

xeCJK 宏包

ctex.org

2013/06/11 3.2.3*

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	23
3	用户手册	2	5.6	字符输出规则	26
3.1	宏包选项	2	5.7	全角右标点后的断行	35
3.2	字体设置与选择	5	5.8	段末孤字处理	37
3.2.1	X _Y TeX 的字体名查找	7	5.9	增加 CJK 子分区	40
3.3	CJK 分区字体设置	8	5.10	标点处理	42
3.4	设置 CJK 字符范围	9	5.11	后备字体	52
3.5	标点符号的处理	9	5.12	CJK 字体族声明方式	53
3.5.1	设置特定标点符号的宽度和间距	9	5.13	字体切换	61
3.5.2	定义标点符号处理格式	10	5.14	数学字体设置	67
3.6	其它	11	5.15	抄录环境中的间距调整	68
4	已知问题和兼容性	12	5.16	xeCJK 其它选项	71
5	xeCJK 代码实现	12	5.17	xeCJK 初始化设置	72
5.1	运行环境检查	12	5.18	兼容性修补	74
5.2	内部工具	13	5.19	xeCJKfntef	79
5.3	功能开关	18	5.20	xeCJK-listings	86
			5.21	xeCJK.cfg	90
			版本历史	91	
			代码索引	92	

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` rev438.

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 = { $\langle \textit{quanjiao} \textit{banjiao} \textit{kaiming} \textit{hangmobanjiao} \textit{CCT} \textit{plain} ... \rangle$ }
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 = { $\langle \text{——} \cdots \rangle$ }
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 = { \footnote \footnotemark \nobreak }
NoBreakCS+	设置不能在全角右标点后断行的控制序列。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。
NoBreakCS-	
New: 2012-12-02	
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。
Mapping	Mapping = $\{ \langle fullwidth-stop full-stop han-trad han-simp ... \rangle \}$
New: 2013-06-07	xeCJK 提供了以上四个 TECKit 映射文件,可以在设置字体的时候通过 Mapping 选项来使用它们。其中 fullwidth-stop 用于将正常句号“。”转换成全角实心句号“。”,full-stop 的作用相反。han-trad 用于将简体中文转换成繁体中文,han-simp 的作用相反。需要注意的是,简繁互换都是简单机械的字字对译,不能做到完全准确,使用时要小心。例如简体的“发挥”和“头发”被转换成繁体的“發揮”和“頭髮”,显然后者应作“頭髮”。也可以根据实际需要,制作新的映射文件,请参考 TECKit 的文档。
\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+	\CJKfamily+ $\{ \langle family \rangle \}$
\CJKfamily-	\CJKfamily- $\{ \langle family \rangle \}$
Updated: 2012-10-27	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></i>switch] [<i></i>features] {}</code>
		声明新的 CJK 字体族 <i></i> family 并指定字体,并定义 <i></i> switch,在文档中可以使用它来切换 CJK 字体族。可以不必指定 <i></i> family,这时候 <i></i> family 将等于 <i></i> switch。事实上,\newCJKfontfamily 是 \setCJKfamilyfont 和 \CJKfamily 的合并。例如
		<code>\newCJKfontfamily[song]\songti{SimSun}</code>
		等价于
		<code>\setCJKfamilyfont{song}{SimSun}</code> <code>\newcommand*{\songti}{\CJKfamily{song}}</code>
<hr/> <hr/>	<code>\CJKfontspec</code>	<code>\CJKfontspec [<i></i>features] {}</code>
		在文档中随机定义新的 CJK 字体族,并马上使用它。
<hr/> <hr/>	<code>\defaultCJKfontfeatures</code> ☆	<code>\defaultCJKfontfeatures {}</code>
		全局设置 CJK 字体族的默认选项。例如,使用
		<code>\defaultCJKfontfeatures{Scale=0.962216}</code>
		可以将全部 CJK 字体缩小为 0.962216。xeCJK 宏包的初始化设置是
		<code>\defaultCJKfontfeatures{Script=CJK}</code>
<hr/> <hr/>	<code>\addCJKfontfeatures</code>	<code>\addCJKfontfeatures {}</code>
		临时增加当前使用的 CJK 字体的选项。例如,使用
		<code>\addCJKfontfeatures{Scale=1.1}</code>
		可以将文档中当前使用的 CJK 字体放大为 1.1。
<hr/> <hr/>	<code>\CJKrmdefault</code>	保存 \textrm 和 \rmfamily 所使用的 CJK 字体族,默认值是 rm。类似西文字体的 \rmdefault。
<hr/> <hr/>	<code>\CJKsfdefault</code>	保存 \textsf 和 \sffamily 所使用的 CJK 字体族,默认值是 sf。类似西文字体的 \sfdefault。
<hr/> <hr/>	<code>\CJKttdefault</code>	保存 \texttt 和 \ttfamily 所使用的 CJK 字体族,默认值是 tt。类似西文字体的 \ttdefault。
<hr/> <hr/>	<code>\CJKfamilydefault</code>	保存 \textnormal 和 \normalfont 所使用的 CJK 字体族。类似西文字体的 \familydefault。初始值是 \CJKrmdefault。如果没有在导言区中修改它,xeCJK 会在导言区结束的时候根据西文字体的情况自动更新 \CJKfamilydefault。因此,在导言区里使用
		<code>\renewcommand\familydefault{\sfdefault}</code>
		就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。
<hr/> <hr/>	<code>\setCJKmathfont</code> ★	<code>\setCJKmathfont [<i></i>features] {}</code>
		设置数学公式中的 CJK 字体族。如果使用了 CJKmath 选项,但是没有使用 \setCJKmathfont 设置数学公式中的 CJK 字体,那么将使用 \CJKfamilydefault 作为数学公式中的 CJK 字体。
<hr/> <hr/>	<code>\setCJKfallbackfamilyfont</code> ★	<code>\setCJKfallbackfamilyfont {family} [<i></i>features] {}</code>
		设置 CJK 字体族 <i></i> family 的备用字体。例如,使用
		<code>\setCJKmainfont{SimSun}</code> <code>\setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}</code>
		可以将 SimSun-ExtB 作为 SimSun 的备用字体。

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,Félkövéř dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrto Ítálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın Ítalik,
Krepko poševno,nghiêng đăm,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντονα,Bold,Negrita,
Lihavoitu,Gras,Félkövéř,Grassetto,Vet,Halvfet,Pogrubiona,Negrto,
Полужирный,Fet,Kalın,Krepko,đăm,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

`\xeCJKDeclarePunctStyle` ★ `\xeCJKDeclarePunctStyle {<style>} {<options>}`

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。

`\xeCJKEditPunctStyle` ★ `\xeCJKEditPunctStyle {<style>} {<options>}`

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

\xeCJKVerbAddon
\xeCJKOffVerbAddon
Updated: 2013-06-04

`\xeCJKVerbAddon`
`\xeCJKOffVerbAddon`
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。 \xeCJKOffVerbAddon 用于在使用 \xeCJKVerbAddon 的环境中局部取消它的作用。由于 listings 宏包有自己的代码对齐机制,所以 \xeCJKVerbAddon 在由 listings 定义的代码环境中无效。

\xeCJKnobreak	……汉字。 \xeCJKnobreak\footnote{脚注}
New: 2012-12-03	\xeCJKnobreak 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

X_gTeX 在配置文件 `unicode-letters.tex` 中将所有 CJK 表意文字的 `\catcode` 设置为 11。因此汉字可以直接用作控制序列的名字,但是当汉字出现在控制序列后面的时候,要用空格分隔开,否则就会出现“! Undefined control sequence.”的错误。

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 X_gTeX 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。

xeCJK 进行了一些处理,使得在使用 X_gTeX 时 `listings` 宏包可以支持 Unicode,因此在 `listings` 定义的代码环境中可以直接使用中文,不再需要通过 `escapechar`。

新版本(3.x)的 xeCJK 完全使用 L_AT_EX3 的语法来编写。L_AT_EX3 放弃了 `\outer` 宏的概念,因此相关工具在遇到 `\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 `cprotect` 宏包提供的 `\cprotect`。它的定义是

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

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

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

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

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

5 xeCJK 代码实现

```
1 <*package>
2 <@@=xeCJK>
```

5.1 运行环境检查

xeCJK 必须使用 X_gTeX 引擎的支持。

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

应该使用较新版本的 `expl3` 宏包。

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

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

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

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:nnx { 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:nnx { xeCJK } }

```

```

41 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }

```

```

42 \cs_new_protected_nopar:Npn \__xeCJK_info:nx { \msg_info:nnx { xeCJK } }

```

```

43 \cs_new_protected_nopar:Npn \__xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }

```

(End definition for __xeCJK_msg_new:nn and others.)

\xeCJK_allow_break:

\xeCJK_no_break:

```

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

```

```

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 Xe_{La}TeX 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 Xe_{La}TeX 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。
HalfRight 304 \xeCJK_new_class:n { HalfLeft }
NormalSpace 305 \xeCJK_new_class:n { HalfRight }
IVS 306 \xeCJK_new_class:n { NormalSpace }
307 \xeCJK_new_class:n { IVS }
(End definition for HalfLeft and others.)

\c_xeCJK_HalfLeft_chars_clist 西文半角左/右标点和前后原始间距的字符类。
\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 }
310 \clist_const:Nn \c_xeCJK_HalfRight_chars_clist
311 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
312 \clist_const:Nn \c_xeCJK_NormalSpace_chars_clist { "2F }
(End definition for \c_xeCJK_HalfLeft_chars_clist, \c_xeCJK_HalfRight_chars_clist, and \c_xeCJK_NormalSpace_chars_clist.)

以下对全角标点符号的归类来源于 XeTeX 的脚本 [unicode-char-prep.pl](#) 和 Unicode 数据库⁷。

\c_xeCJK_OP_chars_clist	Open Punctuation (OP)	U+2018	'	U+201C	"	U+2329	⟨	U+3008	⎵	U+300A	《	U+300C	⎴	U+300E	『
		U+3010	【	U+3014	〔	U+3016	⎶	U+3018	⎷	U+301A	⎮	U+301D	⎵	U+FE17	⎶
		U+FE35	⎵	U+FE37	⎶	U+FE39	⎵	U+FE3B	⎶	U+FE3D	⎷	U+FE3F	⎵	U+FE41	⎶
		U+FE43	⎷	U+FE47	⎶	U+FE59	⎵	U+FE5B	⎶	U+FE5D	⎷	U+FF08	(U+FF3B	[
		U+FF5B	{	U+FF5F	⎵	U+FF62	⎴								

以下代码的第一行是中西文共用的左引号。
313 \clist_const:Nn \c_xeCJK_OP_chars_clist
314 {
315 "2018 , "201C ,
316 "2329 , "3008 , "300A , "300C , "300E , "3010 , "3014 , "3016 , "3018 , "301A ,
317 "301D , "FE17 , "FE35 , "FE37 , "FE39 , "FE3B , "FE3D , "FE3F , "FE41 , "FE43 ,
318 "FE47 , "FE59 , "FE5B , "FE5D , "FF08 , "FF3B , "FF5B , "FF5F , "FF62
319 }

⁵<http://www.unicode.org/reports/tr37/>
⁶<http://tug.org/pipermail/xetex/2013-March/024118.html>
⁷<http://www.unicode.org/reports/tr14/>

(End definition for \c__xeCJK_OP_chars_clist.)

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

	U+FE6A		%		U+FF05		%		U+FFE0		¢	
--	--------	--	---	--	--------	--	---	--	--------	--	---	--

```
320 \clist_const:Nn \c__xeCJK_PO_chars_clist { "FE6A , "FF05 , "FFE0 }
```

(End definition for \c__xeCJK_PO_chars_clist.)

\c__xeCJK_FullLeft_chars_clist 以上两类标点符号出现在文字的左边,不应出现在行尾位置。

```
321 \clist_const:Nx \c__xeCJK_FullLeft_chars_clist
322 {
323   \c__xeCJK_OP_chars_clist ,
324   \c__xeCJK_PO_chars_clist
325 }
```

(End definition for \c__xeCJK_FullLeft_chars_clist.)

\c__xeCJK_CL_chars_clist Close Punctuation (CL)

U+00B7	·	U+2019	’	U+201D	”	U+2014	—	U+2015	–	U+2025	..	U+2026	...
U+2027	·	U+2500	—	U+232A	}	U+3001	、	U+3002	。	U+3009	}	U+300B	》
U+300D	』	U+300F	』	U+3011	】	U+3015	〕	U+3017	〕	U+3019	〕	U+301B	】
U+301E	”	U+301F	”	U+FE11	、	U+FE12	。	U+FE18	⎵	U+FE36	⌋	U+FE38	⌋
U+FE3A	⌋	U+FE3C	⌋	U+FE3E	≈	U+FE40	⌋	U+FE42	┌	U+FE44	┌	U+FE48	┌
U+FE50	,	U+FE52	,	U+FE5A)	U+FE5C	}	U+FE5E)	U+FF09)	U+FF0C	,
U+FF0E	.	U+FF3D]	U+FF5D	}	U+FF60)	U+FF61	。	U+FF63	└	U+FF64	、

以下代码的第一行是中西文共用的一些标点符号。

```
326 \clist_const:Nn \c__xeCJK_CL_chars_clist
327 {
328   "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2027 , "2500 ,
329   "232A , "3001 , "3002 , "3009 , "300B , "300D , "300F , "3011 , "3015 , "3017 ,
330   "3019 , "301B , "301E , "301F , "FE11 , "FE12 , "FE18 , "FE36 , "FE38 , "FE3A ,
331   "FE3C , "FE3E , "FE40 , "FE42 , "FE44 , "FE48 , "FE50 , "FE52 , "FE5A , "FE5C ,
332   "FE5E , "FF09 , "FF0C , "FF0E , "FF3D , "FF5D , "FF60 , "FF61 , "FF63 , "FF64
333 }
```

(End definition for \c__xeCJK_CL_chars_clist.)

\c__xeCJK_NS_chars_clist Nonstarter (NS)

U+3005	々	U+301C	ゝ	U+303B	々	U+303C	☒	U+309B	ˆ	U+309C	ˆ	U+309D	ˆ
U+309E	々	U+30A0	=	U+30FB	・	U+30FD	、	U+30FE	々	U+A015	□	U+FE54	;
U+FE55	:	U+FF1A	:	U+FF1B	;	U+FF65	・	U+FF9E	ˆ	U+FF9F	ˆ		

```
334 \clist_const:Nn \c__xeCJK_NS_chars_clist
335 {
336   "3005 , "301C , "303B , "303C , "309B , "309C , "309D , "309E , "30A0 , "30FB ,
337   "30FD , "30FE , "A015 , "FE54 , "FE55 , "FF1A , "FF1B , "FF65 , "FF9E , "FF9F
338 }
```

(End definition for \c__xeCJK_NS_chars_clist.)

\c__xeCJK_EX_chars_clist Exclamation/Interrogation (EX)

	U+FE15		!		U+FE16		?		U+FE56		?		U+FE57		!		U+FF01		!		U+FF1F		?	
--	--------	--	---	--	--------	--	---	--	--------	--	---	--	--------	--	---	--	--------	--	---	--	--------	--	---	--

```
339 \clist_const:Nn \c__xeCJK_EX_chars_clist
340 { "FE15 , "FE16 , "FE56 , "FE57 , "FF01 , "FF1F }
```

(End definition for \c__xeCJK_EX_chars_clist.)

\c__xeCJK_IS_chars_clist Infix Numeric Separator (IS)

	U+FE10		’		U+FE13		:		U+FE14		;	
--	--------	--	---	--	--------	--	---	--	--------	--	---	--

```
341 \clist_const:Nn \c__xeCJK_IS_chars_clist { "FE10 , "FE13 , "FE14 }
```

(End definition for \c__xeCJK_IS_chars_clist.)

\c__xeCJK_CJ_chars_clist Conditional Japanese Starter (CJ)

U+3041	あ	U+3043	い	U+3045	う	U+3047	え	U+3049	お	U+3063	っ	U+3083	や
U+3085	ゆ	U+3087	よ	U+308E	わ	U+3095	か	U+3096	け	U+30A1	ア	U+30A3	イ
U+30A5	ウ	U+30A7	エ	U+30A9	オ	U+30C3	ツ	U+30E3	ヤ	U+30E5	ユ	U+30E7	ヨ
U+30EE	ワ	U+30F5	カ	U+30F6	ケ	U+30FC	一	U+31F0	ク	U+31F1	シ	U+31F2	ス
U+31F3	ト	U+31F4	ヌ	U+31F5	ハ	U+31F6	ヒ	U+31F7	フ	U+31F8	ヘ	U+31F9	ホ
U+31FA	ム	U+31FB	ラ	U+31FC	リ	U+31FD	ル	U+31FE	レ	U+31FF	ロ	U+FF67	ア
U+FF68	イ	U+FF69	ウ	U+FF6A	エ	U+FF6B	オ	U+FF6C	カ	U+FF6D	キ	U+FF6E	コ
U+FF6F	ク	U+FF70	ケ										

342 \clist_const:Nn \c__xeCJK_CJ_chars_clist

343 {

344 "3041 , "3043 , "3045 , "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E ,
345 "3095 , "3096 , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
346 "30E7 , "30EE , "30F5 , "30F6 , "30FC , "31F0 , "31F1 , "31F2 , "31F3 , "31F4 ,
347 "31F5 , "31F6 , "31F7 , "31F8 , "31F9 , "31FA , "31FB , "31FC , "31FD , "31FE ,
348 "31FF , "FF67 , "FF68 , "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F ,
349 "FF70

350 }

(End definition for \c__xeCJK_CJ_chars_clist.)

\c__xeCJK_PR_chars_clist Prefix Numeric (PR)

U+20A9 | ￥ | U+FE69 | \$ | U+FF04 | \$ | U+FFE1 | £ | U+FFE5 | ¥ | U+FFE6 | ₩ |

351 \clist_const:Nn \c__xeCJK_PR_chars_clist

352 { "20A9 , "FE69 , "FF04 , "FFE1 , "FFE5 , "FFE6 }

(End definition for \c__xeCJK_PR_chars_clist.)

\c__xeCJK_FullRight_chars_clist 以上六类标点符号出现在文字的右边,不应出现在行首位置。

353 \clist_const:Nx \c__xeCJK_FullRight_chars_clist

354 {

355 \c__xeCJK_CL_chars_clist ,
356 \c__xeCJK_NS_chars_clist ,
357 \c__xeCJK_EX_chars_clist ,
358 \c__xeCJK_IS_chars_clist ,
359 \c__xeCJK_CJ_chars_clist ,
360 \c__xeCJK_PR_chars_clist

361 }

(End definition for \c__xeCJK_FullRight_chars_clist.)

\c__xeCJK_CJK_chars_clist CJK 字符类,包括文字和标点符号。

362 \clist_const:Nn \c__xeCJK_CJK_chars_clist

363 {

- Hangul Jamo (谚文字母)

364 "1100 -> "11FF ,

- CJK Radicals Supplement (中日韩部首补充)

365 "2E80 -> "2EFF ,

- Kangxi Radicals (康熙字典部首)

366 "2F00 -> "2FDF ,

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

367 "2FF0 -> "2FFF ,

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

368 "3000 -> "303F ,

- Hiragana (日文平假名)

369 "3040 -> "309F ,

- Katakana (日文片假名)
370 "30A0 -> "30FF ,
- Bopomofo (注音字母)
371 "3100 -> "312F ,
- Hangul Compatibility Jamo (谚文兼容字母)
372 "3130 -> "318F ,
- Kanbun (象形字注释标志)
373 "3190 -> "319F ,
- Bopomofo Extended (注音字母扩展)
374 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)
375 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)
376 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
377 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
378 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
379 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
380 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
381 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
382 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
383 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
384 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
385 "AC00 -> "D7AF ,
- Hangul Jamo Extended-B (谚文扩展 B)
386 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
387 "F900 -> "FAFF ,
- Vertical Forms (竖排形式)
388 "FE10 -> "FE1F ,

- CJK Compatibility Forms (中日韩兼容形式)

```
389      "FE30 -> "FE4F ,
```

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

```
390      "FF00 -> "FFEF ,
```

- Kana Supplement (日文假名增补)

```
391      "1B000 -> "1B0FF ,
```

- Enclosed Ideographic Supplement (带圈表意文字增补)

```
392      "1F200 -> "1F2FF ,
```

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

```
393      "20000 -> "2A6DF ,
```

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

```
394      "2A700 -> "2B73F ,
```

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

```
395      "2B740 -> "2B81F ,
```

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

```
396      "2F800 -> "2FA1F
```

```
397    }
```

(End definition for \c__xeCJK_CJK_chars_clist.)

\c__xeCJK_IVS_chars_clist 包括日文假名浊点和异体字选择符。

```
398 \clist_const:Nn \c__xeCJK_IVS_chars_clist
```

```
399 {
```

- 日文假名浊点

```
400      "3099 -> "309A ,
```

- Variation Selectors (异体字选择符)

```
401      "FE00 -> "FE0F ,
```

- Variation Selectors Supplement (异体字选择符增补)

```
402      "E0100 -> "E01EF
```

```
403    }
```

(End definition for \c__xeCJK_IVS_chars_clist.)

5.5 字符类别处理

\g__xeCJK_base_class_seq

\g__xeCJK_non_CJK_class_seq

\g__xeCJK_CJK_class_seq

```
404 \seq_new:N \g__xeCJK_base_class_seq
```

```
405 \seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq
```

```
406 \seq_new:N \g__xeCJK_non_CJK_class_seq
```

```
407 \seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq
```

```
408 { Default , HalfLeft , HalfRight , NormalSpace , Boundary }
```

```
409 \seq_new:N \g__xeCJK_CJK_class_seq
```

```
410 \cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
```

```
411 {
```

```
412   \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
```

```
413   \tl_const:cn { \__xeCJK_CJK_class_tl:c { \__xeCJK_class_csname:n {#1} } } {#1}
```

```
414 }
```

```
415 \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。

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

\xeCJKDeclareCharClass

```
417 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }  
418 {  
419   \xeCJK_declare_char_class:nx {#2} {#3}  
420   \IfBooleanT {#1} { \xeCJKResetPunctClass }  
421 }
```

(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,支持类似 "1100 -> 11FF 起止范围的使用方式。

__xeCJK_set_char_class_aux:Nnw

```
422 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2  
423 {  
424   \clist_set:Nn \l__xeCJK_tmpa_clist {#2}  
425   \clist_gconcat:ccN  
426     { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmpa_clist  
427   \clist_map_inline:Nn \l__xeCJK_tmpa_clist  
428     {  
429       \str_if_eq:nnF {##1} { -> }  
430       {  
431         \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}  
432         { \xeCJK_class_num:n {#1} }  
433       }  
434     }  
435   \xeCJK_set_char_class:nnn { "3099 } { "309A } { \xeCJK_class_num:n { IVS } }  
436 }  
437 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw  
438 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }  
439 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }  
440 \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_check_num_range:nnNN

```
441 \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4  
442 {  
443   \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }  
444   {  
445     \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }  
446     \int_set_eq:NN #3 #4  
447   }  
448   {  
449     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }  
450     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }  
451   }  
452 }
```

(End definition for __xeCJK_check_num_range:nnNN.)

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

```
453 \cs_new_protected_nopar:Npn \xeCJK_set_char_class:nnn #1#2#3  
454 {  
455   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int  
456   \int_set:Nn \l__xeCJK_tmpc_int {#3}  
457   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }  
458   {  
459     \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int  
460     \int_incr:N \l__xeCJK_tmpa_int  
461   }  
462 }
```

(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。

```

463 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
464 {
465   \int_set:Nn \l__xeCJK_tmpa_int { \xeCJK_class_num:n {#2} }
466   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
467     { \XeTeXcharclass ##1 = \l__xeCJK_tmpa_int }
468 }
(End definition for \__xeCJK_set_char_class_eq:nn.)

```

`\normalspacedchars` 声明前后不加间距的字符。

```

469 \NewDocumentCommand \normalspacedchars { m }
470 {
471   \tl_map_inline:nn {#1}
472     { \XeTeXcharclass '##1 = \xeCJK_class_num:n { NormalSpace } }
473 }
(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。

```

474 \NewDocumentCommand \xeCJKResetPunctClass { }
475 {
476   \xeCJK_declare_char_class:nV { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
477   \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
478   \xeCJK_declare_char_class:nV { FullLeft } \c__xeCJK_FullLeft_chars_clist
479   \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
480 }
(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。

```

481 \NewDocumentCommand \xeCJKResetCharClass { }
482 {
483   \xeCJK_declare_char_class:nV { CJK } \c__xeCJK_CJK_chars_clist
484   \xeCJK_declare_char_class:nV { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
485   \xeCJK_declare_char_class:nV { IVS } \c__xeCJK_IVS_chars_clist
486   \xeCJKResetPunctClass
487 }
(End definition for \xeCJKResetCharClass. This function is documented on page 9.)

