin_process:NN	1443, 1446, 1446
\xeCJK_FullLeft_and_FullRight:N	912, 912		
\xeCJK_FullRight_and_Boundary:	749, 751, 752, 752, 945, 947, 949, 955, 957		

\xeCJK_replace_inter_class_toks:nnnn	504, 504, 1251	\xeCJKDeclareSubCJKBlock	8, 1182, 1182, 1188, 1225
\xeCJK_reverse:nnn	80, 80, 1452, 2181	\xeCJKdisablefallback	2870, 2872
\xeCJK_save_class:nn	282, 282, 299, 300, 301, 302, 303	\xeCJKEditPunctStyle	10, 1776, 1776, 1782
\xeCJK_select_font:	537, 592,	\xeCJKenablefallback	2870, 2870
	817, 841, 863, 876, 1677, 2276, 2279, 2297, 2304, 2682, 3218	\xeCJKnobreak	11, 1007, 1007
\xeCJK_select_font:x	1836, 2276, 2288	\xeCJKnobreakbetweenpuncts	2866, 2868
\xeCJK_set_char_class:nnn	404, 450, 450, 452, 2876	\xeCJKplainchr	2860, 2861
\xeCJK_set_family:nnn	2030, 2030, 2050	\xeCJKResetCharClass	9, 471, 471, 478
\xeCJK_set_family:Vnn	2461	\xeCJKResetPunctClass	
\xeCJK_set_family:Vvv	1875, 2222		9, 393, 464, 464, 476, 1186, 1196, 1205, 2877
\xeCJK_set_family:xxx	1860, 2421, 2427, 2432, 2441, 2446, 2572	\xeCJKRestoreSubCJKBlock	9, 1189, 1199
\xeCJK_set_family_fallback:nnN	1862, 1862, 1881	\xeCJKsetcharclass	2874, 2874
\xeCJK_set_family_fallback:xxN	1859	\xeCJKsetecglue	2863
\xeCJK_set_mathfont:	2554, 2574, 2574	\xeCJKsetemboldenfactor	2856, 2856
\xeCJK_set_visible_space_font:	2704, 2712, 2712	\xeCJKsetkern	9, 1288, 1288
\xeCJK_swap_cs:NN	94, 94, 3161,	\xeCJKsetslantfactor	2856, 2858
	3163, 3165, 3167, 3169, 3171, 3172, 3173, 3174, 3175, 3176	\xeCJKsetup	2, 1283, 2665, 2834, 2851, 2851, 2857, 2859,
\xeCJK_tl_remove_outer_braces:N	82, 82		2860, 2861, 2862, 2864, 2865, 2867, 2869, 2871, 2873, 3117
\xeCJK_tl_remove_outer_braces:n	82, 83, 84, 1944	\xeCJKsetwidth	9, 1285, 1285
\xeCJK_token_value_charcode:N	231, 232, 232, 1699, 1723	\xeCJKVerbAddon	11, 2647, 2647, 2676
\xeCJK_token_value_class:N	230, 230, 247, 256, 1349	\xetex_if_engine:F	10
\xeCJK_visible_space_fallback:	2701, 2701, 2710	\XeTeXcharclass	231, 424, 434, 457, 462
\xeCJK_xetex_mathcode:w	238, 243, 2607	\XeTeXcharglyph	1723
\xeCJKactive	259	\XeTeXdefaultencoding	3058, 3065
\xeCJKallowbreakbetweenpuncts	2866, 2866	\XeTeXglyphbounds	1276, 1280, 1722
\xeCJKCancelSubCJKBlock	9, 1189, 1190	\XeTeXinterchartokenstate	266, 267, 2940
\xeCJKcaption	3054, 3056	\XeTeXinterchartoks	480, 483
\xeCJKDeclareCharClass	9, 390, 390	\XeTeXmathcode	243
\xeCJKDeclarePunctStyle			
	10, 1762, 1762, 1775, 1783, 1784, 1785, 1792, 1800		