```

设置字符类别。

```

488 \xeCJKResetCharClass

```

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

```

489 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
490 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }
491 \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。

```

492 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2
493 { \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。

```

494 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2
495 { \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。

```

496 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
497 {
498   \xeCJK_inter_class_toks:nnx {#1} {#2}
499   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
500 }
501 \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。

```

502 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
503 {
504   \xeCJK_inter_class_toks:nnx {#1} {#2}
505   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
506 }
507 \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。

```

508 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
509 {
510   \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
511   \tl_if_empty:NF \l__xeCJK_tmpa_tl
512   { \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_tl } }
513 }
(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。

```

514 \cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
515 {
516   \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
517   \tl_if_empty:NF \l__xeCJK_tmpa_tl
518   {
519     \tl_replace_all:Nnn \l__xeCJK_tmpa_tl {#3} {#4}
520     \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_tl }
521   }
522 }
(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。

```

523 \cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
524 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
525 \cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
526 { \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:

527 \cs_new_eq:NN \xeCJK_class_group_begin: \c_group_begin_token

528 \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。

529 \AtEndOfPackage

530 {

531 \seq_map_inline:Nn \g__xeCJK_class_seq

532 {

533 \str_if_eq:nnTF {#1} { IVS }

534 { \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } { CJK } }

535 {

536 \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } {#1}

537 \str_if_eq:nnF {#1} { CJK }

538 { \xeCJK_copy_inter_class_toks:nnnn {#1} { IVS } {#1} { CJK } }

539 }

540 }

541 }

542 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }

543 {

544 \xeCJK_inter_class_toks:nnn {#1} { CJK }

545 {

546 \xeCJK_class_group_begin:

547 \xeCJK_select_font:

548 \xeCJK_clear_inter_class_toks:nn {#1} { CJK }

549 \xeCJK_clear_Boundary_and_CJK_toks:

550 \CJKsymbol

551 }

552 \xeCJK_inter_class_toks:nnn { CJK } {#1} { \xeCJK_class_group_end: }

553 }

554 \clist_map_inline:nn { Default , HalfLeft , NormalSpace }

555 {

556 \xeCJK_inter_class_toks:nnn { Boundary } {#1}

557 {

558 \bool_if:nTF

559 {

560 \l__xeCJK_xecglue_bool &&

561 \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&

562 \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl

563 }

564 {

565 \tex_unskip:D

566 \bool_if:nTF

567 {

568 \xeCJK_if_last_node_p:n { CJK } ||

569 \xeCJK_if_last_node_p:n { CJK-space }

570 }

571 { \CJKecglue } { ~ }

572 }

573 {

574 \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }

575 { \xeCJK_if_last_node:nT { CJK-space } { __xeCJK_space_or_xecglue: } }

576 }

577 }

578 \str_if_eq:nnF {#1} { NormalSpace }

579 { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }

580 }

581 \clist_map_inline:nn { Default , HalfRight , NormalSpace }

582 {

583 \xeCJK_inter_class_toks:nnn {#1} { Boundary }

584 {

```

585 \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
586 \peek_meaning_remove:NTF \tex_italiccorrection:D
587 { \tex_italiccorrection:D { \xeCJK_make_node:n { default } } }
588 {
589 \token_if_space:NTF \l_peek_token
590 { { \xeCJK_make_node:n { default-space } } }
591 { { \xeCJK_make_node:n { default } } }
592 }
593 }
594 \str_if_eq:nnF {#1} { NormalSpace }
595 { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
596 }
597 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
598 {
599 \xeCJK_check_for_glue:
600 \xeCJK_class_group_begin:
601 \xeCJK_clear_Boundary_and_CJK_toks:
602 \xeCJK_select_font:
603 \CJKsymbol
604 }

\xeCJK_check_for_glue:

605 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
606 {
607 \bool_if:nTF
608 { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
609 { \CJKglue }
610 {
611 \bool_if:nTF
612 {
613 \xeCJK_if_last_node_p:n { default } ||
614 \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
615 }
616 { \CJKecglue }
617 {
618 \bool_if:nT
619 {
620 \l__xeCJK_xecglue_bool &&
621 \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
622 ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
623 \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_ecglue_skip )
624 }
625 {
626 \tex_unskip:D
627 \bool_if:nTF
628 {
629 \xeCJK_if_last_node_p:n { default-space } ||
630 \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
631 \xeCJK_if_last_node_p:n { default }
632 }
633 { \CJKecglue }
634 {
635 \bool_if:nTF
636 {
637 \xeCJK_if_last_node_p:n { CJK } ||
638 \xeCJK_if_last_node_p:n { CJK-space }
639 }
640 { \bool_if:NTF \l__xeCJK_reserve_space_bool { ~ } { \CJKglue } }
641 { ~ }
642 }
643 }
644 }

```

```

645     }
646 }
(End definition for \xeCJK_check_for_glue:.)

\xeCJK_if_last_node:p:n
\xeCJK_if_last_node:nTF
647 \prg_new_conditional:Nppn \xeCJK_if_last_node:n #1 { p , T , F , TF }
648 {
649     \if_dim:w \use:c { c__xeCJK_#1_node_dim } = \tex_lastkern:D
650     \prg_return_true: \else: \prg_return_false: \fi:
651 }
(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

```

652 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
653 {
654     \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
655     { \dim_gset:cn } { \dim_const:cn }
656     { c__xeCJK_#1_node_dim } {#2}
657 }
658 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
659 {
660     \tex_kern:D - \use:c { c__xeCJK_#1_node_dim }
661     \tex_kern:D \use:c { c__xeCJK_#1_node_dim }
662 }
663 \xeCJK_def_node:nn { CJK } { 11 sp }
664 \xeCJK_def_node:nn { CJK-space } { 12 sp }
665 \xeCJK_def_node:nn { default } { 13 sp }
666 \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。

```

667 \keys_define:nn { xeCJK / options }
668 {
669     CJKglue .code:n =
670     {
671         \cs_set_protected_nopar:Npn \CJKglue {#1}
672         \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
673     }
674 }
675 \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。

xCJKecglue

```

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

```

```

695     {
696       \bool_set_true:N \l__xeCJK_xecglue_bool
697       \cs_set_protected_nopar:Npn \CJKecglue {#1}
698       \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
699       \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
700     } ,
701     xCJKecglue .default:n = { true }
702   }
703   \skip_new:N \l__xeCJK_ecglue_skip
704   \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。

```

705   \keys_define:nn { xeCJK / options }
706   {
707     CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
708     space .meta:n = { CJKspace = true } ,
709     nospace .meta:n = { CJKspace = false }
710   }

```

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

```

711   \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
712   {
713     \xeCJK_class_group_end:
714     { \xeCJK_make_node:n { CJK } }
715     \xeCJK_ignore_spaces:w
716   }

```

\xeCJK_ignore_spaces:w

```

717   \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
718   {
719     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
720     {
721       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
722       { \__xeCJK_space_or_xecglue: } { \CJKecglue }
723     }
724     {
725       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
726       {
727         \tex_unkern:D \tex_unkern:D
728         { \xeCJK_make_node:n { CJK-space } }
729         \bool_if:nT
730         {
731           \token_if_macro_p:N \l_peek_token ||
732           ( \l__xeCJK_reserve_space_bool &&
733             \token_if_other_p:N \l_peek_token )
734         }
735         { \__xeCJK_space_or_xecglue: }
736       }
737       { \cs_if_exist:NF \l_peek_token \exp_not:N }
738     }
739   }

```

(End definition for \xeCJK_ignore_spaces:w.)

```

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

```

\xeCJK_CJK_and_CJK:N

```

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

```

(End definition for \xeCJK_CJK_and_CJK:N.)

```

742   \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
743   { \xeCJK_FullLeft_and_CJK: \CJKsymbol }
744   \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
745   { \xeCJK_FullRight_and_CJK: \CJKsymbol }

```

```

746 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
747 {
748   \clist_map_inline:nn { FullLeft , FullRight }
749   {
750     \xeCJK_inter_class_toks:nxx {#1} {##1}
751     { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
752     \xeCJK_inter_class_toks:nnc {##1} {#1} { xeCJK_##1_and_Default: }
753   }
754 }

755 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
756 { \xeCJK_Boundary_and_FullLeft:N }
757 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
758 { \xeCJK_Boundary_and_FullRight:N }

\xecjk_FullRight_and_Boundary:

759 \xeCJK_app_inter_class_toks:nnn { FullLeft } { Boundary } { \tex_ignorespaces:D }
760 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
761 { \xeCJK_FullRight_and_Boundary: }
(End definition for \xeCJK_FullRight_and_Boundary:.)

\xecjk_FullRight_and_Boundary:

762 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Boundary:
763 { \xeCJK_FullRight_and_Default: \tex_ignorespaces:D }
(End definition for \xeCJK_FullRight_and_Boundary:.)

764 \clist_map_inline:nn { CJK , FullLeft , FullRight }
765 {
766   \clist_map_inline:nn { FullLeft , FullRight }
767   { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
768 }

\xecjk_punct_rule:NN 用于抹去标点符号的左/右空白。

769 \cs_new_protected_nopar:Npn \__xeCJK_punct_rule:NN #1#2
770 {
771   \tex_vrule:D
772   width - \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
773   depth \c_zero_dim
774   height \c_zero_dim \scan_stop:
775 }
(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。

776 \cs_new_protected_nopar:Npn \__xeCJK_punct_glue:NN #1#2
777 {
778   \__xeCJK_punct_hskip:n
779   {
780     \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
781     minus \dim_eval:n { ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} ) / \c_two }
782   }
783 }
784 \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。

785 \cs_new_protected_nopar:Npn \__xeCJK_punct_kern:NN #1#2
786 { \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。

787 \tl_new:N \g__xeCJK_last_punct_tl
(End definition for \g__xeCJK_last_punct_tl.)

```

\xeCJK_FullLeft_and_CJK:

```
788 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_CJK:
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_no_break:
794     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
795   }
796   { \xeCJK_no_break: }
797 }
(End definition for \xeCJK_FullLeft_and_CJK:.)
```

\xeCJK_FullLeft_and_Default:

```
798 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_Default:
799 {
800   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
801   {
802     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
803     \xeCJK_class_group_end: \xeCJK_no_break:
804     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
805   }
806   { \xeCJK_class_group_end: \xeCJK_no_break: \__xeCJK_zero_glue: }
807 }
808 \cs_new_protected_nopar:Npn \__xeCJK_zero_glue:
809 { \skip_horizontal:N \c_zero_skip }
(End definition for \xeCJK_FullLeft_and_Default:.)
```

\xeCJK_FullRight_and_CJK:

```
810 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_CJK:
811 {
812   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
813   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
814   \CJKglue
815 }
(End definition for \xeCJK_FullRight_and_CJK:.)
```

\xeCJK_FullRight_and_Default:

```
816 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Default:
817 {
818   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
819   \xeCJK_class_group_end:
820   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
821 }
(End definition for \xeCJK_FullRight_and_Default:.)
```

\xeCJK_Default_and_FullLeft:nN

```
822 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullLeft:nN #1#2
823 {
824   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#2}
825   \__xeCJK_Default_and_FullLeft_glue:N {#2}
826   \xeCJK_class_group_begin:
827   \xeCJK_select_font:
828   \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
829   \xeCJK_clear_Boundary_and_CJK_toks:
830   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
831   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2}
832   \CJKpunctsymbol {#2}
833 }
834 \cs_new_protected_nopar:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
835 { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_Default_and_FullLeft:nN.)
```


\xeCJK_CJK_and_FullLeft:N

```
836 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
837 {
838   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
839   \__xeCJK_CJK_and_FullLeft_glue:N {#1}
840   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
841   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
842   \CJKpunctsymbol {#1}
843 }
844 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
845 { \CJKglue \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_CJK_and_FullLeft:N.)
```

\xeCJK_Boundary_and_FullLeft:N

```
846 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullLeft:N #1
847 {
848   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
849   \__xeCJK_Boundary_and_FullLeft_glue:N {#1}
850   \xeCJK_class_group_begin:
851   \xeCJK_select_font:
852   \xeCJK_clear_Boundary_and_CJK_toks:
853   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
854   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
855   \CJKpunctsymbol {#1}
856 }
857 \cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
858 {
859   \int_case:nnn { \etex_lastnodetype:D }
860   {
861     { \c_one } { \__xeCJK_zero_glue: }
862     { \c_minus_one } { }
863     { \c_eleven } { }
864     { \c_thirteen } { }
865   }
866   { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
867 }
(End definition for \xeCJK_Boundary_and_FullLeft:N.)
```

\xeCJK_Default_and_FullRight:nN

```
868 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullRight:nN #1#2
869 {
870   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#2}
871   \__xeCJK_Default_and_FullRight_glue:N {#2}
872   \xeCJK_class_group_begin:
873   \xeCJK_select_font:
874   \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
875   \xeCJK_clear_Boundary_and_CJK_toks:
876   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
877   \__xeCJK_punct_if_middle:NT {#2}
878   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2} }
879   \xeCJK_FullRight_symbol:N {#2}
880 }
(End definition for \xeCJK_Default_and_FullRight:nN.)
```

\xeCJK_Boundary_and_FullRight:N

```
881 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullRight:N #1
882 {
883   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
884   \__xeCJK_Default_and_FullRight_glue:N {#1}
885   \xeCJK_class_group_begin:
886   \xeCJK_select_font:
887   \xeCJK_clear_Boundary_and_CJK_toks:
```

```

888 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
889 \__xeCJK_punct_if_middle:NT {#1}
890 { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
891 \xeCJK_FullRight_symbol:N {#1}
892 }

```

(End definition for \xeCJK_Boundary_and_FullRight:N.)

\xeCJK_CJK_and_FullRight:N

```

893 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
894 {
895   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
896   \__xeCJK_CJK_and_FullRight_glue:N {#1}
897   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
898   \__xeCJK_punct_if_middle:NT {#1}
899   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
900   \xeCJK_FullRight_symbol:N {#1}
901 }

```

(End definition for \xeCJK_CJK_and_FullRight:N.)

__xeCJK_CJK_and_FullRight_glue:N

__xeCJK_Default_and_FullRight_glue:N

```

902 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
903 {
904   \__xeCJK_punct_if_long:NTF {#1}
905   { \CJKglue }
906   {
907     \__xeCJK_punct_if_middle:NTF {#1}
908     { \xeCJK_no_break: \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1} }
909     { \xeCJK_no_break: \__xeCJK_zero_glue: }
910   }
911 }

```

912 \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

```

913 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
914 {
915   \xeCJK_no_break:
916   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
917   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
918   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
919   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
920   \CJKpunctsymbol {#1}
921 }

```

(End definition for \xeCJK_FullLeft_and_FullLeft:N.)

\xeCJK_FullLeft_and_FullRight:N

```

922 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
923 {
924   \xeCJK_no_break:
925   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
926   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
927   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
928   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
929   \xeCJK_no_break:
930   \xeCJK_FullRight_symbol:N {#1}
931 }

```

(End definition for \xeCJK_FullLeft_and_FullRight:N.)

\xeCJK_FullRight_and_FullLeft:N

```

932 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
933 {
934   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}

```

```

935 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
936 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
937 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
938 \__xeCJK_punct_break:
939 \CJKpunctsymbol {#1}
940 }
(End definition for \xeCJK_FullRight_and_FullLeft:N.)

```

\xeCJK_FullRight_and_FullRight:N

```

941 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
942 {
943 \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
944 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
945 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
946 \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
947 \xeCJK_no_break:
948 \xeCJK_FullRight_symbol:N {#1}
949 }
(End definition for \xeCJK_FullRight_and_FullRight:N.)

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```

950 \keys_define:nn { xecjk / options }
951 {
952   CheckFullRight .choice: ,
953   CheckFullRight / true .code:n =
954   {
955     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
956     {
957       \cs_set_eq:NN \__xeCJK_save_FullRight_check: \xeCJK_FullRight_and_Boundary:
958       \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N \xeCJK_FullRight_symbol:N
959       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
960       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \xeCJK_check_FullRight_symbol:Nw
961     }
962   } ,
963   CheckFullRight / false .code:n =
964   {
965     \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
966     {
967       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \__xeCJK_save_FullRight_check:
968       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \__xeCJK_save_FullRight_symbol:N
969     }
970   } ,
971   CheckFullRight .default:n = { true }
972 }
(End definition for CheckFullRight. This function is documented on page 5.)

```

\xeCJK_FullRight_symbol:N

```

973 \cs_new_nopar:Npn \xeCJK_FullRight_symbol:N { \CJKpunctsymbol }
(End definition for \xeCJK_FullRight_symbol:N.)

```

\xeCJK_check_FullRight:

```

974 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight:
975 {
976 \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
977 \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
978 \tl_case:Non \l_peek_token { \l__xeCJK_no_break_cs_case_tl } { }
979 \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
980 \xeCJK_class_group_end:
981 }
982 \cs_generate_variant:Nn \tl_case:Nnn { No }

```

(End definition for \xeCJK_check_FullRight:.)

\xeCJK_check_FullRight_symbol:Nw

```
983 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight_symbol:Nw #1
984 { \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:nNNnn

```
985 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
986 {
987   \tl_new:N #2
988   \seq_new:N #3
989   \keys_define:nn { xeCJK / options }
990   {
991     #1 .code:n =
992     {
993       \seq_set_split:Nnn #3 { } {##1}
994       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
995     } ,
996     #1+ .code:n =
997     {
998       \tl_map_inline:nn {##1}
999       { \seq_if_in:NnF #3 {###1} { \seq_put_right:Nn #3 {###1} } }
1000       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1001     } ,
1002     #1- .code:n =
1003     {
1004       \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {###1} }
1005       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1006     }
1007   }
1008 }
1009 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
1010 {
1011   \tl_clear:N #1
1012   \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
1013   #4
1014 }
(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。

```
1015 \xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
1016 \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。

```
1017 \NewDocumentCommand \xeCJKnobreak { }
1018 {
1019   \int_zero:N \l__xeCJK_tmpa_int
1020   \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
1021   {
1022     \int_if_even:nTF \l__xeCJK_tmpa_int
1023     {
1024       \int_incr:N \l__xeCJK_tmpa_int
1025       \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
1026     }
1027     { \skip_add:Nn \l__xeCJK_tmp_skip \tex_lastskip:D }
1028     \tex_unskip:D
1029   }
1030   \xeCJK_no_break:
1031   \int_if_even:nF \l__xeCJK_tmpa_int
1032   { \skip_horizontal:N \l__xeCJK_tmp_skip }
1033 }
```

(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。

```
1034 \keys_define:nn { xeCJK / options }
1035 {
1036   CheckSingle .choice: ,
1037   CheckSingle / true .code:n =
1038   {
1039     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1040     {
1041       \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
1042       \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1043     }
1044   } ,
1045   CheckSingle / false .code:n =
1046   {
1047     \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1048     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
1049   } ,
1050   CheckSingle .default:n = { true } ,
1051   CJKchecksingle .meta:n = { CheckSingle = true }
1052 }
```

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

\xeCJK_check_single:Nw

```
1053 \cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
1054 {
1055   \peek_catcode:NTF \c_catcode_letter_token
1056   { \xeCJK_check_single:NNw #1 }
1057   {
1058     \token_if_other:NNTF \l_peek_token
1059     { \xeCJK_check_single:NNw #1 }
1060     {
1061       \bool_if:nTF
1062       {
1063         \xeCJK_if_blank_x_p:n { \token_get_arg_spec:N \l_peek_token } &&
1064         \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
1065         ( \exp_after:wN \token_if_other_p:N \l_peek_token ||
1066           \exp_after:wN \token_if_letter_p:N \l_peek_token )
1067       }
1068       { \exp_after:wN \xeCJK_check_single:NNw \exp_after:wN #1 }
1069       { \__xeCJK_check_single_save:N #1 }
1070     }
1071   }
1072 }
```

(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。

```
1073 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
1074 {
1075   \xeCJK_peek_catcode_ignore_spaces:NNTF \c_catcode_letter_token
1076   {
1077     \bool_if:NNTF \l__xeCJK_peek_ignore_spaces_bool
1078     { \__xeCJK_check_single_space:NN #1#2 }
1079     { \__xeCJK_check_single_save:N #1 #2 }
1080   }
1081   {
1082     \group_align_safe_begin:
1083     \token_if_other:NNTF \l_peek_token
1084     {
```

```

1085         \group_align_safe_end:
1086         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1087         { \__xeCJK_check_single_space:NN #1#2 }
1088         { \__xeCJK_check_single_save:N #1 #2 }
1089     }
1090     {
1091         \token_if_cs:NTF \l_peek_token
1092         {
1093             \group_align_safe_end:
1094             \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1095             { \xeCJK_check_single_cs:NNn #1#2 { ~ } }
1096             { \xeCJK_check_single_cs:NNn #1#2 { } }
1097         }
1098         {
1099             \group_align_safe_end:
1100             \bool_if:nTF
1101             {
1102                 \l__xeCJK_plain_equation_bool &&
1103                 \token_if_math_toggle_p:N \l_peek_token
1104             }
1105             {
1106                 \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1107                 { \xeCJK_check_single_equation:NNnNw #1 #2 { ~ } }
1108                 { \xeCJK_check_single_equation:NNnNw #1 #2 { } }
1109             }
1110             {
1111                 \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1112                 { \__xeCJK_check_single_save:N #1 #2 ~ }
1113                 { \__xeCJK_check_single_save:N #1 #2 }
1114             }
1115         }
1116     }
1117 }
1118 }

```

(End definition for \xeCJK_check_single:NNw.)

__xeCJK_check_single_space:NN

```

1119 \cs_new_protected_nopar:Npn \__xeCJK_check_single_space:NN #1#2
1120 {
1121     \xeCJK_if_CJK_class:NTF #2
1122     {
1123         \xeCJK_if_CJK_class:NTF \l_peek_token
1124         { \__xeCJK_check_single_save:N #1 #2 }
1125         { \__xeCJK_check_single_save:N #1 #2 ~ }
1126     }
1127     { \__xeCJK_check_single_save:N #1 #2 ~ }
1128 }

```

(End definition for __xeCJK_check_single_space:NN.)

\xeCJK_check_single_equation:NNnNw

```

1129 \cs_new_protected_nopar:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
1130 {
1131     \peek_catcode:NTF \c_math_toggle_token
1132     {
1133         \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1134         \xeCJK_no_break: #2 #4
1135     }
1136     { \__xeCJK_check_single_save:N #1 #2#3#4 }
1137 }

```

(End definition for \xeCJK_check_single_equation:NNnNw.)

\xeCJK_check_single_cs:NNn 在使用 CheckSingle 选项时,在 `tablists` 宏包定义的 `tabenum` 环境中会出现下面的错误:

! Forbidden control sequence found while scanning use of \use_ii:nn.

<inserted text>

\par

1.10 \item

原因在于 tabenum 实际上是一个 TeX 对齐环境(\halign), \par 在其中被重定义为 \cr。而在下面 \tl_case:Nnn 的分支里有对 \par 的 \ifx 判断。解决办法是将判断用 \group_align_safe_begin: 和 \group_align_safe_end: 包起来。或者改用原语 \tex_par:D 作为判断条件。

```
1138 \cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
1139 {
1140   \group_align_safe_begin:
1141   \tl_case:Nnn \l_peek_token
1142     { \l__xeCJK_check_single_cs_case_tl }
1143     { \group_align_safe_end: \use_iii:nnn }
1144     { \xeCJK_check_single_env:nnNn }
1145     {
1146       \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1147       \xeCJK_no_break: #2
1148     }
1149     { \__xeCJK_check_single_save:N #1 #2#3 }
1150 }
1151 \tl_new:N \l__xeCJK_check_single_cs_case_tl
```

(End definition for \xeCJK_check_single_cs:NNn.)

\xeCJK_check_single_env:nnNn

```
1152 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
1153 {
1154   \group_align_safe_begin:
1155   \str_case:x:non {#4}
1156     { \l__xeCJK_inline_env_case_tl }
1157     { \group_align_safe_end: \use_i:nn }
1158     {#1} {#2}
1159   \scan_stop: #3 {#4}
1160 }
1161 \cs_generate_variant:Nn \str_case_x:nnn { no }
```

(End definition for \xeCJK_check_single_env:nnNn.)

NewLineCS

```
1162 \xeCJK_cs_case_keys_define:nNNnn { NewLineCS }
1163 \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
1164 { \group_align_safe_end: \use_ii:nnn }
1165 {
1166   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1167   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1168 }
```

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

EnvCS

```
1169 \xeCJK_cs_case_keys_define:nNNnn { EnvCS }
1170 \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
1171 { \group_align_safe_end: \use:n }
1172 {
1173   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1174   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1175 }
```

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

InlineEnv

```
1176 \keys_define:nn { xeCJK / options }
1177 {
1178   InlineEnv .code:n =
1179   {
```



```

1180     \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
1181     \__xeCJK_update_inline_env_case_tl:
1182   } ,
1183   InlineEnv+      .code:n =
1184   {
1185     \clist_map_inline:nn {#1}
1186     {
1187       \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
1188       { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
1189     }
1190     \__xeCJK_update_inline_env_case_tl:
1191   } ,
1192   InlineEnv-      .code:n =
1193   {
1194     \clist_map_inline:nn {#1}
1195     { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
1196     \__xeCJK_update_inline_env_case_tl:
1197   }
1198 }
1199 \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:

```

1200 \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
1201 {
1202   \tl_clear:N \l__xeCJK_inline_env_case_tl
1203   \seq_map_inline:Nn \l__xeCJK_inline_env_seq
1204   { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { \use_i:nn } } }
1205 }
1206 \tl_new:N \l__xeCJK_inline_env_case_tl

```

(End definition for __xeCJK_update_inline_env_case_tl:.)

PlainEquation

```

1207 \keys_define:nn { xeCJK / options }
1208 { 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

```

1209 \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。

```

1210 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
1211 { s > { \TrimSpaces } m > { \TrimSpaces } m }
1212 {
1213   \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
1214   \IfBooleanT {#1} { \xeCJKResetPunctClass }
1215 }
1216 \@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

```

1217 \bool_new:N \l__xeCJK_sub_cancel_bool
1218 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
1219 {
1220   \bool_if:NF \l__xeCJK_sub_cancel_bool
1221   {
1222     \bool_set_true:N \l__xeCJK_sub_cancel_bool
1223     \__xeCJK_sub_restore_or_cancel:x {#2}
1224     \IfBooleanT {#1} { \xeCJKResetPunctClass }

```

```

1225     }
1226   }
1227   \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
1228   {
1229     \bool_if:NT \l__xeCJK_sub_cancel_bool
1230     {
1231       \bool_set_false:N \l__xeCJK_sub_cancel_bool
1232       \__xeCJK_sub_restore_or_cancel:x {#2}
1233       \IfBooleanT {#1} { \xeCJKResetPunctClass }
1234     }
1235   }

```

(End definition for \xeCJKCancelSubCJKBlock and \xeCJKRestoreSubCJKBlock. These functions are documented on page 9.)

__xeCJK_sub_restore_or_cancel:n

```

1236   \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
1237   {
1238     \clist_map_inline:nn {#1}
1239     {
1240       \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
1241       {
1242         \xeCJK_declare_char_class:nx
1243         { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
1244         { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
1245       }
1246       { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
1247     }
1248   }
1249   \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
1250   \__xeCJK_msg_new:nn { SubBlock-undefined }
1251   {
1252     The~CJK~sub~block~`#1'~is~undefined.\\\
1253     Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
1254     to~declare~it.
1255   }

```

(End definition for __xeCJK_sub_restore_or_cancel:n.)

\xeCJK_declare_sub_char_class:nnn

```

1256   \cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
1257   {
1258     \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
1259     {
1260       \xeCJK_new_class:n { #1/#2 }
1261       \__xeCJK_set_sub_class_toks:nn {#1} {#2}
1262       \xeCJK_new_sub_key:n {#2}
1263     }
1264     \xeCJK_declare_char_class:nn { #1/#2 } {#3}
1265   }
1266   \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

```

1267   \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
1268   {
1269     \seq_map_inline:Nn \g__xeCJK_base_class_seq
1270     {
1271       \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
1272       \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
1273       \str_if_eq:nnTF {##1} { CJK }
1274       {
1275         \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
1276         { \__xeCJK_switch_font:nn {#1} {#2} }
1277       }

```

```

1278     {
1279         \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
1280         { \CJKsymbol }
1281         { \__xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
1282     }
1283 }
1284 \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
1285 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
1286 {
1287     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
1288     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
1289     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
1290     { \__xeCJK_switch_font:nn {#2} {##1} }
1291     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
1292     { \__xeCJK_switch_font:nn {##1} {#2} }
1293 }
1294 \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
1295 \__xeCJK_save_CJK_class:n { #1/#2 }
1296 \clist_map_inline:nn { CJK , FullLeft , FullRight }
1297 {
1298     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
1299     { \__xeCJK_switch_font:nn {#2} {#1} }
1300 }
1301 }
(End definition for \__xeCJK_set_sub_class_toks:nn.)

```

5.10 标点处理

\XeTeXglyphbouds 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。

```

1302 \__xeCJK_msg_new:nn { XeTeX-too-old }
1303 {
1304     \token_to_str:N \XeTeXglyphbouds \ is~not~defined.\\
1305     CJK~punctuation~kerning~will~not~be~available.\\
1306     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
1307 }
1308 \cs_if_exist:NF \XeTeXglyphbouds
1309 {
1310     \__xeCJK_error:n { XeTeX-too-old }
1311     \__xeCJK_after_end_preamble:n { \xeCJKsetup { PunctStyle = plain } }
1312 }

```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```

1313 \NewDocumentCommand \xeCJKsetwidth { m m }
1314 { \tl_map_inline:xn {#1} { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#2} } }
1315 \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。

```

1316 \NewDocumentCommand \xeCJKsetkern { m m m }
1317 { \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

```

1318 \tl_const:Nn \c__xeCJK_left_tl { left }
1319 \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 1320 \keys_define:nn { xeCJK / options }
LongPunct    1321 {
MiddlePunct  1322     AllowBreakBetweenPuncts .choice: ,
PunctWidth

```

```

1323 AllowBreakBetweenPuncts / true .code:n =
1324 { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_allow_break: } ,
1325 AllowBreakBetweenPuncts / false .code:n =
1326 { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_no_break: } ,
1327 AllowBreakBetweenPuncts .default:n = { true } ,
1328 KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
1329 KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
1330 KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
1331 LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
1332 LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
1333 LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
1334 MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
1335 MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
1336 MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
1337 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl
1338 }

```

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

```

\g__xeCJK_special_punct_clist
\__xeCJK_set_special_punct:nn 1339 \clist_new:N \g__xeCJK_special_punct_clist
\__xeCJK_add_special_punct:nn 1340 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
\__xeCJK_sub_special_punct:nn 1341 \cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
1342 \cs_new_nopar:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
1343 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1344 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
1345 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1346 {
1347   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
1348   { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} }{##1} } }
1349   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
1350   \tl_map_inline:xn {#2}
1351   {
1352     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} }{##1} }
1353     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1354   }
1355 }
1356 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1357 {
1358   \tl_map_inline:xn {#2}
1359   {
1360     \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
1361     {
1362       \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} }{##1} }
1363       \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1364     }
1365   }
1366 }
1367 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1368 {
1369   \tl_map_inline:xn {#2}
1370   {
1371     \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} }{##1} }
1372     \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1373   }
1374 }

```

(End definition for \g__xeCJK_special_punct_clist and others.)

```

\__xeCJK_punct_if_right_p:N
\__xeCJK_punct_if_long_p:N 1375 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
\__xeCJK_punct_if_middle_p:N 1376 {
  \__xeCJK_punct_if_mixed_width_p:N 1377   \if_int_compare:w \xeCJK_token_value_class:N #1 = \xeCJK_class_num:n { FullRight }
  \__xeCJK_punct_if_right:NTF 1378   \prg_return_true: \else: \prg_return_false: \fi:
  \__xeCJK_punct_if_long:NTF
  \__xeCJK_punct_if_middle:NTF
  \__xeCJK_punct_if_mixed_width:NTF

```

```

1379 }
1380 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1381 {
1382   \exp_args:Nc
1383   \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1384   {
1385     \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
1386     \prg_return_true: \else: \prg_return_false: \fi:
1387   }
1388 }

```

(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 1389 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
\__xeCJK_punct_dim_csname:nnnn 1390 { g__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
\__xeCJK_use_punct_dim:nnn 1391 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn
\__xeCJK_use_punct_dim:nn 1392 { \__xeCJK_punct_dim_csname:nnnn { \l_xeCJK_punct_style_tl } }
\__xeCJK_gset_punct_dim:nnx 1393 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnnn #1#2#3#4
\__xeCJK_gset_punct_dim:nnnx 1394 { g__\l_xeCJK_current_font_tl/#1/#2/#3/#4/tl }
\__xeCJK_gset_punct_dim:nnnnx 1395 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
1396 { \tl_use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
1397 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
1398 { \tl_use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
1399 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnx #1#2#3
1400 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } {#3} }
1401 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnx #1#2#3#4
1402 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } {#4} }
1403 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnnx #1#2#3#4#5
1404 { \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。

```

1405 \DeclareObjectType { xeCJK / punctuation } { \c_three }
1406 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_three }
1407 {
1408   enabled-global-setting : boolean = true ,
1409   fixed-punct-width      : length  = \c_max_dim ,
1410   fixed-punct-ratio      : real    = \c_one_fp ,
1411   mixed-punct-width      : length  = \KeyValue { fixed-punct-width } ,
1412   mixed-punct-ratio      : real    = \KeyValue { fixed-punct-ratio } ,
1413   middle-punct-width     : length  = \KeyValue { fixed-punct-width } ,
1414   middle-punct-ratio     : real    = \KeyValue { fixed-punct-ratio } ,
1415   fixed-margin-width     : length  = \c_max_dim ,
1416   fixed-margin-ratio     : real    = \c_one_fp ,
1417   mixed-margin-width     : length  = \KeyValue { fixed-margin-width } ,
1418   mixed-margin-ratio     : real    = \KeyValue { fixed-margin-ratio } ,
1419   middle-margin-width    : length  = \KeyValue { fixed-margin-width } ,
1420   middle-margin-ratio    : real    = \KeyValue { fixed-margin-ratio } ,
1421   add-min-bound-to-margin : boolean = false ,
1422   optimize-margin        : boolean = false ,
1423   margin-minimum         : length  = \c_zero_dim ,
1424   enabled-kerning        : boolean = true ,
1425   min-bound-to-kerning   : boolean = false ,
1426   kerning-total-width    : length  = \c_max_dim ,
1427   kerning-total-ratio    : real    = 0.75 ,
1428   optimize-kerning       : boolean = false ,
1429   same-align-margin      : length  = \c_max_dim ,
1430   same-align-ratio       : real    = \c_zero_fp ,
1431   different-align-margin : length  = \c_max_dim ,
1432   different-align-ratio  : real    = \c_zero_fp ,
1433   kerning-margin-width   : length  = \c_max_dim ,
1434   kerning-margin-ratio   : real    = \c_one_fp ,
1435   kerning-margin-minimum : length  = \c_zero_dim

```

```

1436 }
1437 \DeclareTemplateCode { xeCJK / punctuation } { basic } { \c_three }
1438 {
1439     enabled-global-setting = \l__xeCJK_enabled_global_setting_bool ,
1440     fixed-punct-width     = \l__xeCJK_fixed_punct_width_dim ,
1441     fixed-punct-ratio     = \l__xeCJK_fixed_punct_ratio_fp ,
1442     mixed-punct-width     = \l__xeCJK_mixed_punct_width_dim ,
1443     mixed-punct-ratio     = \l__xeCJK_mixed_punct_ratio_fp ,
1444     middle-punct-width    = \l__xeCJK_middle_punct_width_dim ,
1445     middle-punct-ratio    = \l__xeCJK_middle_punct_ratio_fp ,
1446     fixed-margin-width    = \l__xeCJK_fixed_margin_width_dim ,
1447     fixed-margin-ratio    = \l__xeCJK_fixed_margin_ratio_fp ,
1448     mixed-margin-width    = \l__xeCJK_mixed_margin_width_dim ,
1449     mixed-margin-ratio    = \l__xeCJK_mixed_margin_ratio_fp ,
1450     middle-margin-width   = \l__xeCJK_middle_margin_width_dim ,
1451     middle-margin-ratio   = \l__xeCJK_middle_margin_ratio_fp ,
1452     add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
1453     optimize-margin       = \l__xeCJK_optimize_margin_bool ,
1454     margin-minimum        = \l__xeCJK_margin_minimum_dim ,
1455     enabled-kerning        = \l__xeCJK_enabled_kerning_bool ,
1456     min-bound-to-kerning  = \l__xeCJK_min_bound_to_kerning_bool ,
1457     kerning-total-width   = \l__xeCJK_kerning_total_width_dim ,
1458     kerning-total-ratio   = \l__xeCJK_kerning_total_ratio_fp ,
1459     optimize-kerning      = \l__xeCJK_optimize_kerning_bool ,
1460     same-align-margin     = \l__xeCJK_same_align_margin_dim ,
1461     same-align-ratio      = \l__xeCJK_same_align_ratio_fp ,
1462     different-align-margin = \l__xeCJK_different_align_margin_dim ,
1463     different-align-ratio  = \l__xeCJK_different_align_ratio_fp ,
1464     kerning-margin-width  = \l__xeCJK_kerning_margin_width_dim ,
1465     kerning-margin-ratio  = \l__xeCJK_kerning_margin_ratio_fp ,
1466     kerning-margin-minimum = \l__xeCJK_kerning_margin_minimum_dim
1467 }
1468 {
1469     \AssignTemplateKeys
1470     \tl_if_blank:nTF {#3}
1471     { \xeCJK_punct_margin_process:NN {#1} {#2} }
1472     { \xeCJK_punct_kerning_process:NN {#2} {#3} }
1473 }

```

\xeCJK_punct_margin_process:NN

```

1474 \cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
1475 {
1476     \tl_set:Nx \l__xeCJK_tmpa_tl { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
1477     \tl_set:Nx \l__xeCJK_tmppb_tl
1478     {
1479         \__xeCJK_use_punct_dim:nnn { bound }
1480         { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl } {#2}
1481     }
1482     \dim_set:Nn \l__xeCJK_tmp_dim
1483     {
1484         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1485         {
1486             \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
1487             { \use_none:n }
1488             {
1489                 \xeCJK_if_blank_x:nTF \g__xeCJK_punct_width_tl
1490                 { \use_n:n }
1491                 { \g__xeCJK_punct_width_tl \use_none:n }
1492             }
1493         }
1494         { \use_n:n }
1495         {

```

```

1496     \__xeCJK_punct_if_middle:NTF {#2}
1497     { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
1498     {
1499         \__xeCJK_punct_if_mixed_width:NTF {#2}
1500         { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
1501         { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }
1502     }
1503 }
1504 }
1505 \__xeCJK_gset_punct_dim:nnn { glue } {#1} {#2}
1506 {
1507     \dim_max:nn
1508     { \l__xeCJK_margin_minimum_dim }
1509     {
1510         \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1511         {
1512             \__xeCJK_punct_if_middle:NTF {#2}
1513             {
1514                 (
1515                     \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1516                 ) / \c_two
1517             }
1518             {
1519                 \bool_if:NTF \l__xeCJK_optimize_margin_bool
1520                 { \dim_max:nn { \dim_min:nn \l__xeCJK_tpa_tl \l__xeCJK_tpb_tl } }
1521                 { \use:n }
1522                 {
1523                     \l__xeCJK_tmp_dim - ( \l__xeCJK_tpb_tl )
1524                     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1525                 }
1526             }
1527         }
1528     }
1529     \bool_if:NTF \l__xeCJK_optimize_margin_bool
1530     { \dim_min:nn { \l__xeCJK_tpa_tl } }
1531     { \use:n }
1532     {
1533         \__xeCJK_punct_if_middle:NTF {#2}
1534         {
1535             \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
1536             { \l__xeCJK_middle_margin_width_dim }
1537             {
1538                 \fp_use:N \l__xeCJK_middle_margin_ratio_fp
1539                 \etex_dimexpr:D
1540                 ( \l__xeCJK_tpa_tl + \l__xeCJK_tpb_tl ) / \c_two
1541                 \scan_stop:
1542             }
1543         }
1544         {
1545             \__xeCJK_punct_if_mixed_width:NTF {#2}
1546             { \__xeCJK_margin_width_or_ratio:n { mixed } }
1547             { \__xeCJK_margin_width_or_ratio:n { fixed } }
1548         }
1549     }
1550 }
1551 }
1552 }
1553 }

```

(End definition for \xeCJK_punct_margin_process:NN.)

__xeCJK_punct_width_or_ratio:nN

1554 \cs_new_nopar:Npn __xeCJK_punct_width_or_ratio:nN #1#2

```

1555 {
1556   \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
1557   { \use:c { l__xeCJK_#1_punct_width_dim } }
1558   {
1559     \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } = \c_zero_fp
1560     { \c_max_dim }
1561     {
1562       \fp_use:c { l__xeCJK_#1_punct_ratio_fp }
1563       \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
1564     }
1565   }
1566 }
(End definition for \__xeCJK_punct_width_or_ratio:nN.)

```

__xeCJK_margin_width_or_ratio:n

```

1567 \cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
1568 {
1569   \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
1570   { \use:c { l__xeCJK_#1_margin_width_dim } }
1571   {
1572     \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
1573     \etex_dimexpr:D \l__xeCJK_tmpa_tl \scan_stop:
1574   }
1575   \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
1576   { + \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl }
1577 }
(End definition for \__xeCJK_margin_width_or_ratio:n.)

```

\xeCJK_punct_kerning_process:NN

```

1578 \cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
1579 {
1580   \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#2}
1581   {
1582     \dim_eval:n
1583     {
1584       \bool_if:nTF
1585       {
1586         \l__xeCJK_enabled_global_setting_bool &&
1587         \tl_if_exist_p:c { g__xeCJK_punct/kern/#1/#2/tl }
1588       }
1589       { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }
1590       {
1591         \bool_if:NTF \l__xeCJK_enabled_kerning_bool
1592         { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
1593         { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
1594       }
1595       - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1596       - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} )
1597     }
1598   }
1599 }
(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。

```

1600 \cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
1601 {
1602   \dim_eval:n
1603   {
1604     \__xeCJK_use_punct_dim:nnn
1605     { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
1606     { \c__xeCJK_right_tl {#1} +
1607       \__xeCJK_use_punct_dim:nnn

```



```

1608         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1609         { \c__xeCJK_left_tl } {#2}
1610     }
1611 }
(End definition for \__xeCJK_original_kerning_margin:NN.)

```

_xeCJK_calc_kerning_margin:NN

```

1612 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:NN #1#2
1613 {
1614     \dim_max:nn
1615     { \l__xeCJK_kerning_margin_minimum_dim }
1616     {
1617         \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
1618         { \__xeCJK_punct_min_bound:NN {#1} {#2} }
1619         {
1620             \bool_if:NTF \l__xeCJK_optimize_kerning_bool
1621             { \dim_max:nn { \__xeCJK_punct_min_bound:NN {#1} {#2} } }
1622             { \use:n }
1623             {
1624                 \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
1625                 { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
1626                 {
1627                     \fp_compare:nNnTF \l__xeCJK_kerning_total_ratio_fp = \c_zero_fp
1628                     {
1629                         \xeCJK_if_same_class:NNTF {#1} {#2}
1630                         { \__xeCJK_kerning_width_or_ratio:nNN { same } }
1631                         { \__xeCJK_kerning_width_or_ratio:nNN { different } }
1632                     }
1633                     {
1634                         \__xeCJK_calc_kerning_margin:nNN
1635                         {
1636                             \fp_use:N \l__xeCJK_kerning_total_ratio_fp
1637                             \etex_dimexpr:D
1638                             \__xeCJK_use_punct_dim:nn { width } {#1} +
1639                             \__xeCJK_use_punct_dim:nn { width } {#2}
1640                             \scan_stop:
1641                         }
1642                     }
1643                 }
1644                 {#1} {#2}
1645             }
1646         }
1647     }
1648 }
(End definition for \__xeCJK_calc_kerning_margin:NN.)

```

_xeCJK_kerning_width_or_ratio:nNN

```

1649 \cs_new_nopar:Npn \__xeCJK_kerning_width_or_ratio:nNN #1#2#3
1650 {
1651     \dim_compare:nNnTF { \use:c { \l__xeCJK_#1_align_margin_dim } } < \c_max_dim
1652     { \use:c { \l__xeCJK_#1_align_margin_dim } \use_none:nn }
1653     {
1654         \fp_compare:nNnTF { \use:c { \l__xeCJK_#1_align_ratio_fp } } = \c_zero_fp
1655         { \use:n }
1656         { \fp_use:c { \l__xeCJK_#1_align_ratio_fp } \use_ii:nn }
1657     }
1658     {
1659         \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
1660         { \l__xeCJK_kerning_margin_width_dim \use_none:n }
1661         { \fp_use:N \l__xeCJK_kerning_margin_ratio_fp \use:n }
1662     }
1663     { \etex_dimexpr:D \__xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
1664 }

```

(End definition for __xeCJK_kerning_width_or_ratio:nNN.)

__xeCJK_punct_min_bound:NN

```

1665 \cs_new_nopar:Npn \__xeCJK_punct_min_bound:NN #1#2
1666 {
1667   \dim_max:nn
1668   {
1669     \dim_min:nn
1670     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} }
1671     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
1672   }
1673   {
1674     \dim_min:nn
1675     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} }
1676     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
1677   }
1678 }

```

(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。

```

1679 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:nNN #1#2#3
1680 {
1681   \dim_eval:n
1682   {
1683     (#1)
1684     - ( \__xeCJK_use_punct_dim:nnn
1685         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1686         { \c__xeCJK_left_tl } {#2} )
1687     - ( \__xeCJK_use_punct_dim:nnn
1688         { \__xeCJK_punct_if_right:NTF {#3} { glue } { bound } }
1689         { \c__xeCJK_right_tl } {#3} )
1690     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1691     - ( \__xeCJK_use_punct_dim:nn { dimen } {#3} )
1692   }
1693 }

```

(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。

```

1694 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
1695 {
1696   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }
1697   {
1698     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1699     {
1700       \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim }
1701       \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
1702       \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
1703     }
1704     {
1705       { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
1706       \__xeCJK_punct_if_long:NTF {#2}
1707       { \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim } }
1708       {
1709         \UseInstance { xeCJK / punctuation }
1710         { \l_xeCJK_punct_style_tl } {#1} {#2} { }
1711       }
1712     }
1713   }
1714 }

```

(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。

```

1715 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
1716 {
1717   \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_left_tl {#1}
1718   { \xeCJK_glyph_bounds:NN \c_one {#1} }
1719   \__xeCJK_gset_punct_dim:nnx { bound } \c__xeCJK_right_tl {#1}
1720   { \xeCJK_glyph_bounds:NN \c_three {#1} }
1721   \dim_set:Nn \l__xeCJK_tmp_dim
1722   {
1723     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} ) +
1724     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1725   }
1726   \__xeCJK_gset_punct_dim:nnx { width } {#1}
1727   { \dim_use:N \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
1728   \__xeCJK_gset_punct_dim:nnx { dimen } {#1}
1729   { \dim_eval:n { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim } }
1730   \__xeCJK_punct_if_long:NT {#1}
1731   {
1732     \seq_map_inline:Nn \g__xeCJK_punct_style_seq
1733     {
1734       \__xeCJK_gset_punct_dim:nnnx {##1} { kern } {#1} {#1}
1735       {
1736         \bool_if:nTF
1737         {
1738           \str_if_eq_p:nn {#1} { ^^^^2025 } ||
1739           \str_if_eq_p:nn {#1} { ^^^^2026 }
1740         }
1741         { \c_zero_dim }
1742         { - \dim_use:N \l__xeCJK_tmp_dim }
1743       }
1744     }
1745   }
1746 }
1747 \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。

```

1748 \cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
1749 {
1750   \dim_use:N \XeTeXglyphbounds #1 ~
1751   \XeTeXcharglyph \xeCJK_token_value_charcode:N #2 \exp_stop_f:
1752 }
(End definition for \xeCJK_glyph_bounds:NN.)

```

`\xeCJK_get_punct_kerning:NN` 标点挤压。

```

1753 \cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
1754 {
1755   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
1756   {
1757     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1758     { \__xeCJK_gset_punct_dim:nnx { kern } {#1} {#2} { \c_zero_dim } }
1759     {
1760       \UseInstance { xeCJK / punctuation }
1761       { \l_xeCJK_punct_style_tl } { } {#1} {#2}
1762     }
1763   }
1764 }
1765 \cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
(End definition for \xeCJK_get_punct_kerning:NN.)

```

PunctStyle

```
1766 \keys_define:nn { xeCJK / options }
1767 {
1768   PunctStyle .choice: ,
1769   PunctStyle .default:n = { quanjiao } ,
1770   PunctStyle / halfwidth .meta:n = { PunctStyle = banjiao } ,
1771   PunctStyle / fullwidth .meta:n = { PunctStyle = quanjiao } ,
1772   PunctStyle / mixedwidth .meta:n = { PunctStyle = kaiming } ,
1773   PunctStyle / marginkerning .meta:n = { PunctStyle = hangmobanjiangiao } ,
1774   PunctStyle / plain .code:n =
1775     { \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl } ,
1776   PunctStyle / unknown .code:n =
1777     {
1778       \IfInstanceExistTF { xeCJK / punctuation } { \l_keys_value_tl }
1779       { \tl_set:Nx \l_xeCJK_punct_style_tl { \l_keys_value_tl } }
1780       { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
1781     }
1782 }
1783 \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
1784 \__xeCJK_msg_new:nn { punct-style-unknown }
1785 {
1786   Punctuation~style~"#1"~is~unknown. \\\
1787   The~available~styles~are~listed~as~follow.\\\
1788   "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
1789 }
```

(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。

```
1790 \NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
1791 {
1792   \IfInstanceExistTF { xeCJK / punctuation } {#1}
1793     { \__xeCJK_warning:nx { punct-style-already-defined } {#1} }
1794     { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
1795   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
1796 }
1797 \seq_new:N \g__xeCJK_punct_style_seq
1798 \__xeCJK_msg_new:nn { punct-style-already-defined }
1799 {
1800   Punctuation~style~"#1"~is~already~defined!. \\\
1801   The~existing~style~of~"#1"~will~be~overwritten.\\
1802 }
1803 \@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。

```
1804 \NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
1805 {
1806   \IfInstanceExistTF { xeCJK / punctuation } {#1}
1807     { \EditInstance { xeCJK / punctuation } {#1} {#2} }
1808     { \__xeCJK_error:nx { punct-style-unknown } {#1} }
1809 }
1810 \@onlypreamble \xeCJKEditPunctStyle
```

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

默认设置即为全角格式。

```
1811 \xeCJKDeclarePunctStyle { quanjiao } { }
1812 \xeCJKDeclarePunctStyle { hangmobanjiangiao } { enabled-kerning = false }
1813 \xeCJKDeclarePunctStyle { banjiao }
1814 {
1815   fixed-punct-ratio = 0.5 ,
1816   optimize-margin = true ,
1817   kerning-total-ratio = 0.5 ,
```

```

1818     optimize-kerning    = true
1819 }

1820 \xeCJKDeclarePunctStyle { kaiming }
1821 {
1822     fixed-punct-ratio    = 0.5 ,
1823     mixed-punct-ratio    = 0.8 ,
1824     optimize-margin      = true ,
1825     kerning-total-ratio  = 0.5 ,
1826     optimize-kerning     = true
1827 }

1828 \xeCJKDeclarePunctStyle { CCT }
1829 {
1830     fixed-punct-ratio    = 0.7 ,
1831     optimize-margin      = true ,
1832     kerning-total-ratio  = 0.6 ,
1833     optimize-kerning     = true
1834 }

```

5.11 后备字体

AutoFallback 后备字体的宏包选项声明。

```

1835 \keys_define:nn { xeCJK / options }
1836 {
1837     AutoFallback .choice: ,
1838     AutoFallback / true .code:n =
1839     {
1840         \cs_if_eq:NNF \CJKsymbol \xeCJK_fallback_test_glyph:N
1841         {
1842             \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1843             \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
1844         }
1845     } ,
1846     AutoFallback / false .code:n =
1847     {
1848         \cs_if_eq:NNT \CJKsymbol \xeCJK_fallback_test_glyph:N
1849         { \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N }
1850     } ,
1851     AutoFallback .default:n = { true } ,
1852     fallback .meta:n = { AutoFallback = true }
1853 }

```

(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。

```

1854 \cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
1855 {
1856     \xeCJK_glyph_if_exist:NTF {#1}
1857     { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1858     { { \xeCJK_fallback_loop:Nn {#1} { \l_xeCJK_family_tl/FallBack } } }
1859 }

```

(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。

```

1860 \cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
1861 {
1862     \xeCJK_family_if_exist:xF {#2}
1863     {
1864         \xeCJK_select_font:x {#2}
1865         \xeCJK_glyph_if_exist:NTF {#1}
1866         { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1867         { \xeCJK_fallback_loop:Nn {#1} { #2/FallBack } }

```

```

1868     }
1869     {
1870         \xeCJK_family_if_exist:xT { \l__xeCJK_family_tl/FallBack }
1871         { \__xeCJK_warning:nxx { fallback } {#1} {#2} }
1872         \__xeCJK_fallback_save_CJKsymbol:N {#1}
1873     }
1874 }
1875 \__xeCJK_msg_new:nn { fallback }
1876 {
1877     CJKfamily~`#2'~
1878     ( \prop_get:No \g__xeCJK_family_font_name_prop {#2} )~
1879     does~not~contain~glyph~`#1'~(U+\int_to_hexadecimal:n {'#1}).\\
1880 }
1881 \bool_new:N \l__xeCJK_fallback_first_bool
1882 \cs_generate_variant:Nn \prop_get:Nn { No }
(End definition for \xeCJK_fallback_loop:Nn.)

```

\setCJKfallbackfamilyfont

```

1883 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1884 {
1885     \clist_set:Nx \l__xeCJK_tmpa_clist {#3}
1886     \int_compare:nNnTF { \clist_count:N \l__xeCJK_tmpa_clist } > \c_one
1887     { \xeCJK_set_family_fallback:xxN {#1} {#2} \l__xeCJK_tmpa_clist }
1888     { \xeCJK_set_family:xxx { #1/FallBack } {#2} {#3} }
1889 }
(End definition for \setCJKfallbackfamilyfont. This function is documented on page 7.)

```

\xeCJK_set_family_fallback:nnN

```

1890 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnN #1#2#3
1891 {
1892     \group_begin:
1893     \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
1894     \prop_get:NVNF \g__xeCJK_family_font_name_prop
1895     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
1896     { \tl_clear:N \l__xeCJK_font_name_tl }
1897     \clist_map_inline:Nn #3
1898     {
1899         \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
1900         \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
1901         \clist_put_left:cn
1902         { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist } {#2}
1903         \xeCJK_set_family:Vvv \l__xeCJK_fallback_family_tl
1904         { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist }
1905         { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_name_tl }
1906     }
1907     \group_end:
1908 }
1909 \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

```

1910 \bool_new:N \g__xeCJK_auto_fake_bold_bool
1911 \bool_new:N \g__xeCJK_auto_fake_slant_bool
1912 \fp_new:N \g__xeCJK_embolden_factor_fp
1913 \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。

```

1914 \keys_define:nn { xeCJK / options }
1915 {
1916     AutoFakeBold .choices:nn = { true , false }
1917     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,

```

```

1918     AutoFakeBold / unknown .code:n =
1919     {
1920         \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
1921         \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1922     } ,
1923     AutoFakeBold .default:n = { true } ,
1924     AutoFakeSlant .choices:nn = { true , false }
1925     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
1926     AutoFakeSlant / unknown .code:n =
1927     {
1928         \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
1929         \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
1930     } ,
1931     AutoFakeSlant .default:n = { true } ,
1932     EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
1933     SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
1934     BoldFont .meta:n = { AutoFakeBold = true } ,
1935     boldfont .meta:n = { AutoFakeBold = true } ,
1936     SlantFont .meta:n = { AutoFakeSlant = true } ,
1937     slantfont .meta:n = { AutoFakeSlant = true }
1938 }

```

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

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

```

\g__xeCJK_sub_key_seq
1939 \seq_new:N \g__xeCJK_sub_key_seq
1940 \cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
1941 {
1942     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
1943     \keys_define:nn { xeCJK / features }
1944     {
1945         #1 .code:n =
1946         {
1947             \tl_if_blank:nTF {##1}
1948             {
1949                 \bool_set_false:c { l__xeCJK_copy_#1_bool }
1950                 \bool_set_false:c { l__xeCJK_add_#1_bool }
1951                 \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
1952             }
1953             {
1954                 \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
1955                 \str_if_eq:nnTF {##1} { * }
1956                 {
1957                     \bool_set_true:c { l__xeCJK_copy_#1_bool }
1958                     \bool_set_false:c { l__xeCJK_add_#1_bool }
1959                 }
1960                 {
1961                     \bool_set_false:c { l__xeCJK_copy_#1_bool }
1962                     \bool_set_true:c { l__xeCJK_add_#1_bool }
1963                     \__xeCJK_get_sub_features:nn {#1} {##1}
1964                 }
1965             }
1966         } ,
1967         #1 .default:n = { }
1968     }
1969 }

```

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

__xeCJK_get_sub_features:nn

__xeCJK_get_sub_features:wn

```

1970 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
1971 {
1972     \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
1973     \clist_clear:N \l__xeCJK_sub_font_options_clist

```

```

1974 \tl_if_head_eq_meaning:VNTF \l__xeCJK_tmpa_tl [ % ]
1975 {
1976   \exp_after:wN \__xeCJK_get_sub_features:wn \l__xeCJK_tmpa_tl \c_empty_tl
1977   \tl_if_blank:VT \l__xeCJK_sub_font_name_tl
1978   {
1979     \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl
1980     \clist_clear:N \l__xeCJK_sub_font_options_clist
1981   }
1982 }
1983 { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl }
1984 \tl_if_blank:VTF \l__xeCJK_sub_font_name_tl
1985 { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
1986 { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
1987 \clist_set_eq:cN { \l__xeCJK_#1_font_options_clist } \l__xeCJK_sub_font_options_clist
1988 \tl_set_eq:cN { \l__xeCJK_#1_font_name_tl } \l__xeCJK_sub_font_name_tl
1989 }
1990 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:wn [#1] #2
1991 {
1992   \clist_set:Nn \l__xeCJK_sub_font_options_clist {#1}
1993   \str_if_eq:nnTF {#2} { \c_empty_tl }
1994   { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \c_empty_tl }
1995   { \tl_set:Nn \l__xeCJK_sub_font_name_tl {#2} }
1996 }
1997 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
1998 \cs_generate_variant:Nn \tl_if_head_eq_meaning:nNTF { V }
1999 \cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
(End definition for \__xeCJK_get_sub_features:nn and \__xeCJK_get_sub_features:wn.)

```

FallBack

```

2000 \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 2001 \prop_new:N \g__xeCJK_features_id_prop
      ItalicFont 2002 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
      BoldItalicFont 2003 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
      SlantedFont 2004 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
      BoldSlantedFont 2005 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
      BoldFeatures 2006 \prop_put:Nnn \g__xeCJK_features_id_prop { bfsl } { BoldSlanted }
      ItalicFeatures 2007 \prop_map_inline:Nn \g__xeCJK_features_id_prop
      BoldItalicFeatures 2008 {
      SlantedFeatures 2009   \keys_define:nn { xeCJK / features }
      BoldSlantedFeatures 2010 {
      2011     #2Font .tl_set:c = { \l__xeCJK_font_name_#1_tl } ,
      2012     #2Features .clist_set:c = { \l__xeCJK_font_feat_#1_clist }
      2013   }
      2014 }
(End definition for \g__xeCJK_features_id_prop and others.)

```

AutoFakeBold

AutoFakeSlant

```

2015 \keys_define:nn { xeCJK / features }
2016 {
2017   AutoFakeBold .choice: ,
2018   AutoFakeBold / false .code:n =
2019     { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
2020   AutoFakeBold / unknown .code:n =
2021     {
2022       \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
2023       \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
2024     } ,
2025   AutoFakeBold .default:n = { \g__xeCJK_embolden_factor_fp } ,
2026   AutoFakeSlant .choice: ,

```



```

2027     AutoFakeSlant / false .code:n =
2028     { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
2029     AutoFakeSlant / unknown .code:n =
2030     {
2031         \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
2032         \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
2033     } ,
2034     AutoFakeSlant .default:n = { \g__xeCJK_slant_factor_fp }
2035 }

```

(End definition for AutoFakeBold and AutoFakeSlant.)

__xeCJK_set_family_initial:

```

2036 \cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
2037 {
2038     \int_gincr:N \g__xeCJK_family_int
2039     \prop_clear:N \l__xeCJK_add_font_prop
2040     \prop_map_inline:Nn \g__xeCJK_features_id_prop
2041     {
2042         \tl_clear:c { l__xeCJK_font_name_##1_tl }
2043         \clist_clear:c { l__xeCJK_font_feat_##1_clist }
2044     }
2045     \clist_clear:N \l__xeCJK_fontspec_options_clist
2046     \seq_clear:N \l__xeCJK_sub_key_seq
2047     \seq_map_inline:Nn \g__xeCJK_sub_key_seq
2048     {
2049         \bool_set_false:c { l__xeCJK_copy_##1_bool }
2050         \bool_set_false:c { l__xeCJK_add_##1_bool }
2051     }
2052     \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
2053     \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
2054     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
2055     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
2056 }
2057 \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。

```

2058 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
2059 {
2060     \group_begin:
2061     \__xeCJK_set_family_initial:
2062     \tl_set:Nn \l__xeCJK_family_name_tl {#1}
2063     \clist_set:Nn \l__xeCJK_font_options_clist {#2}
2064     \tl_set:Nn \l__xeCJK_font_name_tl {#3}
2065     \clist_concat:NNN \l__xeCJK_font_features_clist
2066         \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
2067     \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_features_clist
2068     \keys_set_known:nVN { xeCJK / features }
2069         \l__xeCJK_font_features_clist \l__xeCJK_fontspec_options_clist
2070     \__xeCJK_parse_features:
2071     \__xeCJK_pass_features:
2072     \__xeCJK_check_family:V \l__xeCJK_family_name_tl
2073     \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
2074     \__xeCJK_save_family_info:
2075     \__xeCJK_set_sub_block_family:
2076     \group_end:
2077 }
2078 \cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , xxx , Vvv }

```

(End definition for \xeCJK_set_family:nnn.)

__xeCJK_remove_duplicate_keys:N

```

2079 \cs_new_protected_nopar:Npn \__xeCJK_remove_duplicate_keys:N #1

```

```

2080 {
2081   \prop_clear:N \l__xeCJK_tmp_prop
2082   \keyval_parse:NNV \__xeCJK_prop_put_aux:n \__xeCJK_prop_put_aux:nn #1
2083   \clist_clear:N #1
2084   \prop_map_inline:Nn \l__xeCJK_tmp_prop
2085   {
2086     \tl_if_blank:nTF {##2}
2087     { \clist_put_right:Nn #1 {##1} }
2088     { \clist_put_right:Nn #1 { ##1 = {##2} } }
2089   }
2090 }
2091 \cs_generate_variant:Nn \keyval_parse:NNn { NNV }
2092 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
2093 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} { } }
2094 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
2095 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} {#2} }
(End definition for \__xeCJK_remove_duplicate_keys:N.)

```

__xeCJK_gset_family_cs:x

```

2096 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_cs:x #1
2097 {
2098   \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n {#1} }
2099   {
2100     \group_begin:
2101     \exp_not:n { \cs_set_eq:NN \__xeCJK_update_family:n \use_none:n }
2102     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
2103     { \exp_not:V \l__xeCJK_fontspec_options_clist }
2104     { \exp_not:V \l__xeCJK_font_name_tl }
2105     \__xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
2106     \group_end:
2107   }
2108 }
(End definition for \__xeCJK_gset_family_cs:x.)

```

__xeCJK_check_family:n

```

2109 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
2110 {
2111   \prop_get:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmpa_tl
2112   {
2113     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl
2114     { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
2115     \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmpa_tl }
2116   }
2117 }
2118 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
2119 \__xeCJK_msg_new:nn { CJKfamily-redef }
2120 { 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

```

2121 \cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
2122 { \prop_put:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2123 \cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
2124 { \prop_put_if_new:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2125 \cs_generate_variant:Nn \__xeCJK_add_font:nn { nx , nV , nv }
2126 \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

```

2127 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
2128 {
2129   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }

```

```

2130     { FakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
2131   }
2132   \cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
2133   {
2134     \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2135     { FakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
2136   }

```

(End definition for __xeCJK_add_fake_bold:n and __xeCJK_add_fake_slant:n.)

```

\__xeCJK_parse_features:
\__xeCJK_parse_features:nn
2137 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:
2138 { \prop_map_function:NN \g__xeCJK_features_id_prop \__xeCJK_parse_features:nn }
2139 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:nn #1#2
2140 {
2141   \__xeCJK_if_font_select:nTF {#1}
2142   {
2143     \__xeCJK_add_font:nv {#2} { l__xeCJK_font_name_#1_tl }
2144     \__xeCJK_if_it_or_sl:nTF {#1}
2145     {
2146       \__xeCJK_if_font_select:nF { bf #1 }
2147       { \__xeCJK_add_font_if_new:nv {#2} { l__xeCJK_font_name_#1_tl } }
2148     }
2149     {
2150       \str_if_eq:nnT {#1} { bf }
2151       {
2152         \__xeCJK_if_font_select:nF { bfit }
2153         { \__xeCJK_add_font:nv { BoldItalic } \l__xeCJK_font_name_bf_tl }
2154         \__xeCJK_if_font_select:nF { bfsl }
2155         { \__xeCJK_add_font:nv { BoldSlanted } \l__xeCJK_font_name_bf_tl }
2156       }
2157     }
2158   }
2159   { \__xeCJK_set_fake_font:nn {#1} {#2} }
2160 }

```

(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
2161 \prg_new_conditional:Npnn \__xeCJK_if_font_select:n #1 { p , T , F , TF }
2162 {
2163   \tl_if_blank:vTF { l__xeCJK_font_name_#1_tl }
2164   { \prg_return_false: } { \prg_return_true: }
2165 }
2166 \prg_new_conditional:Npnn \__xeCJK_if_it_or_sl:n #1 { p , T , F , TF }
2167 {
2168   \if_predicate:w
2169   \bool_if_p:n { \str_if_eq_p:nn { it } {#1} || \str_if_eq_p:nn { sl } {#1} }
2170   \prg_return_true: \else: \prg_return_false: \fi:
2171 }
2172 \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
2173 \cs_new_protected_nopar:Npn \__xeCJK_set_fake_font:nn #1#2
2174 {
2175   \str_if_eq:nnTF {#1} { bf }
2176   { \bool_if:NT \l__xeCJK_auto_fake_bold_bool { \__xeCJK_add_fake_bold:n {#1} } }
2177   {
2178     \bool_if:NTF \l__xeCJK_auto_fake_slant_bool
2179     {
2180       \bool_if:nT
2181       { \__xeCJK_if_it_or_sl_p:n {#1}
2182         || ( \str_if_eq_p:nn {#1} { bfit }

```

```

2183         && ! ( \__xeCJK_if_font_select_p:n { it } ) )
2184     || ( \str_if_eq_p:nn {#1} { bfsl }
2185         && ! ( \__xeCJK_if_font_select_p:n { sl } ) )
2186     }
2187     { \__xeCJK_add_fake_slant:n {#1} }
2188 }
2189 { \__xeCJK_if_it_or_sl:nT {#1} { \__xeCJK_map_it_sl:nn {#1} {#2} } }
2190 \bool_if:nT
2191 {
2192     \l__xeCJK_auto_fake_bold_bool
2193     && ! ( \__xeCJK_if_it_or_sl_p:n {#1} )
2194     && ! ( \__xeCJK_if_font_select_p:n { bf } )
2195 }
2196 { \__xeCJK_add_fake_bold:n {#1} }
2197 }
2198 \__xeCJK_add_font_if_new:nn {#2} { * }
2199 }

```

(End definition for __xeCJK_set_fake_font:nn.)

__xeCJK_map_it_sl:n

```

2199 \cs_new_protected_nopar:Npn \__xeCJK_map_it_sl:nn #1#2
2200 {
2201     \__xeCJK_if_map_font_select:nT {#1}
2202     {
2203         \__xeCJK_add_font:nx {#2} { \__xeCJK_get_map_font:n {#1} }
2204         \__xeCJK_if_font_select:nF { bf #1 }
2205         { \__xeCJK_add_font_if_new:nx { Bold #2 } { \__xeCJK_get_map_font:n {#1} } }
2206     }
2207 }

```

(End definition for __xeCJK_map_it_sl:n.)

__xeCJK_get_map_font:n

```

2208 \cs_new_nopar:Npn \__xeCJK_get_map_font:n #1
2209 { \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

```

2210 \prg_new_conditional:Npnn \__xeCJK_if_map_font_select:n #1 { p , T , F , TF }
2211 {
2212     \xeCJK_if_blank_x:nTF { \__xeCJK_get_map_font:n {#1} }
2213     { \prg_return_false: } { \prg_return_true: }
2214 }

```

(End definition for __xeCJK_if_map_font_select:n.)

__xeCJK_pass_features:

```

2215 \cs_new_protected_nopar:Npn \__xeCJK_pass_features:
2216 {
2217     \prop_map_inline:Nn \g__xeCJK_features_id_prop
2218     {
2219         \clist_if_empty:cF { l__xeCJK_font_feat_##1_clist }
2220         {
2221             \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2222             { ##2Features = { \exp_not:v { l__xeCJK_font_feat_##1_clist } } }
2223         }
2224     }
2225     \prop_map_inline:Nn \l__xeCJK_add_font_prop
2226     { \clist_put_right:Nn \l__xeCJK_fontspec_options_clist { ##1 = {##2} } }
2227 }

```

(End definition for __xeCJK_pass_features:.)

\g__xeCJK_family_name_prop

\g__xeCJK_family_font_name_prop

\g__xeCJK_family_font_options_prop

```

2228 \prop_new:N \g__xeCJK_family_name_prop
2229 \prop_new:N \g__xeCJK_family_font_name_prop
2230 \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:

```
2231 \cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
2232 {
2233   \group_begin:
2234   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
2235     { \clist_remove_all:Nn \l__xeCJK_font_options_clist {##1} }
2236   \prop_gput:NVV \g__xeCJK_family_font_name_prop
2237     \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
2238   \prop_gput:NVV \g__xeCJK_family_font_options_prop
2239     \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
2240   \group_end:
2241 }
```

(End definition for __xeCJK_save_family_info:.)

_xeCJK_set_sub_block_family:

```
2242 \cs_new_protected_nopar:Npn \_xeCJK_set_sub_block_family:
2243 {
2244   \seq_map_inline:Nn \l__xeCJK_sub_key_seq
2245     {
2246       \tl_set:Nx \l__xeCJK_sub_family_tl { \l__xeCJK_family_name_tl/##1 }
2247       \bool_if:cT { l__xeCJK_copy_##1_bool } { \_xeCJK_copy_sub_family:n {##1} }
2248       \bool_if:cT { l__xeCJK_add_##1_bool }
2249         {
2250           \xeCJK_set_family:Vvv \l__xeCJK_sub_family_tl
2251             { l__xeCJK_##1_font_options_clist } { l__xeCJK_##1_font_name_tl }
2252         }
2253     }
2254 }
2255 \cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
2256 {
2257   \__xeCJK_check_family:V \l__xeCJK_sub_family_tl
2258   \prop_get:NVNT \g__xeCJK_family_font_name_prop
2259     \l__xeCJK_family_name_tl \l__xeCJK_tmpa_tl
2260   {
2261     \prop_gput:NVV \g__xeCJK_family_font_name_prop
2262       \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl
2263   }
2264   \prop_get:NVNT \g__xeCJK_family_font_options_prop
2265     \l__xeCJK_family_name_tl \l__xeCJK_tmpa_clist
2266   {
2267     \clist_remove_all:Nn \l__xeCJK_tmpa_clist { #1 = * }
2268     \prop_gput:NVV \g__xeCJK_family_font_options_prop
2269       \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
2270   }
2271   \cs_gset_protected_nopar:cpx { \_xeCJK_family_csname:n { \l__xeCJK_sub_family_tl } }
2272     {
2273       \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }
2274       {
2275         \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
2276           { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_tmpa_tl
2277         {
2278           \__xeCJK_gset_family_nfss_cs:xx
2279             { \l__xeCJK_sub_family_tl } { \exp_not:N \l__xeCJK_tmpa_tl }
2280         }
2281       }
2282     }
2283 }
```

(End definition for _xeCJK_set_sub_block_family:.)

__xeCJK_copy_family:nn

```

2284 \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
2285 {
2286   \xeCJK_family_if_exist:xT {#2}
2287   {
2288     \tl_map_inline:nn
2289     {
2290       \g__xeCJK_family_name_prop
2291       \g__xeCJK_family_font_name_prop
2292       \g__xeCJK_family_font_options_prop
2293     }
2294     {
2295       \prop_get:NnNT ##1 {#2} \l__xeCJK_tmpa_tl
2296       { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_tl }
2297     }
2298     \cs_gset_eq:cc
2299     { \__xeCJK_family_nfss_csname:n {#1} }
2300     { \__xeCJK_family_nfss_csname:n {#2} }
2301   }
2302 }
2303 \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
2304 \tl_new:N \l_xeCJK_current_font_tl
2305 \tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
2306 \cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
2307 \cs_new_protected_nopar:Npn \xeCJK_select_font:
2308 {
2309   \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
2310   {
2311     \tl_set:Nx \l__xeCJK_current_coord_tl { \l_xeCJK_current_font_tl }
2312     \__xeCJK_family_use:x { \l_xeCJK_family_tl }
2313     \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coord_tl }
2314   }
2315 }
2316 \cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
2317 {
2318   \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
2319   {
2320     \tl_set:Nx \l__xeCJK_current_coord_tl { \__xeCJK_font_csname:n {#1} }
2321     \__xeCJK_family_use:x {#1}
2322     \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coord_tl }
2323   }
2324 }
2325 \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。

```

2326 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
2327 {
2328   \str_if_eq:nnF {#1} {#2}
2329   {
2330     \__xeCJK_info:nxx { CJK-block } {#1} {#2}
2331     \str_if_eq:nnTF {#2} { CJK }
2332     { \xeCJK_select_font: }
2333     { \__xeCJK_block_select_font:n {#2} }
2334   }
2335 }
2336 \__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。

```
2337 \cs_new_protected_nopar:Npn \_xeCJK_block_select_font:n #1
2338 {
2339   \cs_if_exist_use:cF { \_xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
2340   {
2341     \tl_set:Nx \l__xeCJK_current_coor_tl
2342     { \_xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
2343     \xeCJK_family_if_exist:xF { \l_xeCJK_family_tl/#1 }
2344     {
2345       \_xeCJK_copy_family:xx { \l_xeCJK_family_tl/#1 }
2346       {
2347         \cs_if_exist:cTF
2348         { \_xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
2349         { \CJKfamilydefault/#1 } { \l_xeCJK_family_tl }
2350       }
2351     }
2352     \_xeCJK_family_use:x { \l_xeCJK_family_tl/#1 }
2353     \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2354   }
2355 }
```

(End definition for `_xeCJK_block_select_font:n`.)

```
\_xeCJK_family_csname:n
  \_xeCJK_family_nfss_csname:n 2356 \cs_new_nopar:Npn \_xeCJK_family_csname:n #1 { xeCJK/family/#1 }
\_xeCJK_family_use:x 2357 \cs_new_nopar:Npn \_xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
\_xeCJK_gset_family_nfss_cs:xx 2358 \cs_new_nopar:Npn \_xeCJK_family_use:x #1 { \use:c { \_xeCJK_family_nfss_csname:n {#1} } }
2359 \cs_new_protected_nopar:Npn \_xeCJK_gset_family_nfss_cs:xx #1#2
2360 {
2361   \prop_gput:Nxx \g__xeCJK_family_name_prop {#1} {#2}
2362   \cs_gset_protected_nopar:cpx { \_xeCJK_family_nfss_csname:n {#1} }
2363   {
2364     \tl_set:Nx \exp_not:N \f@encoding { \c__xeCJK_encoding_tl }
2365     \tl_set:Nx \exp_not:N \f@family {#2}
2366     \exp_not:N \selectfont
2367   }
2368 }
2369 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
(End definition for \_xeCJK_family_csname:n and others.)
```

`\xeCJK_family_if_exist:xF`

```
2370 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
2371 {
2372   \cs_if_exist:cTF { \_xeCJK_family_nfss_csname:n {#1} }
2373   { \use_i:nn }
2374   { \cs_if_exist_use:cTF { \_xeCJK_family_csname:n {#1} } }
2375   { \prg_return_true: } { \prg_return_false: }
2376 }
```

(End definition for `\xeCJK_family_if_exist:xF`.)

`\CJKfamily` 用于切换 CJK 字体族。

```
2377 \NewDocumentCommand \CJKfamily { t+ t- m }
2378 {
2379   \xeCJK_if_blank_x:nTF {#3}
2380   {
2381     \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
2382     \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
2383   }
2384   {
2385     \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
2386     {
2387       \xeCJK_family_if_exist:xF {#3}
2388     }
2389   }
2390 }
```

```

2388         {
2389             \tl_set:Nx \l_xeCJK_family_tl {#3}
2390             \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2391             \IfBooleanT {#1} { \__xeCJK_family_use:x {#3} }
2392         }
2393         { \__xeCJK_family_unknown_warning:x {#3} }
2394     }
2395 }
2396 \tex_ignorespaces:D
2397 }

```

(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。

```

2398 \tl_new:N \l_xeCJK_family_tl
(End definition for \l_xeCJK_family_tl.)

```

`__xeCJK_gobble_CJKfamily:`

```

2399 \cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
2400 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
2401 \DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }
(End definition for \__xeCJK_gobble_CJKfamily:.)

```

`\xeCJK_family_if_exist_use:x`

```

2402 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
2403 {
2404     \xeCJK_family_if_exist:xF {#1}
2405     { \__xeCJK_family_use:x {#1} }
2406     { \__xeCJK_family_unknown_warning:x {#1} }
2407 }
(End definition for \xeCJK_family_if_exist_use:x.)

```

`__xeCJK_family_unknown_warning:n`

```

2408 \cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:n #1
2409 {
2410     \prop_if_empty:NF \g__xeCJK_family_font_name_prop
2411     {
2412         \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
2413         {
2414             \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
2415             \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
2416         }
2417     }
2418 }
2419 \cs_generate_variant:Nn \__xeCJK_family_unknown_warning:n { x }
2420 \seq_new:N \g__xeCJK_unknown_family_seq
2421 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
2422 {
2423     Unknown~CJK~family~'\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
2424     Try~to~use~'\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2425 }
2426 \cs_new_nopar:Npn \__xeCJK_msg_def_family_map:n #1
2427 {
2428     \str_case:x:nnn {#1}
2429     {
2430         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
2431         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
2432         \CJKttdefault { \token_to_str:N \setCJKmonofont }
2433     }
2434     { \token_to_str:N \setCJKfamilyfont {#1} }
2435     [...]{}
2436 }
2437 \cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1

```



```

2438 {
2439   \str_case_x:nnn {#1}
2440   {
2441     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
2442     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
2443     \CJKttdefault { \token_to_str:N \CJKttdefault }
2444   }
2445   {#1}
2446 }

```

(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
\setCJKmonofont
2447 \NewDocumentCommand \setCJKmainfont { O{} m }
2448 {
2449   \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2}
2450   \normalfont
2451 }
2452 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
2453 \NewDocumentCommand \setCJKsansfont { O{} m }
2454 {
2455   \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2}
2456   \normalfont
2457 }
2458 \NewDocumentCommand \setCJKmonofont { O{} m }
2459 {
2460   \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2}
2461   \normalfont
2462 }

```

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

```

2463 \@onlypreamble \setCJKmainfont
2464 \@onlypreamble \setCJKmathfont
2465 \@onlypreamble \setCJKsansfont
2466 \@onlypreamble \setCJKmonofont
2467 \@onlypreamble \setCJKromanfont

```

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

```

\newCJKfontfamily
\CJKfontspec
2468 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
2469 { \xeCJK_set_family:xxx {#1} {#2} {#3} }
2470 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
2471 {
2472   \tl_set:Nx \l__xeCJK_tmpa_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
2473   \cs_new_protected_nopar:Npx #2 { \CJKfamily { \l__xeCJK_tmpa_tl } }
2474   \xeCJK_set_family:xxx { \l__xeCJK_tmpa_tl } {#3} {#4}
2475 }
2476 \NewDocumentCommand \CJKfontspec { O{} m }
2477 {
2478   \xeCJK_fontspec:xx {#1} {#2}
2479   \tex_ignorespaces:D
2480 }

```

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

\xeCJK_fontspec:nn

```

2481 \cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
2482 {
2483   \tl_set:Nx \l__xeCJK_tmpa_tl { \tl_to_str:n { CJKfontspec/#1/#2/id } }
2484   \tl_if_exist:cF { \l__xeCJK_tmpa_tl }
2485   {
2486     \tl_set:Nx \l__xeCJK_tmpb_tl
2487       { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
2488     \tl_gset_eq:cN { \l__xeCJK_tmpa_tl } \l__xeCJK_tmpb_tl
2489     \xeCJK_set_family:Vnn \l__xeCJK_tmpb_tl {#1} {#2}
2490   }

```

```

2491 \exp_args:Nc \CJKfamily { \l__xeCJK_tmpa_tl }
2492 }
2493 \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
2494 \clist_new:N \g__xeCJK_default_features_clist
2495 \NewDocumentCommand \defaultCJKfontfeatures { m }
2496 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
2497 \@onlypreamble \defaultCJKfontfeatures
2498 \NewDocumentCommand \addCJKfontfeatures { m }
2499 {
2500   \xeCJK_add_font_features:x {#1}
2501   \tex_ignorespaces:D
2502 }
2503 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
(End definition for \defaultCJKfontfeatures and \addCJKfontfeatures. These functions are documented on page 6.)

```

\xeCJK_add_font_features:n

```

2504 \cs_new_protected_nopar:Npn \xeCJK_add_font_features:n #1
2505 {
2506   \prop_get:NVNTF \g__xeCJK_family_font_name_prop \l_xeCJK_family_tl \l__xeCJK_tmpa_tl
2507   {
2508     \clist_set:Nn \l__xeCJK_tmpa_clist {#1}
2509     \seq_map_function:NN
2510       \g__xeCJK_CJK_sub_class_seq \__xeCJK_parse_sub_class_features:n
2511     \prop_get:NVNT \g__xeCJK_family_font_options_prop
2512       \l_xeCJK_family_tl \l__xeCJK_tmpb_clist
2513     {
2514       \clist_concat:NNN
2515         \l__xeCJK_tmpa_clist \l__xeCJK_tmpb_clist \l__xeCJK_tmpa_clist
2516     }
2517     \xeCJK_fontspec:VV \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_tl
2518   }
2519   { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
2520 }
2521 \cs_generate_variant:Nn \xeCJK_add_font_features:n { x }
2522 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
2523 {
2524   \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
2525   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
2526 }
(End definition for \xeCJK_add_font_features:n.)

```

__xeCJK_parse_sub_class_features:n

```

2527 \cs_new_protected_nopar:Npn \__xeCJK_parse_sub_class_features:n #1
2528 {
2529   \clist_if_in:NnT \l__xeCJK_tmpa_clist {#1}
2530   {
2531     \clist_remove_all:Nn \l__xeCJK_tmpa_clist {#1}
2532     \prop_get:NoNF \g__xeCJK_family_font_name_prop
2533       { \l_xeCJK_family_tl/#1 } \l__xeCJK_tmpb_tl
2534     {
2535       \prop_get:NxNF \g__xeCJK_family_font_name_prop
2536         { \CJKfamilydefault/#1 } \l__xeCJK_tmpb_tl
2537       { \tl_set_eq:NN \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
2538     }
2539     \clist_set:Nx \l__xeCJK_tmpa_clist
2540       { #1 = { [ \exp_not:V \l__xeCJK_tmpa_clist ] { \exp_not:V \l__xeCJK_tmpb_tl } } }
2541     \clist_map_break:
2542   }
2543 }
2544 \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。

```
2545 \__xeCJK_at_end_preamble:n
2546 {
2547   \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
2548   \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
2549   {
2550     \tl_gset:Nx \CJKfamilydefault
2551     {
2552       \str_case:x:nnn { \familydefault }
2553       {
2554         { \rmdefault } { \exp_not:N \CJKrmdefault }
2555         { \sfdefault } { \exp_not:N \CJKsfdefault }
2556         { \ttdefault } { \exp_not:N \CJKttdefault }
2557       }
2558       { \CJKfamilydefault }
2559     }
2560   }
2561   \cs_undefine:N \__xeCJK_family_default_wrap:n
2562   \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
2563   { \__xeCJK_warning:nx { no-CJKfamily } { \CJKfamilydefault } }
2564   {
2565     \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2566     {
2567       \tl_set_eq:NN \l__xeCJK_tmpa_tl \CJKfamilydefault
2568       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
2569       { \use:n }
2570       {
2571         \xeCJK_family_if_exist:xF { \CJKrmdefault }
2572         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
2573       }
2574       {
2575         \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
2576         { \prop_map_break:n { \tl_gset:Nn \CJKfamilydefault {#1} } }
2577       }
2578       \__xeCJK_warning:nxx { CJKfamilydefault-undefined }
2579       { \l__xeCJK_tmpa_tl } { \CJKfamilydefault }
2580     }
2581     \CJKfamily { \CJKfamilydefault }
2582     \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
2583   }
2584 }
2585 \__xeCJK_msg_new:nn { no-CJKfamily }
2586 {
2587   It~seems~that~you~have~not~declare~a~CJKfamily.\\
2588   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
2589   '\__xeCJK_msg_def_family_map:n {#1}'\\
2590   in~the~preamble~to~declare~the~default~CJKfamily.\\
2591 }
2592 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
2593 {
2594   Undefined~CJK~default~family~'\__xeCJK_msg_family_map:n {#1}'~
2595   has~been~replaced~by~'\__xeCJK_msg_family_map:n {#2}'.\\
2596   Try~to~use~'\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2597 }
```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```
2598 \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。

```
2599 \NewDocumentCommand \setCJKmathfont { O{} m }  
2600 { \xeCJK_set_family:xxx { \c__xeCJK_math_tl } {#1} {#2} }  
2601 \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。

```
2602 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:  
2603 {  
2604   \xeCJK_family_if_exist:xF { \c__xeCJK_math_tl }  
2605   { \tl_set:Nx \l__xeCJK_tmpa_tl { \c__xeCJK_math_tl } }  
2606   {  
2607     \xeCJK_family_if_exist:xF { \CJKfamilydefault }  
2608     { \tl_set:Nx \l__xeCJK_tmpa_tl { \CJKfamilydefault } }  
2609     { \use_none:nnnnn }  
2610   }  
2611   \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl  
2612   {  
2613     \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmpb_tl }  
2614     \DeclareSymbolFont \c__xeCJK_math_tl  
2615       \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \mddefault \shapedefault  
2616     \SetSymbolFont \c__xeCJK_math_tl { bold }  
2617       \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \bfdefault \shapedefault  
2618     \int_const:Nn \c__xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }  
2619     \clist_concat:NNN  
2620       \l__xeCJK_tmpa_clist \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist  
2621     \clist_concat:NNN  
2622       \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_clist \c__xeCJK_FullRight_chars_clist  
2623     \clist_map_inline:Nn \l__xeCJK_tmpa_clist  
2624       {  
2625         \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}  
2626         { \c_seven } { \c__xeCJK_math_fam_int }  
2627       }  
2628   }  
2629 }
```

(End definition for \xeCJK_set_mathfont:.)

\xeCJK_gset_mathcode:nnnn

```
2630 \cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4  
2631 {  
2632   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int  
2633   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }  
2634   {  
2635     \tex_global:D \xeCJK_xetex_mathcode:w  
2636     \l__xeCJK_tmpa_int = #3 ~ #4 \l__xeCJK_tmpa_int  
2637     \int_incr:N \l__xeCJK_tmpa_int  
2638   }  
2639 }
```

(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。

```

2640 \keys_define:nn { xeCJK / options }
2641 {
2642   Verb .choice: ,
2643   Verb / true .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_zero } ,
2644   Verb / env .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_one } ,
2645   Verb / env+ .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_two } ,
2646   Verb / false .code:n = { \int_set_eq:NN \l__xeCJK_verb_case_int \c_three }
2647 }
2648 \int_new:N \l__xeCJK_verb_case_int
2649 \__xeCJK_after_preamble:n
2650 {
2651   \tl_put_right:Nn \verbatim@font
2652   {
2653     \if_case:w \l__xeCJK_verb_case_int
2654       \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2655       \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2656     \or:
2657       \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
2658         { \xeCJKVerbAddon }
2659         {
2660           \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2661           \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2662         }
2663     \or:
2664       \int_compare:nNnT \etex_currentgrouptype:D = \c_fourteen
2665         { \xeCJKVerbAddon }
2666     \fi:
2667   }
2668 }

```

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

`\xeCJKVerbAddon` `\xeCJKOffVerbAddon` `\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。

```

2669 \NewDocumentCommand \xeCJKVerbAddon { }
2670 {
2671   \int_compare:nNnF \etex_currentgrouplevel:D = \c_zero
2672   {
2673     \bool_if:NF \l__xeCJK_listings_env_bool
2674     {
2675       \__xeCJK_set_verb_exspace:
2676       \__xeCJK_verb_addon:
2677     }
2678   }
2679 }
2680 \bool_new:N \l__xeCJK_listings_env_bool
2681 \NewDocumentCommand \xeCJKOffVerbAddon { } { }
2682 \cs_new_protected_nopar:Npn \__xeCJK_verb_addon:
2683 {
2684   \bool_if:NF \l__xeCJK_verb_addon_bool
2685   {
2686     \bool_set_true:N \l__xeCJK_verb_addon_bool
2687     \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
2688     \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
2689     \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
2690     \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
2691     \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
2692     \cs_set_eq:NN \__xeCJK_verb_CJKglue: \CJKglue

```

```

2693 \cs_set_eq:NN \__xeCJK_verb_CJKecglue: \CJKecglue
2694 \cs_set_eq:NN \__xeCJK_verb_check_for_glue: \xeCJK_check_for_glue:
2695 \cs_set_eq:NN \__xeCJK_verb_ignore_spaces:w \xeCJK_ignore_spaces:w
2696 \cs_set_protected_nopar:Npx \xeCJKOffVerbAddon
2697 {
2698   \__xeCJK_reset_char_class:n { FullLeft }
2699   \__xeCJK_reset_char_class:n { FullRight }
2700   \__xeCJK_reset_char_class:n { HalfLeft }
2701   \__xeCJK_reset_char_class:n { HalfLeft }
2702   \__xeCJK_reset_char_class:n { NormalSpace }
2703   \skip_set:Nn \tex_rightskip:D { \skip_use:N \tex_rightskip:D }
2704   \exp_not:c
2705     { bool_set_ \bool_if:NTF \l__xeCJK_xecglue_bool { true } { false } :N }
2706     \l__xeCJK_xecglue_bool
2707   \exp_not:n
2708     {
2709       \cs_set_eq:NN \CJKglue \__xeCJK_verb_CJKglue:
2710       \cs_set_eq:NN \CJKecglue \__xeCJK_verb_CJKecglue:
2711       \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_verb_check_for_glue:
2712       \cs_set_eq:NN \xeCJK_ignore_spaces:w \__xeCJK_verb_ignore_spaces:w
2713     }
2714   }
2715   \tex_output:D \exp_after:wN \exp_after:wN \exp_after:wN
2716   {
2717     \exp_after:wN \exp_after:wN
2718     \exp_after:wN \xeCJKOffVerbAddon
2719     \exp_after:wN \use:n
2720     \tex_the:D \tex_output:D
2721   }
2722   \xeCJKsetup { xCJKecglue = false }
2723   \cs_set_protected_nopar:Npn \CJKglue
2724     { \skip_horizontal:N \g__xeCJK_verb_exspace_skip }
2725   \cs_set_protected_nopar:Npn \CJKecglue
2726     { \skip_horizontal:n { 0.5 \g__xeCJK_verb_exspace_skip } }
2727   \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
2728   \cs_set_eq:NN \xeCJK_ignore_spaces:w \CJKecglue
2729   \skip_set:Nn \tex_rightskip:D { \c_zero_dim plus 1 fil }
2730 }
2731 }
2732 \cs_new_protected_nopar:Npn \__xeCJK_reset_char_class:n #1
2733 {
2734   \int_set:Nn \l__xeCJK_tmpa_int { \xeCJK_class_num:n {#1} }
2735   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
2736     { \XeTeXcharclass ##1 = \l__xeCJK_tmpa_int }
2737 }
2738 \bool_new:N \l__xeCJK_verb_addon_bool
2739 \cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon

```

(End definition for \xeCJKVerbAddon and \xeCJKOffVerbAddon. These functions are 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。

```

2740 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
2741 {
2742   \tl_if_exist:cF { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
2743   {
2744     \group_begin:
2745     \xeCJK_select_font:
2746     \use:x
2747     {
2748       \group_end:

```

```

2749 \tl_gset:cx
2750 \exp_not:n { { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2751 {
2752   \exp_not:N \dim_max:nn { \c_zero_dim }
2753   {
2754     \c_two \tex_fontdimen:D \c_two \tex_font:D -
2755     \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00
2756   }
2757 }
2758 }
2759 }
2760 \skip_gset:Nn \g__xeCJK_verb_exspace_skip
2761 { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2762 }
2763 \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。

```

2764 \cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
2765 { {
2766   \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
2767   { \xeCJK_set_visible_space_font: }
2768   \textvisiblespace
2769 } }
2770 \AtEndOfPackage
2771 {
2772   \tl_put_left:Nn \fontspec_visible_space: { \scan_stop: }
2773   \cs_gset_eq:NN \fontspec_visible_space_fallback: \xeCJK_visible_space_fallback:
2774 }
(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。

```

2775 \cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
2776 {
2777   \tl_set:Nx \l__xeCJK_current_coord_tl { xeCJK/space/\curr@fontshape/\f@size }
2778   \exp_after:wN \__xeCJK_set_visible_space_size:n
2779   \exp_after:wN { \dim_use:N \tex_fontdimen:D \c_two \tex_font:D }
2780   \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coord_tl }
2781 }
2782 \cs_new_protected_nopar:Npn \__xeCJK_set_visible_space_size:n #1
2783 {
2784   \tl_set:Nx \f@encoding { \g_fontspec_encoding_tl }
2785   \tl_set:Nx \f@family { lmtt }
2786   \selectfont
2787   \dim_compare:nNf {#1} = { \tex_fontdimen:D \c_two \tex_font:D }
2788   {
2789     \fontsize
2790     {
2791       \dim_eval:n
2792       {
2793         \f@size pt *
2794         \dim_ratio:nn {#1} { \tex_fontdimen:D \c_two \tex_font:D }
2795       }
2796     }
2797     { \f@baselineskip }
2798     \selectfont
2799   }

```

```
2800 }
(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。

```
2801 \keys_define:nn { xeCJK / options }
2802 {
2803   LocalConfig .choice: ,
2804   LocalConfig / false .code:n =
2805     { \bool_gset_false:N \g__xeCJK_config_bool } ,
2806   LocalConfig / true .code:n =
2807     {
2808       \bool_gset_true:N \g__xeCJK_config_bool
2809       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
2810     } ,
2811   LocalConfig / unknown .code:n =
2812     {
2813       \bool_gset_true:N \g__xeCJK_config_bool
2814       \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
2815     } ,
2816   LocalConfig .default:n = { true }
2817 }
2818 \tl_new:N \g__xeCJK_config_name_tl
2819 \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

```
2820 \keys_define:nn { xeCJK / options }
2821 {
2822   CJKNumber .bool_gset:N = \g__xeCJK_number_bool ,
2823   indentfirst .bool_gset:N = \g__xeCJK_indent_bool ,
2824   normalindentfirst .meta:n = { indentfirst = false }
2825 }
(End definition for CJKNumber and indentfirst. These functions are documented on page 3.)
```

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

```
2826 \keys_define:nn { xeCJK / options }
2827 {
2828   quiet .code:n =
2829     {
2830       \msg_redirect_module:nnn { xeCJK } { warning } { info }
2831       \msg_redirect_module:nnn { xeCJK } { info } { none }
2832       \xeCJK_if_package_loaded:nF { fontspec }
2833       { \PassOptionsToPackage { quiet } { fontspec } }
2834     } ,
2835   silent .code:n =
2836     {
2837       \msg_redirect_module:nnn { xeCJK } { warning } { none }
2838       \msg_redirect_module:nnn { xeCJK } { info } { none }
2839       \xeCJK_if_package_loaded:nF { fontspec }
2840       { \PassOptionsToPackage { silent } { fontspec } }
2841     } ,
2842   unknown .code:n =
2843     {
2844       \xeCJK_if_package_loaded:nTF { fontspec }
2845       { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2846       { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2847     }
2848 }
2849 \__xeCJK_msg_new:nn { key-unknown }
```



```

2850 {
2851   Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~`#1'.\\
2852   The~key~`#1'~is~being~ignored.
2853 }
(End definition for quiet and silent.)

```

5.17 xeCJK 初始化设置

```

\CJKsymbol
\CJKpunctsymbol
2854 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2855 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}
(End definition for \CJKsymbol and \CJKpunctsymbol.)
xeCJK 宏包的初始化设置。

2856 \keys_set:nn { xeCJK / options }
2857 {
2858   CJKglue      = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
2859   CJKecglue    = { ~ } ,
2860   xCJKecglue    = false ,
2861   CheckSingle   = false ,
2862   PlainEquation = false ,
2863   CheckFullRight = false ,
2864   CJKspace      = false ,
2865   CJKmath       = false ,
2866   CJKnumber     = false ,
2867   xeCJKactive   = true ,
2868   LocalConfig   = true ,
2869   indentfirst   = true ,
2870   Verb         = env ,
2871   EmboldenFactor = 4 ,
2872   SlantFactor    = 0.167 ,
2873   PunctStyle     = quanjiao ,
2874   NewLineCS      = { \par \[ } ,
2875   EnvCS          = { \begin \end } ,
2876   NoBreakCS      = { \footnote \footnotemark \nobreak } ,
2877   KaiMingPunct   = { ^^^^3002 ^^^^ff0e ^^^^ff1f ^^^^ff01 } ,
2878   LongPunct      = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^2025 ^^^^2026 } ,
2879   MiddlePunct    = { ^^^^2014 ^^^^2015 ^^^^2027 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
2880   AllowBreakBetweenPuncts = false
2881 }
2882 \defaultCJKfontfeatures { Script = CJK }

执行宏包选项,并载入 fontspec 宏包。

2883 \ProcessKeysOptions { xeCJK / options }
2884 \RequirePackage { fontspec } [ 2011/09/13 ]

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

2885 \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。

2886 \bool_if:NT \g__xeCJK_indent_bool { \RequirePackage { indentfirst } }
对不能通过 \xeCJKsetup 设置的选项给出警告。

2887 \keys_define:nn { xeCJK / options }
2888 {
2889   LocalConfig .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2890   CJKnumber   .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2891   indentfirst .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } }
2892 }
2893 \__xeCJK_msg_new:nn { option-invalid }
2894 {
2895   The~`#1'~option~only~can~be~set~in~the~optional~argument~to~the\\
2896   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\

```

```

2897 Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
2898 }

```

\CJKrmdefault
 \CJKsfdefault
 \CJKttdefault
 \CJKfamilydefault

```

2899 \tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
2900 \tl_if_exist:NF \CJKsfdefault { \tl_gset:Nn \CJKsfdefault { sf } }
2901 \tl_if_exist:NF \CJKttdefault { \tl_gset:Nn \CJKttdefault { tt } }
2902 \tl_new:N \l__xeCJK_family_default_init_tl
2903 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
2904 \tl_gset:Nx \l__xeCJK_family_default_init_tl
2905 {
2906   \exp_not:N \__xeCJK_family_default_wrap:n
2907   {
2908     \tl_if_exist:NTF \CJKfamilydefault
2909       { \exp_not:V \CJKfamilydefault }
2910       { \exp_not:N \CJKrmdefault }
2911   }
2912 }
2913 \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。

```

2914 \NewDocumentCommand \xeCJKsetup { +m }
2915 {
2916   \keys_set:nn { xeCJK / options } {#1}
2917   \tex_ignorespaces:D
2918 }

```

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

\xeCJKseteboldenfactor

\xeCJKsetslantfactor

```

2919 \NewDocumentCommand \xeCJKseteboldenfactor { m }
2920 { \xeCJKsetup { EmboldenFactor = {#1} } }
2921 \NewDocumentCommand \xeCJKsetslantfactor { m }
2922 { \xeCJKsetup { SlantFactor = {#1} } }

```

(End definition for \xeCJKseteboldenfactor and \xeCJKsetslantfactor.)

\punctstyle

\xeCJKplainchr

```

2923 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2924 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }

```

(End definition for \punctstyle and \xeCJKplainchr.)

\CJKsetecglue

```

2925 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2926 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue

```

(End definition for \CJKsetecglue.)

\CJKspace

\CJKnospace

```

2927 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
2928 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }

```

(End definition for \CJKspace and \CJKnospace.)

\xeCJKallowbreakbetweenpuncts

\xeCJKnobreakbetweenpuncts

```

2929 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
2930 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
2931 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
2932 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }

```

(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts.)

```

\XeCJKenablefallback
\XeCJKdisablefallback
2933 \NewDocumentCommand \XeCJKenablefallback { }
2934 { \XeCJKsetup { AutoFallBack = true } }
2935 \NewDocumentCommand \XeCJKdisablefallback { }
2936 { \XeCJKsetup { AutoFallBack = false } }
(End definition for \XeCJKenablefallback and \XeCJKdisablefallback.)

```

```

\XeCJKsetcharclass
2937 \NewDocumentCommand \XeCJKsetcharclass { m m m }
2938 {
2939   \XeCJK_set_char_class:nnn {#1} {#2} {#3}
2940   \XeCJKResetPunctClass
2941 }
(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 \LaTeX 2_ϵ 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。

```

2942 \cs_if_exist:cT { ? - \token_to_str:N \nobreakspace }
2943 {
2944   \exp_args:NNv \ProvideTextCommandDefault \nobreakspace
2945   { ? - \token_to_str:N \nobreakspace }
2946 }
(End definition for \nobreakspace.)

```

`\XeCJK@text@composite` 这里修改的目的是让 `\'{^e}` 等直接展开成在 xunicode 中由下面定义的 `\char"1EBF\relax`,也就是让这些定义生效:

```
\DeclareUTFcomposite[\UTFencname]{x1EBF}{\'}{^e}
```

按照 source2e 中 `\@text@composite` 的定义,目前 xunicode 对这种三个记号组合的定义不会起作用。`\'{^e}` 将会被展开成 `\char"00EA\relax\char"0301\relax` 这种组合字符。由于中间的 `\relax` 会造成状态转移,按照 xeCJK 的机制,就会插入 kern,最终破坏了 \XeTeX 将两个字符正确的组合起来。除此之外,还可以将 xunicode 中所有 `\char` 数字后面的 `\relax` 全部改成普通空格,这样就不会留下多余的 `\relax`。或者使用 lowercase 技巧,直接由 Unicode 得到实际字符,而不需要通过 `\char`。但是,如果在 xeCJK 中修改的话,还要 `\ReloadXunicode` 才会起作用。需要注意的是,如果原来字体中缺少 é 这个字符,但是有 ê 这个字符的话,`\'{^e}` 会输出为 ê,现在则是留下一个空白。

```

2947 \cs_new_nopar:Npn \XeCJK@text@composite #1#2#3 \@text@composite
2948 {
2949   \str_if_eq:nnTF {#3} { \@empty }
2950   { \orig@text@composite #1#2#3 \@text@composite }
2951   { \__XeCJK_text_composite:Nnw #1#2#3 }
2952 }
2953 \cs_new_nopar:Npn \__XeCJK_text_composite:Nnw #1#2 \@empty
2954 {

```

```

2955 \tl_if_empty:nTF {#2}
2956 { \orig@text@composite #1 \@empty \@text@composite }
2957 { \cs_if_exist_use:cF { \token_to_str:N #1 - \token_to_str:N #2 } }
2958 }
2959 \cs_new_eq:NN \orig@text@composite \@text@composite
2960 \cs_set_eq:NN \@text@composite \xeCJK@text@composite
(End definition for \xeCJK@text@composite.)
2961 \__xeCJK_msg_new:nn { conflict-package }
2962 {
2963   The~`#1'~package~is~too~old. \\\
2964   Please~update~an~up~to~date~version~of~it\\
2965   using~your~TeX~package~manager~or~from~CTAN.
2966 }
2967 \xeCJK_if_package_loaded:nTF { realscripts }
2968 {
2969   \@ifpackagelater { realscripts } { 2010/10/10 } { }
2970   {
2971     \__xeCJK_error:nx { conflict-package }
2972     {
2973       \xeCJK_if_package_loaded:nTF { xltextra }
2974       { xltextra } { realscripts }
2975     }
2976   }
2977 }
2978 {
2979   \cs_new_eq:NN \__xeCJK_dim_max:nn \dim_max:nn
2980   \cs_new_eq:NN \__xeCJK_dim_min:nn \dim_min:nn
2981   \__xeCJK_at_end_preamble:n
2982   {
2983     \xeCJK_if_package_loaded:nT { realscripts }
2984     {
2985       \@ifpackagelater { realscripts } { 2010/10/10 } { }
2986       {
2987         \cs_gset_eq:NN \dim_max:nn \__xeCJK_dim_max:nn
2988         \cs_gset_eq:NN \dim_min:nn \__xeCJK_dim_min:nn
2989       }
2990     }
2991     \cs_undefine:N \__xeCJK_dim_max:nn
2992     \cs_undefine:N \__xeCJK_dim_min:nn
2993   }
2994 }

```

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

```

2995 \RenewDocumentCommand \fontfamily { m }
2996 {
2997   \tl_set:Nx \f@family {#1}
2998   \__xeCJK_update_family:n
2999   {
3000     \str_case:nnn {#1}
3001     {
3002       { \rmdefault } { \CJKfamily { \CJKrmdefault } }
3003       { \sfdefault } { \CJKfamily { \CJKsfdefault } }
3004       { \ttdefault } { \CJKfamily { \CJKttdefault } }
3005       { \familydefault } { \CJKfamily { \CJKfamilydefault } }
3006     }
3007     { }
3008   }
3009 }
3010 \cs_new_eq:NN \__xeCJK_update_family:n \use:n
(End definition for \fontfamily.)

```

`\xeCJK@fix@penalty` 对 $\text{\LaTeX 2}_{\epsilon}$ 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。

```

3012 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
3013 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
3014 \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。

```

3015 \cs_new_protected_nopar:Npn \xeCJK@italiccorr
3016 {
3017   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
3018   {
3019     \xeCJK_if_last_node:nTF { default }
3020     {
3021       \tex_unkern:D \tex_unkern:D \@@italiccorr
3022       { \xeCJK_make_node:n { default } }
3023     }
3024     {
3025       \xeCJK_if_last_node:nTF { CJK }
3026       {
3027         \tex_unkern:D \tex_unkern:D \@@italiccorr
3028         { \xeCJK_make_node:n { CJK } } \use:n
3029       }
3030       {
3031         \xeCJK_if_last_node:nTF { CJK-space }
3032         {
3033           \tex_unkern:D \tex_unkern:D \@@italiccorr
3034           { \xeCJK_make_node:n { CJK-space } } \use:n
3035         }
3036         { \@@italiccorr \use_none:n }
3037       }
3038     }
3039   }
3040 }

```

`\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:`)。

```

3038 {
3039   \exp_after:wN \exp_after:wN \exp_after:wN
3040   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3041   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3042   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3043   \xeCJK_ignore_spaces:w
3044 }
3045 }
3046 {
3047   { \@@italiccorr }
3048 }
(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。

```

3050 \__xeCJK_after_end_preamble:n
3051 {
3052   \int_compare:nNnF
3053   { \c_three + \seq_count:N \g__xeCJK_new_class_seq } = \xe@alloc@intercharclass
3054   {
3055     \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
3056     {

```

```

3057         \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
3058         { \__xeCJK_set_others_toks:n {#1} }
3059     }
3060 }
3061 }
3062 \cs_new_protected_nopar:Npn \__xeCJK_set_others_toks:n #1
3063 {
3064     \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
3065     \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
3066     {
3067         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
3068         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
3069         \xeCJK_app_inter_class_toks:nnx {##1} { Others }
3070         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
3071         \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
3072         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
3073         \xeCJK_if_blank_x:nT
3074         { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
3075         {
3076             \xeCJK_copy_inter_class_toks:nnnn
3077             { Others } { Boundary } { Default } { Boundary }
3078         }
3079         \xeCJK_if_blank_x:nT
3080         { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
3081         {
3082             \xeCJK_copy_inter_class_toks:nnnn
3083             { Boundary } { Others } { Boundary } { Default }
3084         }
3085     }
3086 }
(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:
3087 \cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
3088 \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。在使用 T1 编码的时候，`\r{u}` 的实际定义是 `\char183` 与常被用作中文间隔号的 U+00B7 冲突。

```

3089 \__xeCJK_after_preamble:n
3090 {
3091     \tl_map_inline:nn
3092     {
3093         \textellipsis \textemdash \textperiodcentered \textcentereddot
3094         \textquoteleft \textquoteright \textquotedblleft \textquotedblright
3095     }
3096     {
3097         \cs_gset_nopar:Npx #1
3098         { \__xeCJK_group_begin: \makeXeCJKinactive \exp_not:o {#1} \__xeCJK_group_end: }
3099     }
3100     \tl_put_left:Nn \tipaencoding { \makeXeCJKinactive }
3101     \cs_new_eq:NN \__xeCJK_aux_r:n \r
3102     \cs_set_nopar:Npn \r #1
3103     {
3104         \bool_if:nTF
3105         {
3106             \str_if_eq_x_p:nn { \f@encoding } { T1 } &&
3107             \str_if_eq_x_p:nn {#1} { u }
3108         }
3109         { { \makeXeCJKinactive \__xeCJK_aux_r:n {#1} } }
3110         { \__xeCJK_aux_r:n {#1} }
3111     }

```

```

3112 \xeCJK_if_package_loaded:nT { pifont }
3113 {
3114   \RenewDocumentCommand \Pifont { m }
3115   { \makexeCJKinactive \usefont { U } { #1 } { m } { n } }
3116 }
3117 }

```

简单处理与 `hyperref` 宏包的兼容问题。

```

3118 \__xeCJK_after_end_preamble:n
3119 {
3120   \xeCJK_if_package_loaded:nT { hyperref }
3121   {
3122     \pdfstringdefDisableCommands
3123     {
3124       \__xeCJK_gobble_CJKfamily:
3125       \xeCJK_cs_clear:N \makexeCJKinactive
3126       \xeCJK_cs_clear:N \__xeCJK_group_begin:
3127       \xeCJK_cs_clear:N \__xeCJK_group_end:
3128     }
3129   }
3130 }

```

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

```

3131 \__xeCJK_after_end_preamble:n
3132 {
3133   \bool_if:nT
3134   { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
3135   { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
3136 }

```

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。

```

3137 \tl_set:cn { ver@CJK.sty } { 9999/99/99 }
3138 \tl_set_eq:cc { ver@CJKulem.sty } { ver@CJK.sty }
3139 \tl_set_eq:cc { ver@CJKpunct.sty } { ver@CJK.sty }

```

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

```

3140 \cs_if_exist:NF \CJK@ifundefined
3141 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
3142 \NewDocumentCommand \xeCJKcaption { o m }
3143 {
3144   \IfNoValueF { #1 } { \XeTeXdefaultencoding "#1" }
3145   \use:x
3146   {
3147     \char_set_catcode_letter:n { 64 }
3148     \file_input:n { #2.cpx }
3149     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
3150   }
3151   \XeTeXdefaultencoding "UTF-8"
3152 }

```

(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`。

判断过于繁琐,应该在 `ctex` 包中妥善处理。这段代码应在 `ctex` 包发布新版本后删去。

```

3153 \cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
3154 { \AtEndOfPackage { \RequirePackage { xeCJKfntef } } }
3155 {
3156   \__xeCJK_at_end_preamble:n
3157   {
3158     \xeCJK_if_package_loaded:nF { xeCJKfntef }
3159     {
3160       \xeCJK_if_package_loaded:nTF { CJKfntef }
3161       { \RequirePackage { xeCJKfntef } }
3162       {

```

```

3163         \xCJK_if_package_loaded:nT { ulem }
3164         { \RequirePackage { xeCJKfntef } }
3165     }
3166 }
3167 }
3168 }

```

导言区末尾检测到 listings 时,自动载入 xeCJK-listings。

```

3169 \__xeCJK_at_end_preamble:n
3170 {
3171     \xCJK_if_package_loaded:nT { listings }
3172     { \RequirePackage { xeCJK-listings } }
3173 }

```

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。

```

3174 \bool_if:NT \g__xeCJK_number_bool
3175 {
3176     \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
3177     \cs_set_protected:Npn \CJKaddEncHook #1#2
3178     { \cs_set:cpn { __xeCJK_enc_#1_hook: } {#2} }
3179     \cs_set_protected_nopar:Npn \Unicode #1#2
3180     { \tex_char:D \etex_numexpr:D (#1) * \c_two_hundred_fifty_six + (#2) \scan_stop: }
3181     \RequirePackage { CJKnumb }
3182     \cs_if_exist_use:c { __xeCJK_enc_ \CJK@UnicodeEnc _hook: }
3183     \tl_set:Nn \CJK@tenthousand { \Unicode { 78 } { 7 } }
3184     \tl_set:Nn \CJK@hundredmillion { \Unicode { 78 } { 191 } }
3185 }

```

最后引入本地配置文件。

```

3186 \bool_if:NT \g__xeCJK_config_bool
3187 {
3188     \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
3189     \@onefilewithoptions
3190     { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
3191 }
3192 </package>

```

5.19 xeCJKfntef

```

3193 <*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。

```

3194 \PassOptionsToPackage { normalem } { ulem }
3195 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
3196 \ProcessOptions \scan_stop:
3197 \RequirePackage { xeCJK }
3198 \RequirePackage { ulem }
3199 \RequirePackage { CJKfntef }
3200 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

```

\xeCJK_hook_for_ulem:

```

3201 \cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
3202 {
3203     \bool_if:NF \l__xeCJK_ulem_hook_used_bool
3204     {
3205         \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
3206         \xeCJKsetup { CheckFullRight = false }
3207         \bool_if:NTF \l__xeCJK_ulem_skip_punct_bool
3208         { \cs_set_eq:NN \__xeCJK_ulem_leader_type: \UL@leadtype }
3209         {
3210             \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
3211             \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:

```



```

3212     }
3213     \__xeCJK_ulem_initial:
3214     \xeCJK_glue_to_skip:nN
3215     {
3216         \cs_set_eq:NN \ \tex_space:D
3217         \cs_set_eq:NN \penalty \tex_penalty:D
3218         \cs_set_eq:NN \hskip \skip_horizontal:N
3219         \CJKglue
3220     } \l__xeCJK_ccglue_skip
3221     \xeCJK_glue_to_skip:nN
3222     {
3223         \cs_set_eq:NN \ \tex_space:D
3224         \cs_set_eq:NN \penalty \tex_penalty:D
3225         \cs_set_eq:NN \hskip \skip_horizontal:N
3226         \CJKecglue
3227     } \l__xeCJK_ecglue_skip
3228     \cs_set_protected_nopar:Npn \CJKglue
3229     { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
3230     \cs_set_protected_nopar:Npn \CJKecglue
3231     { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
3232 }
3233 }
3234 \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 3236 \cs_set_eq:NN \CJK@UL \CJK@@UL
3237 \tl_replace_once:Nnn \CJK@UL { \ULon }
3238 { \bool_set_true:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3239 \tl_replace_once:Nnn \CJK@@UL { \ULon }
3240 { \bool_set_false:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3241 \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: 3243 \cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_begin:
3244 { \xeCJK_cs_clear:N \UL@leadtype }
3245 \cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_end:
3246 { \cs_set_eq:NN \UL@leadtype \__xeCJK_ulem_leader_type: }
3247 \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。

```

3248 \cs_new_protected_nopar:Npn \__xeCJK_ulem_initial:
3249 {
3250     \xeCJK_swap_cs:NN
3251     \__xeCJK_Default_and_FullLeft_glue:N \__xeCJK_ulem_Default_and_FullLeft_glue:N
3252     \xeCJK_swap_cs:NN
3253     \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N
3254     \xeCJK_swap_cs:NN
3255     \__xeCJK_CJK_and_FullLeft_glue:N \__xeCJK_ulem_CJK_and_FullLeft_glue:N
3256     \xeCJK_swap_cs:NN
3257     \__xeCJK_CJK_and_FullRight_glue:N \__xeCJK_ulem_CJK_and_FullRight_glue:N
3258     \xeCJK_swap_cs:NN
3259     \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
3260     \xeCJK_swap_cs:NN \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
3261     \xeCJK_swap_cs:NN \xeCJK_FullLeft_and_Default: \__xeCJK_ulem_FullLeft_and_Default:
3262     \xeCJK_swap_cs:NN \xeCJK_FullLeft_and_CJK: \__xeCJK_ulem_FullLeft_and_CJK:
3263     \xeCJK_swap_cs:NN \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
3264     \xeCJK_swap_cs:NN \xeCJK_FullRight_and_CJK: \__xeCJK_ulem_FullRight_and_CJK:
3265     \xeCJK_swap_cs:NN \xeCJK_CJK_and_CJK:N \__xeCJK_ulem_CJK_and_CJK:N

```

```

3266 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3267 {
3268   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3269   {
3270     \str_if_eq:nnTF {##1} {####1}
3271     {
3272       \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
3273       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3274       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
3275       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3276     }
3277     {
3278       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
3279       { \__xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
3280     }
3281   }
3282 }
3283 }
(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。

```

3284 \cs_new_nopar:Npn \xeCJK_if_ulem_patch:TF
3285 {
3286   \if_meaning:w \ \LA@space
3287   \exp_after:wN \use_ii:nn
3288   \else:
3289   \exp_after:wN \use_i:nn
3290   \fi:
3291 }
(End definition for \xeCJK_if_ulem_patch:TF.)

```

`__xeCJK_ulem_CJK_and_CJK:N`

```

3292 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_CJK:N
3293 {
3294   \xeCJK_if_ulem_patch:TF
3295   {
3296     \xeCJK_class_group_end:
3297     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
3298     \__xeCJK_ulem_class_group_begin:
3299     \CJKsymbol
3300   }
3301   { \__xeCJK_ulem_CJK_and_CJK:N }
3302 }
(End definition for \__xeCJK_ulem_CJK_and_CJK:N.)

```

`_xeCJK_ulem_class_group_begin:`

```

3303 \cs_new_protected_nopar:Npn \_xeCJK_ulem_class_group_begin:
3304 {
3305   \xeCJK_class_group_begin:
3306   \xeCJK_clear_Boundary_and_CJK_toks:
3307   \xeCJK_select_font:
3308 }
(End definition for \_xeCJK_ulem_class_group_begin:.)

```

`__xeCJK_ulem_between_CJK_blocks:nnN`

```

3309 \cs_new_protected_nopar:Npn \__xeCJK_ulem_between_CJK_blocks:nnN #1#2
3310 {
3311   \xeCJK_if_ulem_patch:TF
3312   {
3313     \xeCJK_class_group_end:
3314     \UL@stop \__xeCJK_ulem_ccglue: \UL@start

```

```

3315         \xeCJK_class_group_begin:
3316         \xeCJK_clear_Boundary_and_CJK_toks:
3317         \__xeCJK_switch_font:nn {#1} {#2}
3318         \CJKsymbol
3319     }
3320     {
3321         \skip_horizontal:N \l__xeCJK_ccglue_skip
3322         \__xeCJK_switch_font:nn {#1} {#2}
3323         \CJKsymbol
3324     }
3325 }
(End definition for \__xeCJK_ulem_between_CJK_blocks:nnN.)

```

_xeCJK_ulem_Default_and_FullLeft_glue:N

```

3326 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
3327 {
3328     \xeCJK_if_ulem_patch:TF
3329     {
3330         \UL@stop
3331         \__xeCJK_ulem_skip_punct_begin:
3332         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3333         \UL@start
3334     }
3335     { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
3336 }
(End definition for \__xeCJK_ulem_Default_and_FullLeft_glue:N.)

```

_xeCJK_ulem_Boundary_and_FullLeft_glue:N

```

3337 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
3338 {
3339     \xeCJK_if_ulem_patch:TF
3340     {
3341         \UL@stop
3342         \__xeCJK_ulem_skip_punct_begin:
3343         \int_compare:nNf \etex_lastnodetype:D = \c_one
3344         { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
3345         \UL@start
3346     }
3347     { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
3348 }
(End definition for \__xeCJK_ulem_Boundary_and_FullLeft_glue:N.)

```

_xeCJK_ulem_CJK_and_FullLeft_glue:N

```

3349 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
3350 {
3351     \xeCJK_if_ulem_patch:TF
3352     {
3353         \xeCJK_class_group_end:
3354         \UL@stop
3355         \__xeCJK_ulem_skip_punct_begin:
3356         \__xeCJK_ulem_ccglue:
3357         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3358         \UL@start
3359         \__xeCJK_ulem_class_group_begin:
3360     }
3361     { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
3362 }
(End definition for \__xeCJK_ulem_CJK_and_FullLeft_glue:N.)

```

_xeCJK_ulem_Default_and_FullRight_glue:N

```

3363 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
3364 {

```

```

3365 \xeCJK_if_ulem_patch:TF
3366 {
3367     \UL@stop
3368     \__xeCJK_ulem_skip_punct_begin:
3369     \__xeCJK_punct_if_long:NTF {#1}
3370     { \__xeCJK_ulem_ccglue: }
3371     {
3372         \__xeCJK_punct_if_middle:NT {#1}
3373         {
3374             \xeCJK_no_break:
3375             \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
3376         }
3377         \xeCJK_no_break:
3378     }
3379     \UL@start
3380 }
3381 { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
3382 }
(End definition for \__xeCJK_ulem_Default_and_FullRight_glue:N)

```

_xeCJK_ulem_CJK_and_FullRight_glue:N

```

3383 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
3384 {
3385     \xeCJK_if_ulem_patch:TF
3386     {
3387         \xeCJK_class_group_end:
3388         \__xeCJK_Default_and_FullRight_glue:N {#1}
3389         \__xeCJK_ulem_class_group_begin:
3390     }
3391     { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
3392 }
(End definition for \__xeCJK_ulem_CJK_and_FullRight_glue:N)

```

_xeCJK_ulem_FullLeft_and_Default:

```

3393 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_Default:
3394 {
3395     \xeCJK_if_ulem_patch:TF
3396     {
3397         \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3398         {
3399             \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3400             \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3401             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
3402             \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3403         }
3404         { \xeCJK_class_group_end: \UL@stop }
3405         \__xeCJK_ulem_skip_punct_end:
3406         \xeCJK_no_break:
3407         \UL@start
3408     }
3409     { \__xeCJK_ulem_FullLeft_and_Default: }
3410 }
(End definition for \__xeCJK_ulem_FullLeft_and_Default:.)

```

_xeCJK_ulem_FullLeft_and_CJK:

```

3411 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_CJK:
3412 {
3413     \xeCJK_if_ulem_patch:TF
3414     {
3415         \xeCJK_FullLeft_and_Default:
3416         \__xeCJK_ulem_class_group_begin:
3417     }

```

```

3418     { \__xeCJK_ulem_FullLeft_and_CJK: }
3419   }

```

(End definition for __xeCJK_ulem_FullLeft_and_CJK:.)

__xeCJK_ulem_FullRight_and_Default:

```

3420 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_Default:
3421 {
3422   \xeCJK_if_ulem_patch:TF
3423   {
3424     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3425     \xeCJK_class_group_end:
3426     \UL@stop
3427     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3428     \__xeCJK_ulem_skip_punct_end:
3429     \UL@start
3430   }
3431   { \__xeCJK_ulem_FullRight_and_Default: }
3432 }

```

(End definition for __xeCJK_ulem_FullRight_and_Default:.)

__xeCJK_ulem_FullRight_and_CJK:

```

3433 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_CJK:
3434 {
3435   \xeCJK_if_ulem_patch:TF
3436   {
3437     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3438     \xeCJK_class_group_end:
3439     \UL@stop
3440     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3441     \__xeCJK_ulem_ccglue:
3442     \__xeCJK_ulem_skip_punct_end:
3443     \UL@start
3444     \__xeCJK_ulem_class_group_begin:
3445   }
3446   { \__xeCJK_ulem_FullRight_and_CJK: }
3447 }

```

(End definition for __xeCJK_ulem_FullRight_and_CJK:.)

__xeCJK_ulem_punct_hskip:n

```

3448 \cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_hskip:n
3449 {
3450   \xeCJK_if_ulem_patch:TF
3451   { \__xeCJK_ulem_hskip:n }
3452   { \__xeCJK_ulem_punct_hskip:n }
3453 }

```

(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 3454 \cs_new_protected_nopar:Npn __xeCJK_ulem_glue:n #1

```

3455 {
3456   \xeCJK_if_ulem_patch:TF
3457   { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
3458   { \skip_horizontal:n {#1} }
3459 }
3460 \cs_new_protected_nopar:Npn \__xeCJK_ulem_ccglue:
3461 { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders }
3462 \cs_new_protected_nopar:Npn \__xeCJK_ulem_hskip:n #1
3463 {
3464   \int_compare:nNnTF \tex_lastkern:D = \c_three
3465   { \skip_horizontal:n {#1} }
3466   { \skip_set:Nn \UL@skip {#1} \UL@leaders }
3467 }

```

(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。

```

3468 \RenewDocumentCommand \CJKUnderdot { m }
3469 {
3470   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3471   \CJK@preUnderdot
3472   \__xeCJK_make_under_symbol:n { \CJK@underdotSkip }
3473   \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3474   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
3475   \__xeCJK_restore_output_CJKsymbol:
3476   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3477   #1
3478   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3479   \cs_set_eq:NN \CJKsymbol \__xeCJK_save_under_dot_CJKsymbol:N
3480   \tex_output:D \exp_after:wN { \l__xeCJK_underdot_output_tl }
3481   \CJK@postUnderdot
3482   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3483   \tex_ignorespaces:D
3484 }
3485 \box_new:N \g__xeCJK_under_symbol_box
(End definition for \CJKUnderdot.)

```

\CJKUnderanysymbol

```

3486 \RenewDocumentCommand \CJKUnderanysymbol { m m m }
3487 {
3488   \group_begin:
3489   \hbox_set:Nn \CJK@underdotBox {#2}
3490   \__xeCJK_make_under_symbol:n {#1}
3491   \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3492   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
3493   \__xeCJK_restore_output_CJKsymbol:
3494   #3
3495   \group_end:
3496   \tex_ignorespaces:D
3497 }
(End definition for \CJKUnderanysymbol.)

```

__xeCJK_restore_output_CJKsymbol: \CJKUnderdot 中对 \CJKsymbol 的修改会影响到页眉和页脚, 需要小心处理。

```

3498 \cs_new_protected:Npn \__xeCJK_restore_output_CJKsymbol:
3499 {
3500   \tl_set:Nx \l__xeCJK_underdot_output_tl
3501   { \exp_after:wN \exp_not:n \tex_the:D \tex_output:D }
3502   \tex_output:D \exp_after:wN
3503   {
3504     \exp_after:wN \cs_set_eq:NN
3505     \exp_after:wN \CJKsymbol
3506     \exp_after:wN \__xeCJK_save_under_dot_CJKsymbol:N
3507     \l__xeCJK_underdot_output_tl
3508   }
3509 }
3510 \tl_new:N \l__xeCJK_underdot_output_tl
(End definition for \__xeCJK_restore_output_CJKsymbol:.)

```

__xeCJK_make_under_symbol:n

```

3511 \cs_new_protected:Npn \__xeCJK_make_under_symbol:n #1
3512 {
3513   \hbox_set:Nn \l__xeCJK_tmp_box { ^^^^4e00 }
3514   \vbox_gset_to_ht:Nnn \g__xeCJK_under_symbol_box \c_zero_dim
3515   {
3516     \skip_vertical:n {#1}

```

```

3517         \hbox_to_zero:n
3518         {
3519             \tex_kern:D - \box_wd:N \l__xeCJK_tmp_box
3520             \tex_hss:D \box_use:N \CJK@underdotBox \tex_hss:D
3521         }
3522         \tex_vss:D
3523     }
3524 }
(End definition for \__xeCJK_make_under_symbol:n.)

```

__xeCJK_under_CJKsymbol:N

```

3525 \cs_new_protected_nopar:Npn \__xeCJK_under_CJKsymbol:N #1
3526 {
3527     \__xeCJK_save_under_dot_CJKsymbol:N {#1}
3528     \hbox_overlap_left:n { \box_use:N \g__xeCJK_under_symbol_box }
3529     { \xeCJK_make_node:n { CJK } }
3530     \xeCJK_ignore_spaces:w
3531 }
(End definition for \__xeCJK_under_CJKsymbol:N.)

```

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

```

3532 \RenewDocumentEnvironment { CJKfilltwosides } { m }
3533 {
3534     \dim_set:Nn \l__xeCJK_fill_width_dim {#1}
3535     \cs_set_eq:NN \CJKglue \tex_hfil:D
3536     \cs_set_eq:NN \ \__xeCJK_fill_newline:
3537     \mode_if_vertical:T { \tex_indent:D }
3538     \tex_vbox:D \c_group_begin_token
3539     \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
3540     \tex_ignorespaces:D
3541 }
3542 {
3543     \tex_unskip:D
3544     \c_group_end_token
3545     \c_group_end_token
3546 }
3547 \dim_new:N \l__xeCJK_fill_width_dim
(End definition for CJKfilltwosides.)

```

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

```

3548 \cs_new_protected_nopar:Npn \__xeCJK_fill_newline:
3549 {
3550     \tex_unskip:D
3551     \c_group_end_token
3552     \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
3553     \tex_ignorespaces:D
3554 }
(End definition for \__xeCJK_fill_newline:.)
3555 </fntef>

```

5.20 xeCJK-listings

仿照 [luatexja](#) 宏包中 `lltjp-listings` 的处理,支持 `listings` 宏包。

```

3556 <*listings>
3557 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
3558 \ProcessOptions \scan_stop:
3559 \RequirePackage { xeCJK }
3560 \RequirePackage { listings }

```

```

3561 \lst@AddToHook { Init } { \__xeCJK_listings_initial_hook: }
3562 \lst@AddToHook { SelectCharTable } { \__xeCJK_listings_toks_hook: }
3563 \lst@AddToHook { OutputBox }
3564 {
3565     \l__xeCJK_restore_listings_toks_tl
3566     \__xeCJK_listings_output_IVS:
3567 }
3568 \lst@AddToHook { PreSet } { \bool_set_true:N \l__xeCJK_listings_env_bool }

```

__xeCJK_listings_initial_hook: 为使代码行号结果正确,需要在 \lst@numberstyle 中恢复 \XeTeXinterchartoks。在 listings 环境中换页时,对 \XeTeXinterchartoks 的修改会影响到页眉和页脚,需要在 \output 中恢复成正常定义。这里使用 \use:n 是为了在 \tex_output:D 中不增加额外的分组。加入 \tex_noindent:D 是为了防止汉字出现在首行的时候可能会产生额外空行。

```

3569 \cs_new_protected_nopar:Npn \__xeCJK_listings_initial_hook:
3570 {
3571     \tex_noindent:D
3572     \bool_gset_false:N \g__xeCJK_listings_IVS_bool
3573     \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
3574     \tex_output:D \exp_after:wN \exp_after:wN \exp_after:wN
3575     {
3576         \exp_after:wN \exp_after:wN
3577         \exp_after:wN \l__xeCJK_restore_listings_toks_tl
3578         \exp_after:wN \use:n
3579         \tex_the:D \tex_output:D
3580     }
3581 }

```

(End definition for __xeCJK_listings_initial_hook:.)

__xeCJK_listings_toks_hook: 采用不同的 \XeTeXinterchartoks 处理方式,输入的时候是将汉字加入到 listings 的输出队列,实际输出的时候是普通文字。

```

3582 \cs_new_protected_nopar:Npn \__xeCJK_listings_toks_hook:
3583 {
3584     \tl_set:Nx \l__xeCJK_restore_listings_toks_tl
3585     {
3586         \__xeCJK_backup_inter_class_toks:nn { Boundary } { Default }
3587         \__xeCJK_backup_inter_class_toks:nn { Boundary } { CJK }
3588         \__xeCJK_backup_inter_class_toks:nn { Boundary } { IVS }
3589         \__xeCJK_backup_inter_class_toks:nn { Boundary } { FullLeft }
3590         \__xeCJK_backup_inter_class_toks:nn { Boundary } { FullRight }
3591     }
3592     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3593     {
3594         \tl_put_right:Nx \l__xeCJK_restore_listings_toks_tl
3595         { \__xeCJK_backup_inter_class_toks:nn { Boundary } { CJK/##1 } }
3596     }
3597     \xeCJK_inter_class_toks:nnn { Boundary } { Default }
3598     { \__xeCJK_listings_process_Default:N }
3599     \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
3600     { \__xeCJK_listings_process_CJK:nN { \c_two } }
3601     \xeCJK_inter_class_toks:nnn { Boundary } { IVS }
3602     { \__xeCJK_listings_process_IVS:nN { \c_zero } }
3603     \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
3604     { \__xeCJK_listings_process_CJK:nN { \c_two } }
3605     \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
3606     { \__xeCJK_listings_process_CJK:nN { \c_two } }
3607     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3608     {
3609         \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
3610         { \__xeCJK_listings_process_CJK:nN { \c_two } }
3611     }
3612 }
3613 \tl_new:N \l__xeCJK_restore_listings_toks_tl

```



```

3614 \cs_new_nopar:Npn \__xeCJK_backup_inter_class_toks:nn #1#2
3615 {
3616   \xeCJK_inter_class_toks:nnn {#1} {#2}
3617   { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
3618 }
(End definition for \__xeCJK_listings_toks_hook:.)

```

_xeCJK_listings_process_Default:N 对于 \charcode 大于 255 的字符,根据 \catcode 进行处理。

```

\__xeCJK_listings_process_CJK:nN
3619 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_Default:N #1
3620 {
3621   \token_if_letter:NTF #1
3622   { \lst@ProcessLetter #1 }
3623   { \lst@ProcessOther #1 }
3624 }
3625 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_CJK:nN #1#2
3626 {
3627   \token_if_letter:NTF #2
3628   { \__xeCJK_listings_process_letter:nN {#1} #2 }
3629   { \__xeCJK_listings_process_other:nN {#1} #2 }
3630 }
(End definition for \__xeCJK_listings_process_Default:N and \__xeCJK_listings_process_CJK:nN.)

```

_xeCJK_listings_append:nN 普通 CJK 字符的宽度为一般基本宽度的两倍,IVS 类不增加宽度。这里有一个问题,对 CJK 字符类中的一些半角字符(例如半角日文假名)没有区分开。listings 通过重定义 \lst@Append 将代码写入外部文件,因此需要保留。

```

3631 \cs_new_protected_nopar:Npn \__xeCJK_listings_append:nN #1#2
3632 {
3633   \int_add:Nn \lst@length { #1 - \c_one }
3634   \lst@Append #2
3635 }
(End definition for \__xeCJK_listings_append:nN.)

```

_xeCJK_listings_process_letter:nN 在 letter 类中区分汉字和西文字母。

```

\__xeCJK_listings_process_other:nN
3636 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_letter:nN
3637 {
3638   \lst@whitespacefalse
3639   \bool_if:NTF \l__xeCJK_listings_letter_bool
3640   { \lst@lettertrue }
3641   {
3642     \lst@ifletter
3643     \lst@Output
3644     \else:
3645     \lst@OutputOther
3646     \lst@lettertrue
3647     \fi:
3648     \bool_set_true:N \l__xeCJK_listings_letter_bool
3649   }
3650   \__xeCJK_listings_append:nN
3651 }
3652 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_other:nN #1#2
3653 {
3654   \lst@whitespacefalse
3655   \bool_if:NTF \l__xeCJK_listings_letter_bool
3656   {
3657     \lst@Output
3658     \bool_set_false:N \l__xeCJK_listings_letter_bool
3659     \lst@letterfalse
3660   }
3661   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
3662   \cs_set_eq:NN \lst@lastother #2
3663   \__xeCJK_listings_append:nN {#1} #2
3664 }

```

```

3665 \bool_new:N \l__xeCJK_listings_letter_bool
3666 \cs_set_protected_nopar:Npn \lst@AppendLetter
3667 {
3668   \bool_if:NTF \l__xeCJK_listings_letter_bool
3669   {
3670     \lst@Output
3671     \bool_set_false:N \l__xeCJK_listings_letter_bool
3672     \lst@lettertrue
3673   }
3674   { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
3675   \lst@Append
3676 }
3677 \cs_set_protected_nopar:Npn \lst@AppendOther
3678 {
3679   \bool_if:NTF \l__xeCJK_listings_letter_bool
3680   {
3681     \lst@Output
3682     \bool_set_false:N \l__xeCJK_listings_letter_bool
3683     \lst@letterfalse
3684   }
3685   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
3686   \tex_futurelet:D \lst@lastother \lst@Append
3687 }
(End definition for \__xeCJK_listings_process_letter:nN and \__xeCJK_listings_process_other:nN.)

```

`__xeCJK_listings_process_IVS:nN` IVS 类作为 letter 处理,不用增加 `\lst@length`。

```

3688 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_IVS:nN
3689 {
3690   \reverse_if:N \lst@ifflexible
3691   \bool_gset_true:N \g__xeCJK_listings_IVS_bool
3692   \fi:
3693   \__xeCJK_listings_process_letter:nN
3694 }
(End definition for \__xeCJK_listings_process_IVS:nN.)

```

`__xeCJK_listings_output_IVS:` 在使用 `columns=fixed` 选项时, `listings` 会在输出盒子里的每个字符之间加入 `\hss`,这就破坏了 \XeTeX 将基本字和 IVS 正确的组合起来。

```

3695 \cs_new_protected_nopar:Npn \__xeCJK_listings_output_IVS:
3696 {
3697   \reverse_if:N \lst@ifflexible
3698   \bool_if:NT \g__xeCJK_listings_IVS_bool
3699   {
3700     \bool_gset_false:N \g__xeCJK_listings_IVS_bool
3701     \xeCJK_cs_clear:N \lst@FillOutputBox
3702     \cs_set_eq:NN \CJKglue \tex_hss:D
3703   }
3704   \fi:
3705 }
3706 \bool_new:N \g__xeCJK_listings_IVS_bool
(End definition for \__xeCJK_listings_output_IVS:.)

```

`__xeCJK_listings_peek_active_loop:TF` `\lstinline` 通过判断参数中第一个字符是否是 active 类来区分它是否被用在其它宏的参数之中。如果这第一个字符不在 `listings` 预定义的符号表中,判断就会出问题。我们在这里通过一个循环跳过这些字符。

```

3707 \cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
3708 {
3709   \token_if_active:NTF #3
3710   { #1#3 }
3711   {
3712     \token_if_cs:NTF #3
3713     { #2#3 }
3714     {

```

```

3715         \int_compare:nNnTF { '#3 } > { \lst@ifec 255 \else: 127 \fi: }
3716         { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
3717         { #2#3 }
3718     }
3719 }
3720 }
3721 \cs_set_eq:NN \lst@ifNextCharActive \__xeCJK_listings_peek_active_loop:TF
(End definition for \__xeCJK_listings_peek_active_loop:TF.)

```

__xeCJK_listings_inside_convert:nw 当 \lstinline 被使用在参数中时, listings 会使用一个循环逐个将 \lstinline 参数中的字符设置为活动字符。我们可以通过 \tl_set_rescan:Nnn 来完成这里的 \catcode 转换, 避免将 \charcode 超过 255 的字符都设置为活动字符。

```

3722 \cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
3723 {
3724     \tl_set_rescan:Nnn \l__xeCJK_tmpa_tl { } {#1}
3725     \__xeCJK_set_listings_escape:
3726     \tl_put_right:NV \lst@arg \l__xeCJK_tmpa_tl
3727 }
3728 \cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
3729 \cs_new_protected_nopar:Npn \__xeCJK_listings_inline_group:w
3730 {
3731     \exp_after:wN \__xeCJK_listings_inline_group:n
3732     \exp_after:wN { \if_false: } \fi:
3733 }
3734 \cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
3735 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
3736 {
3737     \tl_set_rescan:Nnn \lst@arg { } {#1}
3738     \__xeCJK_set_listings_escape:
3739     \lst@InlineGJEnd
3740 }
(End definition for \__xeCJK_listings_inside_convert:nw and \__xeCJK_listings_inline_group:w.)

```

__xeCJK_set_listings_escape: 由于我们在上面的修改, 需要保留 \ 用于转义 \lstinline 参数中的某些 T_EX 特殊字符, 与原来宏包一致。

```

3741 \group_begin:
3742 \char_set_catcode_active:N \/
3743 \char_set_lccode:nn { '\ } { '\ }
3744 \tl_to_lowercase:n
3745 {
3746     \group_end:
3747     \cs_new_protected:Npn \__xeCJK_set_listings_escape:
3748     { \xeCJK_swap_cs:NN / \__xeCJK_listings_escape:N }
3749     \cs_new_protected:Npn \__xeCJK_listings_escape:N #1
3750     { \cs_if_eq:NNTF / #1 { \__xeCJK_listings_escape:N } {#1} }
3751 }
(End definition for \__xeCJK_set_listings_escape:.)
3752 </listings>

```

5.21 xeCJK.cfg

```

3753 <*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。

```

3754
3755 </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。.....	44
删除多余的 <code>default-itscorr</code> 结点。.....	29
取消 <code>\cprotect</code> 的外部宏限制。.....	78
改用 <code>\indentfirst</code> 宏包处理缩进的问题。.....	72
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。..	68
放弃对 <code>\outer</code> 宏的特殊处理。.....	1
<code>__xeCJK_switch_font:nn</code> : 改进定义,加快切换速度。...	61
<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。.....	71
<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。.....	75
<code>\xeCJK_fallback_loop:Nn</code> : 调整备用字体的循环方式。..	52
<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。..	79
<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。.....	70
<code>\xeCJKOffVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。.....	68

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。.....	39
增加小宏包 <code>xeCJKfntef</code> ,用于处理下划线的问题。.....	79
对于与 <code>xltxtra</code> 的冲突给出错误警告。.....	75
<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。.....	38
<code>__xeCJK_set_char_class_eq:nn</code> : 交换参数的顺序。.....	25
<code>__xeCJK_set_verb_exspace::</code> : 调整间距的计算方法。...	69
<code>\fontfamily</code> : 修改主要 CJK 字体族的自动更新方式。....	75
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。.....	35
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。.....	40
<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。.....	39
<code>\xeCJK_check_single:NNw</code> : 改进定义,减少使用 <code>peek</code> 函数的次数。.....	37
<code>\xeCJK_fallback_loop:Nn</code> : 修正备用字体的不正确警告。..	52
<code>\xeCJK_hook_for_ulem::</code> : 完全处理下划线里的标点符号的有关问题。.....	79
<code>\xeCJK_peek_catcode_ignore_spaces:NTF</code> : 新增有省略空格标识的 <code>peek</code> 函数。.....	16
<code>\xeCJK_save_class:nn</code> : 使用 <code>\xeCJK_save_class:nn</code> 保存 \XeTeX 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> 。.....	24
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。.....	36

v3.1.2

General: 修正重定义 <code>\CJKfamilydefault</code> 无效的问题,恢复容错能力。.....	66
--	----

解决在下划线状态下使用 <code>\makebox</code> 时的错误。.....	81
<code>__xeCJK_check_single_space:NN</code> : 使用 <code>\xeCJK_if_CJK_class:NTF</code> 来代替 <code>\int_case:nnn</code> 判断是否是 CJK 字符类。.....	38
<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。.....	63
<code>\fontfamily</code> : 不将参数完全展开。.....	75
<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。.....	74
<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。.....	68
<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> 。.....	63
<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。.....	64
<code>\xeCJK_Boundary_and_FullLeft:N</code> : 当全角左标点前面是 <code>hlist:none\glue</code> 和 <code>penalty</code> 等节点时,压缩其左空白。..	33
<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。.....	32

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> 。.....	68
--	----

v3.2.2

General: 增加小宏包 <code>xeCJK-listings</code> ,用于支持 <code>listings</code> 宏包。.....	86
<code>__xeCJK_ulem_CJK_and_FullRight_glue:N</code> : 修正下划线不能跳过全角右标点的问题。.....	83
<code>\xeCJK@text@composite</code> : 修正某些重音不能正确显示的问题。.....	74

v3.2.3

General: 不再改变 CJK 字符类的 <code>\catcode</code> 。.....	24
完善对 <code>listings</code> 宏包的支持。.....	86
提供四个 <code>TECkit</code> 映射文件用于句号转换和简繁互换。..	1
根据 \XeTeX 的脚本重新整理全角标点符号。.....	19
解决 <code>CheckSingle</code> 选项与 <code>tablists</code> 宏包的冲突。.....	38
<code>__xeCJK_listings_initial_hook::</code> : 解决 <code>listings</code> 环境中代码行号输出不正确的问题,并解决在其中跨页时对页眉和页脚的影响。.....	87
<code>__xeCJK_listings_process_CJK:nN</code> : 在 <code>listings</code> 环境中对 <code>\charcode</code> 大于 255 的字符根据其 <code>\catcode</code> 区分 <code>letter</code> 和 <code>other</code> 。.....	88
<code>__xeCJK_restore_output_CJKsymbol::</code> : 解决 <code>\CJKunderdot</code> 跨页使用时影响到页眉页脚的问题。.....	85
<code>__xeCJK_ulem_FullLeft_and_CJK::</code> : 修正全角左标点下划线与 <code>\CJKunderdot</code> 连用时结果不正常的问题。.....	83
<code>\xeCJKOffVerbAddon</code> : 新增 <code>\xeCJKOffVerbAddon</code> 用于局部取消 <code>\xeCJKOffVerbAddon</code> 的影响;并解决跨页使用时影响到页眉页脚的问题。.....	68

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

\/	3742, 3743	__xeCJK_error:n	36, 37, 1310
\@italiccorr	3013, 3021, 3027, 3033, 3036, 3047	__xeCJK_error:nx	36, 38, 274, 285, 1246, 1780, 1808, 2845, 2971
\@empty	2949, 2953, 2956, 3722	__xeCJK_fallback_save_CJKsymbol:N	1842, 1849, 1857, 1866, 1872
\@ifpackagelater	19, 22, 2969, 2985	__xeCJK_family_csname:n	2098, 2271, 2348, 2356, 2356, 2374
\@onefilewithoptions	3189	__xeCJK_family_default_wrap:n	2547, 2561, 2903, 2906
\@onlypreamble	1216, 1803, 1810, 2463, 2464, 2465, 2466, 2467, 2497	__xeCJK_family_nfss_csname:n	2114, 2299, 2300, 2356, 2357, 2358, 2362, 2372
\@pkgextension	51	__xeCJK_family_unknown_warning:n	2408, 2408, 2419
\@text@composite	2947, 2950, 2956, 2959, 2960	__xeCJK_family_unknown_warning:x	2393, 2406
\[2874	__xeCJK_family_use:x	2312, 2321, 2352, 2356, 2358, 2391, 2405
\\	5, 6, 7, 13, 14, 15, 16, 296, 297, 1252, 1304, 1305, 1786, 1787, 1788, 1800, 1801, 1879, 2423, 2524, 2587, 2588, 2589, 2590, 2595, 2851, 2895, 2896, 2963, 2964, 3536, 3743	__xeCJK_fill_newline:	3536, 3548, 3548
__xeCJK_Boundary_and_FullLeft_glue:N	849, 857, 3259	__xeCJK_font_csname:n	2305, 2306, 2318, 2320, 2339, 2342
__xeCJK_CJK_and_FullLeft_glue:N	839, 844, 3255	__xeCJK_get_map_font:n	2203, 2205, 2208, 2208, 2212
__xeCJK_CJK_and_FullRight_glue:N	896, 902, 902, 912, 3257	__xeCJK_get_sub_features:Vn	1900
__xeCJK_CJK_class_tl:c	413	__xeCJK_get_sub_features:nn	1963, 1970, 1970, 1997
__xeCJK_CJK_class_tl:n	247, 250, 252	__xeCJK_get_sub_features:wn	1970, 1976, 1990
__xeCJK_Default_and_FullLeft_glue:N	825, 834, 3251	__xeCJK_gobble_CJKfamily:	2399, 2399, 3124
__xeCJK_Default_and_FullRight_glue:N	871, 884, 902, 912, 3253, 3388	__xeCJK_gobble_CJKfamily:wn	2400, 2401
__xeCJK_add_fake_bold:n	2127, 2127, 2176, 2195	__xeCJK_group_begin:	3087, 3087, 3098, 3126
__xeCJK_add_fake_slant:n	2127, 2132, 2187	__xeCJK_group_end:	3087, 3088, 3098, 3127
__xeCJK_add_font:nV	2153, 2155	__xeCJK_gset_family_cs:x	2073, 2096, 2096
__xeCJK_add_font:nn	2121, 2121, 2125	__xeCJK_gset_family_nfss_cs:xx	2105, 2278, 2356, 2359
__xeCJK_add_font:nv	2143	__xeCJK_gset_punct_dim:nnnx	1389, 1403, 1734
__xeCJK_add_font:nx	2203	__xeCJK_gset_punct_dim:nnx	1389, 1401, 1505, 1580, 1700, 1701, 1702, 1707, 1717, 1719, 1758
__xeCJK_add_font_if_new:nn	2121, 2123, 2126, 2197	__xeCJK_gset_punct_dim:nnx	1389, 1399, 1726, 1728
__xeCJK_add_font_if_new:nv	2147	__xeCJK_if_font_select:n	2161
__xeCJK_add_font_if_new:nx	2205	__xeCJK_if_font_select:nF	2146, 2152, 2154, 2204
__xeCJK_add_special_punct:nn	1329, 1332, 1335, 1339, 1356	__xeCJK_if_font_select:nTF	2141, 2161
__xeCJK_after_end_preamble:n	52, 59, 1311, 3050, 3118, 3131	__xeCJK_if_font_select:p:n	2161, 2183, 2185, 2193
__xeCJK_after_preamble:n	52, 57, 2649, 3089	__xeCJK_if_it_or_sl:n	2166
__xeCJK_at_end_preamble:n	52, 55, 2545, 2981, 3156, 3169	__xeCJK_if_it_or_sl:nT	2189
__xeCJK_aux_r:n	3101, 3109, 3110	__xeCJK_if_it_or_sl:nTF	2144, 2161
__xeCJK_backup_inter_class_toks:nn	3586, 3587, 3588, 3589, 3590, 3595, 3614	__xeCJK_if_it_or_sl_p:n	2161, 2181, 2192
__xeCJK_block_select_font:n	2333, 2337, 2337	__xeCJK_if_map_font_select:n	2210
__xeCJK_calc_kerning_margin:NN	1592, 1612, 1612	__xeCJK_if_map_font_select:nT	2201
__xeCJK_calc_kerning_margin:NNN	1625, 1634, 1679, 1679	__xeCJK_if_map_font_select:nTF	2210
__xeCJK_check_family:V	2072, 2257	__xeCJK_if_map_font_select:p:n	2210
__xeCJK_check_family:n	2109, 2109, 2118	__xeCJK_info:nx	36, 42
__xeCJK_check_num_range:nnNN	441, 441, 455, 2632	__xeCJK_info:nxx	36, 43, 2330
__xeCJK_check_single_save:N	1041, 1048, 1069, 1079, 1088, 1112, 1113, 1124, 1125, 1127, 1133, 1136, 1146, 1149	__xeCJK_int_until_do:wn	182, 183, 184, 185
__xeCJK_check_single_space:NN	1078, 1087, 1119, 1119	__xeCJK_kerning_width_or_ratio:NNN	1630, 1631, 1649, 1649
__xeCJK_class_csname:n	273, 276, 279, 284, 287, 292, 292, 293, 413, 416, 1240, 1258, 3064	__xeCJK_listings_append:nN	3631, 3631, 3650, 3663
__xeCJK_clear_Boundary_and_CJK_toks:n	524, 525	__xeCJK_listings_escape:N	3748, 3749, 3750
__xeCJK_copy_family:nn	2284, 2284, 2303	__xeCJK_listings_initial_hook:	3561, 3569, 3569
__xeCJK_copy_family:xx	2345	__xeCJK_listings_inline_group:n	3731, 3735
__xeCJK_copy_sub_family:n	2247, 2255	__xeCJK_listings_inline_group:w	3722, 3729, 3734
__xeCJK_dim_max:nn	2979, 2987, 2991	__xeCJK_listings_inside_convert:nw	3722, 3722, 3728
__xeCJK_dim_min:nn	2980, 2988, 2992	__xeCJK_listings_output_IVS:	3566, 3695, 3695
__xeCJK_document_left_hook:	69, 75	__xeCJK_listings_peek_active_loop:TF	3707, 3707, 3716, 3721
__xeCJK_document_right_hook:	71, 77	__xeCJK_listings_process_CJK:nN	3600, 3604, 3606, 3610, 3619, 3625
__xeCJK_error:n	36, 37, 1310	__xeCJK_listings_process_Default:N	3598, 3619, 3619
__xeCJK_error:nx	36, 38, 274, 285, 1246, 1780, 1808, 2845, 2971	__xeCJK_listings_process_IVS:nN	3602, 3688, 3688
__xeCJK_fallback_save_CJKsymbol:N	1842, 1849, 1857, 1866, 1872	__xeCJK_listings_process_letter:nN	3628, 3636, 3636, 3693
__xeCJK_family_csname:n	2098, 2271, 2348, 2356, 2356, 2374		
__xeCJK_family_default_wrap:n	2547, 2561, 2903, 2906		
__xeCJK_family_nfss_csname:n	2114, 2299, 2300, 2356, 2357, 2358, 2362, 2372		
__xeCJK_family_unknown_warning:n	2408, 2408, 2419		
__xeCJK_family_unknown_warning:x	2393, 2406		
__xeCJK_family_use:x	2312, 2321, 2352, 2356, 2358, 2391, 2405		
__xeCJK_fill_newline:	3536, 3548, 3548		
__xeCJK_font_csname:n	2305, 2306, 2318, 2320, 2339, 2342		
__xeCJK_get_map_font:n	2203, 2205, 2208, 2208, 2212		
__xeCJK_get_sub_features:Vn	1900		
__xeCJK_get_sub_features:nn	1963, 1970, 1970, 1997		
__xeCJK_get_sub_features:wn	1970, 1976, 1990		
__xeCJK_gobble_CJKfamily:	2399, 2399, 3124		
__xeCJK_gobble_CJKfamily:wn	2400, 2401		
__xeCJK_group_begin:	3087, 3087, 3098, 3126		
__xeCJK_group_end:	3087, 3088, 3098, 3127		
__xeCJK_gset_family_cs:x	2073, 2096, 2096		
__xeCJK_gset_family_nfss_cs:xx	2105, 2278, 2356, 2359		
__xeCJK_gset_punct_dim:nnnx	1389, 1403, 1734		
__xeCJK_gset_punct_dim:nnx	1389, 1401, 1505, 1580, 1700, 1701, 1702, 1707, 1717, 1719, 1758		
__xeCJK_gset_punct_dim:nnx	1389, 1399, 1726, 1728		
__xeCJK_if_font_select:n	2161		
__xeCJK_if_font_select:nF	2146, 2152, 2154, 2204		
__xeCJK_if_font_select:nTF	2141, 2161		
__xeCJK_if_font_select:p:n	2161, 2183, 2185, 2193		
__xeCJK_if_it_or_sl:n	2166		
__xeCJK_if_it_or_sl:nT	2189		
__xeCJK_if_it_or_sl:nTF	2144, 2161		
__xeCJK_if_it_or_sl_p:n	2161, 2181, 2192		
__xeCJK_if_map_font_select:n	2210		
__xeCJK_if_map_font_select:nT	2201		
__xeCJK_if_map_font_select:nTF	2210		
__xeCJK_if_map_font_select:p:n	2210		
__xeCJK_info:nx	36, 42		
__xeCJK_info:nxx	36, 43, 2330		
__xeCJK_int_until_do:wn	182, 183, 184, 185		
__xeCJK_kerning_width_or_ratio:NNN	1630, 1631, 1649, 1649		
__xeCJK_listings_append:nN	3631, 3631, 3650, 3663		
__xeCJK_listings_escape:N	3748, 3749, 3750		
__xeCJK_listings_initial_hook:	3561, 3569, 3569		
__xeCJK_listings_inline_group:n	3731, 3735		
__xeCJK_listings_inline_group:w	3722, 3729, 3734		
__xeCJK_listings_inside_convert:nw	3722, 3722, 3728		
__xeCJK_listings_output_IVS:	3566, 3695, 3695		
__xeCJK_listings_peek_active_loop:TF	3707, 3707, 3716, 3721		
__xeCJK_listings_process_CJK:nN	3600, 3604, 3606, 3610, 3619, 3625		
__xeCJK_listings_process_Default:N	3598, 3619, 3619		
__xeCJK_listings_process_IVS:nN	3602, 3688, 3688		
__xeCJK_listings_process_letter:nN	3628, 3636, 3636, 3693		

_xeCJK_listings_process_other:nN	3629, 3636, 3652	_xeCJK_set_fake_font:nn	2159, 2173, 2173
_xeCJK_listings_toks_hook:	3562, 3582, 3582	_xeCJK_set_family_initial:	2036, 2036, 2061
_xeCJK_make_under_symbol:n	3472, 3490, 3511, 3511	_xeCJK_set_listings_escape:	3725, 3738, 3741, 3747
_xeCJK_map_it_sl:n	2199	_xeCJK_set_others_toks:n	3050, 3058, 3062
_xeCJK_map_it_sl:nn	2189, 2199	_xeCJK_set_special_punct:nn	1328, 1331, 1334, 1339, 1345
_xeCJK_margin_width_or_ratio:n	1546, 1547, 1567, 1567	_xeCJK_set_sub_block_family:	2075, 2242, 2242
_xeCJK_msg_def_family_map:n	2424, 2426, 2589, 2596	_xeCJK_set_sub_class_toks:nn	1261, 1267, 1267
_xeCJK_msg_family_map:n	2120, 2423, 2437, 2594, 2595	_xeCJK_set_verb_exspace:	2675, 2740, 2740
_xeCJK_msg_new:nn	36, 36, 294, 1250, 1302, 1784, 1798, 1875, 2119, 2336, 2421, 2522, 2585, 2592, 2849, 2893, 2961	_xeCJK_set_visible_space_size:n	2778, 2782
_xeCJK_original_kerning_margin:NN	1593, 1600, 1600, 1663	_xeCJK_space_or_xecglue:	575, 687, 692, 699, 722, 735
_xeCJK_parse_features:	2070, 2137, 2137	_xeCJK_space_skip_scale:nnn	127, 131, 144, 148, 158
_xeCJK_parse_features:nn	2137, 2138, 2139	_xeCJK_special_punct_seq:n	1341, 1344, 1347, 1349, 1353, 1360, 1363, 1372
_xeCJK_parse_sub_class_features:n	2510, 2527, 2527	_xeCJK_special_punct_tl:nN	1342, 1348, 1352, 1362, 1371, 1385
_xeCJK_pass_features:	2071, 2215, 2215	_xeCJK_sub_restore_or_cancel:n	1236, 1236, 1249
_xeCJK_peek_after_do:w	216, 227	_xeCJK_sub_restore_or_cancel:x	1223, 1232
_xeCJK_peek_catcode_false:w	190, 209	_xeCJK_sub_special_punct:nn	1330, 1333, 1336, 1339, 1367
_xeCJK_peek_catcode_ignore_spaces_branches:w	193, 195, 200	_xeCJK_swap_cs_aux:w	96, 98, 99
_xeCJK_peek_catcode_true:w	189, 206	_xeCJK_switch_font:nn	1276, 1281, 1290, 1292, 1299, 2326, 2326, 3317, 3322
_xeCJK_peek_ignore_spaces_branches:w	218, 220, 224	_xeCJK_text_composite:Nnw	2951, 2953
_xeCJK_prop_put_aux:n	2082, 2092	_xeCJK_tl_remove_outer_braces:w	85, 86, 89
_xeCJK_prop_put_aux:nn	2082, 2094	_xeCJK_token_value_charcode:w	233, 236, 241
_xeCJK_punct_break:	938, 1324, 1326	_xeCJK_ulem_Boundary_and_FullLeft_glue:N	3259, 3337, 3337, 3347
_xeCJK_punct_dim_csname:nn	1389, 1389, 1396, 1400	_xeCJK_ulem_CJK_and_CJK:N	3265, 3292, 3292, 3301
_xeCJK_punct_dim_csname:nnn	1389, 1391, 1398, 1402, 1696, 1755	_xeCJK_ulem_CJK_and_FullLeft_glue:N	3255, 3349, 3349, 3361
_xeCJK_punct_dim_csname:nnnn	1389, 1392, 1393, 1404	_xeCJK_ulem_CJK_and_FullRight_glue:N	3257, 3383, 3383, 3391
_xeCJK_punct_glue:NN	776, 776, 794, 804, 813, 820, 835, 845, 866, 908, 979, 3332, 3344, 3357, 3375, 3402, 3427, 3440	_xeCJK_ulem_Default_and_FullLeft_glue:N	3251, 3326, 3326, 3335
_xeCJK_punct_hskip:n	778, 784, 3260	_xeCJK_ulem_Default_and_FullRight_glue:N	3253, 3363, 3363, 3381
_xeCJK_punct_if_long:NT	1730	_xeCJK_ulem_FullLeft_and_CJK:	3262, 3411, 3411, 3418
_xeCJK_punct_if_long:NTF	904, 1375, 1706, 3369	_xeCJK_ulem_FullLeft_and_Default:	3261, 3393, 3393, 3409
_xeCJK_punct_if_long_p:N	1375	_xeCJK_ulem_FullRight_and_CJK:	3264, 3433, 3433, 3446
_xeCJK_punct_if_middle:NT	877, 889, 898, 3372	_xeCJK_ulem_FullRight_and_Default:	3263, 3420, 3420, 3431
_xeCJK_punct_if_middle:NTF	790, 800, 907, 1375, 1496, 1512, 1533, 3397	_xeCJK_ulem_between_CJK_blocks:nnN	3273, 3275, 3279, 3309, 3309
_xeCJK_punct_if_middle_p:N	1375	_xeCJK_ulem_ccglue:	3297, 3314, 3356, 3370, 3441, 3454, 3460
_xeCJK_punct_if_mixed_width:NTF	1375, 1499, 1545	_xeCJK_ulem_class_group_begin:	3298, 3303, 3303, 3359, 3389, 3416, 3444
_xeCJK_punct_if_mixed_width_p:N	1375	_xeCJK_ulem_glue:n	3229, 3231, 3454, 3454
_xeCJK_punct_if_right:N	1375	_xeCJK_ulem_hskip:n	3451, 3454, 3457, 3462
_xeCJK_punct_if_right:NTF	1375, 1605, 1608, 1685, 1688	_xeCJK_ulem_initial:	3213, 3248, 3248
_xeCJK_punct_if_right_p:N	1375	_xeCJK_ulem_leader_type:	3208, 3246, 3247
_xeCJK_punct_kern:NN	785, 785, 918, 927, 936, 945	_xeCJK_ulem_punct_hskip:n	3260, 3448, 3448, 3452
_xeCJK_punct_min_bound:NN	1618, 1621, 1665, 1665	_xeCJK_ulem_skip_punct_begin:	3210, 3243, 3243, 3331, 3342, 3355, 3368
_xeCJK_punct_rule:NN	769, 769, 792, 802, 812, 818, 831, 841, 854, 878, 890, 899, 977, 3400, 3424, 3437	_xeCJK_ulem_skip_punct_end:	3211, 3243, 3245, 3405, 3428, 3442
_xeCJK_punct_width_or_ratio:nN	1497, 1500, 1501, 1554, 1554	_xeCJK_under_CJKsymbol:N	3474, 3492, 3525, 3525
_xeCJK_remove_duplicate_keys:N	2067, 2079, 2079	_xeCJK_update_cs_case_tl:NNnn	994, 1000, 1005, 1009
_xeCJK_reset_char_class:n	2698, 2699, 2700, 2701, 2702, 2732	_xeCJK_update_family:n	2101, 2998, 3010
_xeCJK_restore_output_CJKsymbol:	3475, 3493, 3498, 3498	_xeCJK_update_inline_env_case_tl:	1181, 1190, 1196, 1200, 1200
_xeCJK_save_CJK_class:n	410, 415, 1295	_xeCJK_use_punct_dim:nn	1389, 1395, 1515, 1524, 1563, 1638, 1639, 1690, 1691, 1729
_xeCJK_save_FullRight_check:	957, 967		
_xeCJK_save_FullRight_symbol:N	958, 968, 984		
_xeCJK_save_family_info:	2074, 2231, 2231		
_xeCJK_save_under_dot_CJKsymbol:N	3473, 3479, 3491, 3506, 3527		
_xeCJK_set_char_class_aux:Nnw	422, 431, 437, 2625		
_xeCJK_set_char_class_eq:nn	463, 463, 2687, 2688, 2689, 2690, 2691		

`__xeCJK_use_punct_dim:nnn`
 772, 780, 781, 786, 1389, 1397, 1476, 1479, 1595, 1596,
 1604, 1607, 1670, 1671, 1675, 1676, 1684, 1687, 1723, 1724
`__xeCJK_verb_CJKecglue:` 2693, 2710
`__xeCJK_verb_CJKglue:` 2692, 2709
`__xeCJK_verb_addon:` 2676, 2682
`__xeCJK_verb_check_for_glue:` 2694, 2711
`__xeCJK_verb_ignore_spaces:w` 2695, 2712
`__xeCJK_warning:n` 39, 2519
`__xeCJK_warning:nx` 36, 40, 1793, 2415, 2563, 2889, 2890, 2891
`__xeCJK_warning:nxx` 41, 1871, 2115, 2578
`__xeCJK_zero_glue:` 806, 808, 861, 909, 2654, 2655, 2660, 2661

`\` 1253, 1304, 2851, 2896, 2897, 3216, 3223, 3286

A

`\addCJKfontfeature` 2503, 2524
`\addCJKfontfeatures` 6, 2494, 2498, 2503
`\addtohook` 3200
`\AfterEndPreamble` 65
`\AfterPreamble` 64
`KaiMingPunct` 4
`KaiMingPunct+` 4
`KaiMingPunct-` 4
`FallBack` 7
`\AllowBreakBetweenPuncts` 1320
`Mapping` 5
`\AssignTemplateKeys` 1469
`\AtBeginDocument` 68
`\AtEndOfPackage` 529, 2770, 3154
`\AtEndPreamble` 63
`\AutoFakeBold` 1914, 2015
`\AutoFakeSlant` 1914, 2015
`\AutoFallBack` 1835

B

`\begin` 2875
`\bfdefault` 2617
`\BoldFeatures` 2001
`\BoldFont` 2001
`\BoldItalicFeatures` 2001
`\BoldItalicFont` 2001
`\BoldSlantedFeatures` 2001
`\BoldSlantedFont` 2001
`\bool_gset_false:N` 2805, 3572, 3700
`\bool_gset_true:N` 1920, 1928, 2808, 2813, 3691
`\bool_if:cT` 2247, 2248
`\bool_if:NF` 1220, 1243, 2673, 2684, 3203
`\bool_if:NT` 1229, 1575,
 2176, 2582, 2886, 3174, 3186, 3470, 3476, 3478, 3482, 3698
`\bool_if:nT` 618, 729, 2180, 2190, 3133
`\bool_if:NTF` 640, 721,
 725, 1077, 1086, 1094, 1106, 1111, 1484, 1519, 1529,
 1591, 1617, 1620, 2178, 2705, 3207, 3639, 3655, 3668, 3679
`\bool_if:nTF` 88, 443,
 558, 566, 607, 611, 627, 635, 1061, 1100, 1584, 1736, 3104
`\bool_if:p:n` 2169
`\bool_new:N` 213, 704, 1217,
 1881, 1910, 1911, 2680, 2738, 2819, 3234, 3241, 3665, 3706
`\bool_set_eq:NN` 2052, 2053
`\bool_set_false:c` 1949, 1950, 1958, 1961, 2049, 2050

`\bool_set_false:N`
 191, 691, 1231, 2019, 2028, 3240, 3658, 3671, 3682
`\bool_set_true:c` 1957, 1962
`\bool_set_true:N` 198,
 686, 696, 1222, 2022, 2031, 2686, 3205, 3238, 3568, 3648
`\Boundary` 299
`\box_new:N` 30, 3485
`\box_use:N` 3520, 3528
`\box_wd:N` 3519

C

`\c__xeCJK_CJ_chars_clist` 342, 342, 359
`\c__xeCJK_CJK_chars_clist` 362, 362, 483, 2620
`\c__xeCJK_CL_chars_clist` 326, 326, 355
`\c__xeCJK_config_ext_tl` 3188, 3190
`\c__xeCJK_encoding_tl` 2364, 2615, 2617, 2885, 2885
`\c__xeCJK_EX_chars_clist` 339, 339, 357
`\c__xeCJK_FullLeft_chars_clist` 321, 321, 478, 2620
`\c__xeCJK_FullRight_chars_clist` 353, 353, 479, 2622
`\c__xeCJK_HalfLeft_chars_clist` 308, 308, 476
`\c__xeCJK_HalfRight_chars_clist` 308, 310, 477
`\c__xeCJK_IS_chars_clist` 341, 341, 358
`\c__xeCJK_IVS_chars_clist` 398, 398, 485
`\c__xeCJK_left_tl` 794,
 804, 824, 831, 835, 838, 841, 845, 848, 854, 866, 878,
 890, 899, 916, 934, 1318, 1318, 1480, 1596, 1609, 1670,
 1675, 1686, 1701, 1717, 1723, 3332, 3344, 3357, 3399, 3402
`\c__xeCJK_math_family_tl` 2613, 2615, 2617
`\c__xeCJK_math_tl` .. 2600, 2601, 2604, 2605, 2614, 2616, 2618
`\c__xeCJK_NormalSpace_chars_clist` 308, 312, 484
`\c__xeCJK_NS_chars_clist` 334, 334, 356
`\c__xeCJK_OP_chars_clist` 313, 313, 323
`\c__xeCJK_package_ext_tl` 48, 51
`\c__xeCJK_PO_chars_clist` 320, 320, 324
`\c__xeCJK_PR_chars_clist` 351, 351, 360
`\c__xeCJK_punct_style_plain_tl` 1698, 1757, 1775, 1783
`\c__xeCJK_right_tl` 792,
 802, 812, 813, 818, 820, 870, 883, 895, 908, 925, 943,
 976, 977, 979, 1318, 1319, 1480, 1595, 1606, 1671, 1676,
 1689, 1702, 1719, 1724, 3375, 3400, 3424, 3427, 3437, 3440
`\c_catcode_letter_token` 1055, 1075
`\c_eleven` 561, 621, 863, 1020
`\c_empty_tl` 1976, 1993, 1994
`\c_four` 119, 140, 3055
`\c_fourteen` 2657, 2664
`\c_group_begin_token` 527, 3538, 3539, 3552
`\c_group_end_token` 528, 3544, 3545, 3551
`\c_math_toggle_token` 719, 1131
`\c_max_dim` 1409, 1415, 1426, 1429,
 1431, 1433, 1510, 1535, 1556, 1560, 1569, 1624, 1651, 1659
`\c_minus_one` 862
`\c_one` . 266, 300, 861, 1718, 1886, 2487, 2644, 3055, 3343, 3633
`\c_one_fp` 1410, 1416, 1434
`\c_one_thousand` 113, 167
`\c_seven` 134, 149, 2626
`\c_space_tl` 692
`\c_space_token` 197, 222
`\c_ten` 614, 630
`\c_ten_thousand` 45
`\c_thirteen` 864
`\c_three` 118, 139, 302, 1405, 1406, 1437, 1720, 2646, 3053, 3464
`\c_two` 117, 127, 133, 301, 781, 1516,
 1540, 2645, 2754, 2779, 2787, 2794, 3600, 3604, 3606, 3610

\c_two_hundred_fifty_five	303	\clist_const:Nn	308,
\c_two_hundred_fifty_six	3180	310, 312, 313, 320, 326, 334, 339, 341, 342, 351, 362, 398	
\xeCJK_math_fam_int	2618, 2626	\clist_const:Nx	321, 353
\xeCJK_space_skip_tl	111, 111, 562, 622	\clist_count:N	1886
\c_zero	44, 267, 299, 2643, 2671, 3017, 3602	\clist_gconcat:ccN	425
\c_zero_dim	773, 774, 1423, 1435,	\clist_gconcat:NNN	439
1700, 1701, 1702, 1707, 1741, 1758, 2729, 2752, 2858, 3514		\clist_gset:Nn	1340, 2496
\c_zero_fp	1430, 1432, 1559, 1627, 1654	\clist_if_empty:cF	2219
\c_zero_skip	115, 124, 129, 146, 809	\clist_if_in:NnT	2529
\char_set_catcode:n	3149	\clist_map_break:	2541
\char_set_catcode_active:N	3742	\clist_map_function:nN	415
\char_set_catcode_ignore:n	268	\clist_map_inline:cn	466, 2735
\char_set_catcode_letter:n	3147	\clist_map_inline:Nn	427, 1343, 1380, 1897, 2623
\char_set_lccode:n	3743	\clist_map_inline:nn	542, 554, 581, 748, 764, 766, 1185, 1194, 1238, 1296
\char_value_catcode:n	3149	\clist_new:c	277, 288
\CheckFullRight	950	\clist_new:N	34, 35, 1339, 2494
\CheckSingle	1034	\clist_put_left:cn	1901
\CJK	299	\clist_put_left:cx	2129, 2134
\CJK@eUL	3236, 3236, 3239	\clist_put_right:Nn	2087, 2088, 2226
\CJK@hundredmillion	3184	\clist_put_right:Nx	2221
\CJK@ifundefined	3140, 3141	\clist_remove_all:Nn	2235, 2267, 2531
\CJK@postUnderdot	3481	\clist_set:Nn	424, 1992, 2063, 2508
\CJK@preUnderdot	3471	\clist_set:Nx	1885, 2539
\CJK@tenthousand	3183	\clist_set_eq:cN	1987
\CJK@UL	3236, 3236, 3237	\cs:w	104, 3135
\CJK@underdotBox	3489, 3520	\cs_end:	104, 247, 1385, 3135
\CJK@underdotSkip	3472	\cs_generate_variant:Nn	252,
\CJK@UnicodeEnc	3176, 3182	439, 440, 491, 501, 507, 982, 1161, 1249, 1266, 1315,	
\CJKaddEncHook	3177	1747, 1765, 1882, 1909, 1997, 1998, 1999, 2078, 2091,	
\CJKecglue	571, 574, 579, 595, 616, 633, 676, 680, 687, 697, 699,	2118, 2125, 2126, 2172, 2303, 2369, 2419, 2493, 2521, 2544	
722, 2655, 2661, 2693, 2710, 2725, 2727, 2728, 3226, 3230		\cs_gset_eq:cc	2298
xCJKecglue	3	\cs_gset_eq:NN	103, 2773, 2987, 2988, 3473, 3491
\CJKfamily	5,	\cs_gset_nopar:Npx	73, 3097
2377, 2377, 2400, 2473, 2491, 2581, 3002, 3003, 3004, 3005		\cs_gset_protected_nopar:cpx	2098, 2271, 2362
\CJKfamilydefault	6, 2348, 2349,	\cs_if_eq:NNF	955, 1039, 1840
2536, 2548, 2550, 2558, 2563, 2565, 2567, 2568, 2572,		\cs_if_eq:NNT	965, 1047, 1848
2576, 2579, 2581, 2607, 2608, 2899, 2908, 2909, 2913, 3005		\cs_if_eq:NNTF	3153, 3750
\CJKfilltwosides	3532	\cs_if_exist:cT	2942
\CJKfixedspacing	2739	\cs_if_exist:cTF	2347, 2372
\CJKfontspec	6, 2468, 2476	\cs_if_exist:NF	737, 1308, 3140
\CJKglue	609, 640, 667, 671, 741, 814, 845,	\cs_if_exist:p:N	3134
905, 2654, 2660, 2692, 2709, 2723, 3219, 3228, 3535, 3702		\cs_if_exist_use:c	3182
\CJKmath	2598	\cs_if_exist_use:cF	2309, 2318, 2339, 2766, 2957
\CJKnospace	2927, 2928	\cs_if_exist_use:cTF	1486, 2374
\CJKnumber	2820	\cs_if_free:NNTF	3141
\CJKpunctsymbol	832, 842, 855, 920, 939, 973, 2854, 2855	\cs_new:Npn	84, 86
\CJKrmdefault	6, 2430,	\cs_new_eq:cN	293
2441, 2449, 2554, 2568, 2571, 2572, 2899, 2899, 2910, 3002		\cs_new_eq:NN	92,
\CJKsetecglue	2925, 2925, 2926	93, 238, 243, 527, 528, 784, 912, 2325, 2452, 2503, 2739,	
\CJKsfdefault	6, 2431, 2442, 2455, 2555, 2899, 2900, 3003	2903, 2926, 2959, 2979, 2980, 3010, 3012, 3087, 3088, 3101	
\CJKspace	705, 2927, 2927	\cs_new_nopar:Npn	80, 158, 230, 232, 236, 241, 250, 292, 416, 492, 973,
\CJKsymbol	550, 603, 741, 743,	1341, 1342, 1389, 1391, 1393, 1395, 1397, 1554, 1567,	
745, 1280, 1281, 1840, 1842, 1843, 1848, 1849, 2854,		1600, 1612, 1649, 1665, 1679, 1748, 2208, 2306, 2356,	
2854, 3299, 3318, 3323, 3473, 3474, 3479, 3491, 3492, 3505		2357, 2358, 2426, 2437, 2854, 2855, 2947, 2953, 3284, 3614	
\CJKttdefault	6, 2432, 2443, 2460, 2556, 2899, 2901, 3004	\cs_new_protected:Npn	55, 57, 59, 94, 182, 184, 186, 214,
\CJKunderanysymbol	3486, 3486	985, 1009, 1200, 3498, 3511, 3707, 3722, 3735, 3747, 3749	
\CJKunderdot	3468, 3468	\cs_new_protected_nopar:Npn	36, 37, 38,
\clist_clear:c	2043	39, 40, 41, 42, 43, 44, 45, 69, 71, 82, 101, 168, 195, 220,	
\clist_clear:N	1973, 1980, 2045, 2083	271, 282, 410, 422, 441, 453, 463, 489, 494, 496, 502,	
\clist_concat:NNN	2065, 2514, 2619, 2621	508, 514, 523, 525, 605, 652, 658, 717, 741, 762, 769,	

776, 785, 788, 798, 808, 810, 816, 822, 834, 836, 844, 846, 857, 868, 881, 893, 902, 913, 922, 932, 941, 974, 983, 1053, 1073, 1119, 1129, 1138, 1152, 1236, 1256, 1267, 1345, 1356, 1367, 1399, 1401, 1403, 1474, 1578, 1694, 1715, 1753, 1854, 1860, 1890, 1940, 1970, 1990, 2036, 2058, 2079, 2092, 2094, 2096, 2109, 2121, 2123, 2127, 2132, 2137, 2139, 2173, 2199, 2215, 2231, 2242, 2255, 2284, 2307, 2316, 2326, 2337, 2359, 2399, 2402, 2408, 2481, 2504, 2527, 2602, 2630, 2682, 2732, 2740, 2764, 2775, 2782, 3015, 3062, 3201, 3243, 3245, 3248, 3292, 3303, 3309, 3326, 3337, 3349, 3363, 3383, 3393, 3411, 3420, 3433, 3448, 3454, 3460, 3462, 3525, 3548, 3569, 3582, 3619, 3625, 3631, 3636, 3652, 3688, 3695, 3729	\end 2875 \EnvCS 1169 Verb 5 \etex_currentgrouplevel:D 2671 \etex_currentgroupstype:D 2657, 2664 \etex_dimexpr:D 1539, 1563, 1573, 1637, 1663 \etex_fontcharwd:D 1727, 2755 \etex_glueshrink:D 154 \etex_gluestretch:D 153 \etex_iffontchar:D 108 \etex_lastnodetype:D ... 561, 614, 621, 630, 859, 1020, 3343 \etex_numexpr:D 3180 NewLineCS 3 NewLineCS+ 3 NewLineCS- 3 \exp_after:wN 103, 104, 170, 171, 185, 199, 200, 205, 206, 208, 209, 223, 224, 227, 233, 1065, 1066, 1068, 1976, 2715, 2717, 2718, 2719, 2778, 2779, 3039, 3040, 3041, 3042, 3135, 3287, 3289, 3480, 3501, 3502, 3504, 3505, 3506, 3574, 3576, 3577, 3578, 3731, 3732 \exp_args:Nc 276, 1382, 2491 \exp_args:NNv 2944 \exp_args:No 1064 \exp_args:NV 83 \exp_not:c 751, 2704 \exp_not:N 204, 737, 2105, 2275, 2276, 2279, 2364, 2365, 2366, 2554, 2555, 2556, 2752, 2906, 2910 \exp_not:n 499, 505, 2101, 2102, 2547, 2707, 2750, 3501 \exp_not:o 76, 3098 \exp_not:V 512, 520, 2103, 2104, 2540, 2909 \exp_not:v 2209, 2222 \exp_stop_f: 108, 175, 178, 256, 1751
\cs_new_protected_nopar:Npx 2473 \cs_set:cpn 3178 \cs_set_eq:NN 96, 97, 98, 188, 687, 692, 699, 957, 958, 959, 960, 967, 968, 1041, 1042, 1048, 1324, 1326, 1842, 1843, 1849, 2101, 2400, 2547, 2654, 2655, 2660, 2661, 2692, 2693, 2694, 2695, 2709, 2710, 2711, 2712, 2727, 2728, 2960, 3141, 3208, 3216, 3217, 3218, 3223, 3224, 3225, 3236, 3246, 3474, 3479, 3492, 3504, 3535, 3536, 3662, 3702, 3721, 3728, 3734	\cs_set_nopar:Npn 3102 \cs_set_protected:Npn 3177 \cs_set_protected_nopar:Npn 671, 680, 697, 2723, 2725, 3179, 3228, 3230, 3666, 3677 \cs_set_protected_nopar:Npx 2696 \cs_to_str:N 2472 \cs_undefine:c 1348, 1371, 2114 \cs_undefine:N 99, 2561, 2991, 2992 \curr@fontshape 2742, 2750, 2761, 2766, 2777 \CurrentOption 3195, 3557

\FullRight [299](#)

G

\g__xeCJK_after_end_preamble_hook_tl [54](#), [60](#), [65](#), [72](#)
\g__xeCJK_after_preamble_hook_tl [53](#), [58](#), [64](#), [68](#)
\g__xeCJK_at_end_preamble_hook_tl [52](#), [56](#), [63](#), [70](#)
\g__xeCJK_auto_fake_bold_bool [1910](#), [1917](#), [1920](#), [2052](#)
\g__xeCJK_auto_fake_slant_bool [1911](#), [1925](#), [1928](#), [2053](#)
\g__xeCJK_base_class_seq [404](#), [404](#), [405](#), [1269](#)
\g__xeCJK_CJK_class_seq [404](#), [409](#), [412](#), [524](#), [3065](#)
\g__xeCJK_CJK_sub_class_seq
..... [1209](#), [1209](#), [1285](#), [1294](#), [2234](#), [2510](#), [3266](#), [3268](#), [3592](#), [3607](#)
\g__xeCJK_class_seq [269](#), [269](#), [278](#), [289](#), [405](#), [531](#)
\g__xeCJK_config_bool [2805](#), [2808](#), [2813](#), [2819](#), [3186](#)
\g__xeCJK_config_name_tl [2809](#), [2814](#), [2818](#), [3190](#)
\g__xeCJK_default_features_clist [2066](#), [2494](#), [2496](#)
\g__xeCJK_embolden_factor_fp .. [1912](#), [1921](#), [1932](#), [2025](#), [2054](#)
\g__xeCJK_family_font_name_prop
..... [1878](#), [1894](#), [2111](#), [2228](#), [2229](#),
[2236](#), [2258](#), [2261](#), [2291](#), [2410](#), [2506](#), [2532](#), [2535](#), [2562](#), [2575](#)
\g__xeCJK_family_font_options_prop
..... [2228](#), [2230](#), [2238](#), [2264](#), [2268](#), [2292](#), [2511](#)
\g__xeCJK_family_int [2038](#), [2057](#), [2487](#)
\g__xeCJK_family_name_prop
..... [2113](#), [2228](#), [2228](#), [2275](#), [2290](#), [2361](#), [2611](#)
\g__xeCJK_features_id_prop [2001](#),
[2001](#), [2002](#), [2003](#), [2004](#), [2005](#), [2006](#), [2007](#), [2040](#), [2138](#), [2217](#)
\g__xeCJK_indent_bool [2823](#), [2886](#)
\g__xeCJK_last_punct_tl
..... [787](#), [787](#), [790](#), [792](#), [794](#), [800](#), [802](#), [804](#), [812](#),
[813](#), [818](#), [820](#), [830](#), [840](#), [853](#), [876](#), [888](#), [897](#), [917](#), [918](#),
[919](#), [926](#), [927](#), [928](#), [935](#), [936](#), [937](#), [944](#), [945](#), [946](#), [976](#),
[977](#), [979](#), [3397](#), [3399](#), [3400](#), [3402](#), [3424](#), [3427](#), [3437](#), [3440](#)
\g__xeCJK_listings_IVS_bool ... [3572](#), [3691](#), [3698](#), [3700](#), [3706](#)
\g__xeCJK_math_bool [2582](#), [2598](#)
\g__xeCJK_new_class_seq [269](#), [270](#), [279](#), [3053](#), [3057](#)
\g__xeCJK_non_CJK_class_seq [404](#), [406](#), [407](#), [746](#)
\g__xeCJK_number_bool [2822](#), [3174](#)
\g__xeCJK_punct_style_seq [1732](#), [1788](#), [1794](#), [1797](#)
\g__xeCJK_punct_width_tl [1337](#), [1489](#), [1491](#)
\g__xeCJK_slant_factor_fp [1913](#), [1929](#), [1933](#), [2034](#), [2055](#)
\g__xeCJK_spacefactor_int [113](#), [126](#), [143](#), [161](#), [164](#), [166](#), [167](#), [585](#)
\g__xeCJK_special_punct_clist . [1339](#), [1339](#), [1340](#), [1343](#), [1380](#)
\g__xeCJK_sub_key_seq [1939](#), [1939](#), [1942](#), [2047](#)
\g__xeCJK_under_symbol_box [3485](#), [3514](#), [3528](#)
\g__xeCJK_unknown_family_seq [2412](#), [2414](#), [2420](#)
\g__xeCJK_verb_exspace_skip [2724](#), [2726](#), [2760](#), [2763](#)
\g_fontspeg_encoding_tl [2784](#), [2885](#)
\group_align_safe_begin: [192](#), [217](#), [1082](#), [1140](#), [1154](#)
\group_align_safe_end:
.. [189](#), [190](#), [216](#), [1085](#), [1093](#), [1099](#), [1143](#), [1157](#), [1164](#), [1171](#)
\group_begin: [70](#), [1892](#), [2060](#), [2100](#), [2233](#), [2744](#), [3087](#), [3488](#), [3741](#)
\group_end: . [70](#), [1907](#), [2076](#), [2106](#), [2240](#), [2748](#), [3088](#), [3495](#), [3746](#)

H

\HalfLeft [304](#)
\HalfRight [304](#)
\hbox_overlap_left:n [3528](#)
\hbox_set:Nn [170](#), [3489](#), [3513](#)
\hbox_to_zero:n [3517](#)
CheckFullRight [5](#)
CheckSingle [3](#)
\hskip [3218](#), [3225](#)

I

\icprotect [3134](#), [3135](#)
MiddlePunct [4](#)
MiddlePunct+ [4](#)
MiddlePunct- [4](#)
\if_case:w [175](#), [178](#), [2653](#)
\if_catcode:w [203](#)
\if_cs_exist:w [247](#), [1385](#)
\if_dim:w [649](#)
\if_false: [3732](#)
\if_int_compare:w [183](#), [255](#), [1377](#)
\if_meaning:w [197](#), [222](#), [3286](#)
\if_predicate:w [2168](#)
\IfBooleanF [2381](#)
\IfBooleanT [420](#), [1214](#), [1224](#), [1233](#), [2391](#)
\IfBooleanTF [2385](#)
\ifCTEX@fntef [3153](#)
\IfInstanceExistTF [1778](#), [1792](#), [1806](#)
\IfNoValueF [3144](#)
\IfNoValueTF [449](#), [450](#), [2472](#)
\indentfirst [2820](#)
\InlineEnv [1176](#)
\int_add:Nn [3633](#)
\int_case:nnn [859](#)
\int_compare:nNnF [2671](#), [3052](#), [3343](#)
\int_compare:nNnT [2664](#)
\int_compare:nNnTF [113](#), [126](#), [143](#), [1886](#), [2657](#), [3017](#), [3464](#), [3715](#)
\int_compare_p:nNn [561](#), [614](#), [621](#), [630](#)
\int_const:cn [287](#)
\int_const:Nn [2618](#)
\int_div_truncate:nn [163](#)
\int_eval:n [251](#), [2487](#)
\int_gincr:N [2038](#)
\int_gset_eq:NN [167](#), [585](#)
\int_if_even:nF [1031](#)
\int_if_even:nTF [1022](#)
\int_if_exist:cF [1258](#)
\int_if_exist:cTF [273](#), [284](#), [1240](#)
\int_incr:N [460](#), [1024](#), [2637](#)
\int_max:nn [450](#)
\int_min:nn [449](#)
\int_new:N [27](#), [28](#), [29](#), [166](#), [2057](#), [2648](#)
\int_set:cn [3064](#)
\int_set:Nn [445](#), [449](#), [450](#), [456](#), [465](#), [2734](#)
\int_set_eq:NN [446](#), [2643](#), [2644](#), [2645](#), [2646](#)
\int_step_inline:nnnn [3055](#)
\int_to_hexadecimal:n [1879](#)
\int_while_do:nNnn [1020](#)
\int_zero:N [1019](#)
\ItalicFeatures [2001](#)
\ItalicFont [2001](#)
\IVS [304](#)

J

CJKecglue [3](#)
CJKglue [3](#)
CJKmath [3](#)
CJKnumber [3](#)
CJKspace [3](#)

K

\KaiMingPunct [1320](#)

\keys_define:nn	259, 667, 676, 705, 950, 989, 1034, 1176, 1207, 1320, 1766, 1835, 1914, 1943, 2009, 2015, 2598, 2640, 2801, 2820, 2826, 2887	\l__xeCJK_mixed_margin_ratio_fp	1449
\keys_set:nn	2856, 2916	\l__xeCJK_mixed_margin_width_dim	1448
\keys_set_known:nVN	2068	\l__xeCJK_mixed_punct_ratio_fp	1443
\keyval_parse:NnN	2091	\l__xeCJK_mixed_punct_width_dim	1442
\keyval_parse:NNV	2082	\l__xeCJK_new_line_cs_case_tl	1163, 1167, 1174
\KeyValue	1411, 1412, 1413, 1414, 1417, 1418, 1419, 1420	\l__xeCJK_new_line_cs_seq	1163
		\l__xeCJK_no_break_cs_case_tl	978, 1016
		\l__xeCJK_no_break_cs_seq	1016
		\l__xeCJK_optimize_kerning_bool	1459, 1620
		\l__xeCJK_optimize_margin_bool	1453, 1519, 1529
		\l__xeCJK_peek_ignore_spaces_bool	191, 198, 213, 721, 725, 1077, 1086, 1094, 1106, 1111
		\l__xeCJK_plain_equation_bool	1102, 1208
		\l__xeCJK_reserve_space_bool	640, 707, 732
		\l__xeCJK_restore_listings_toks_tl	3565, 3573, 3577, 3584, 3594, 3613
		\l__xeCJK_same_align_margin_dim	1460
		\l__xeCJK_same_align_ratio_fp	1461
		\l__xeCJK_slant_factor_fp	2032, 2055, 2135
		\l__xeCJK_sub_cancel_bool	1217, 1220, 1222, 1229, 1231, 1243
		\l__xeCJK_sub_family_tl	2246, 2250, 2257, 2262, 2269, 2271, 2279
		\l__xeCJK_sub_font_name_tl	1977, 1979, 1983, 1984, 1985, 1986, 1988, 1994, 1995
		\l__xeCJK_sub_font_options_clist	1973, 1980, 1987, 1992
		\l__xeCJK_sub_key_seq	1954, 2046, 2244
		\l__xeCJK_tmp_box	30, 170, 3513, 3519
		\l__xeCJK_tmp_dim	31, 1482, 1510, 1515, 1523, 1721, 1729, 1742
		\l__xeCJK_tmp_prop	33, 2081, 2084, 2093, 2095
		\l__xeCJK_tmp_skip	32, 1025, 1027, 1032
		\l__xeCJK_tmpa_clist	34, 424, 426, 427, 1885, 1886, 1887, 2265, 2267, 2269, 2508, 2515, 2517, 2529, 2531, 2539, 2540, 2620, 2622, 2623
		\l__xeCJK_tmpa_int	27, 293, 455, 457, 459, 460, 465, 467, 1019, 1022, 1024, 1031, 2632, 2633, 2636, 2637, 2734, 2736
		\l__xeCJK_tmpa_tl	25, 510, 511, 512, 516, 517, 519, 520, 1476, 1520, 1530, 1540, 1573, 1576, 1972, 1974, 1976, 1979, 1983, 2111, 2113, 2115, 2259, 2262, 2276, 2279, 2295, 2296, 2472, 2473, 2474, 2483, 2484, 2488, 2491, 2506, 2517, 2537, 2567, 2579, 2605, 2608, 2611, 3724, 3726
		\l__xeCJK_tmppb_clist	35, 2512, 2515
		\l__xeCJK_tmppb_int	28, 455, 457, 2632, 2633
		\l__xeCJK_tmppb_tl	26, 1477, 1520, 1523, 1540, 1576, 2486, 2488, 2489, 2533, 2536, 2537, 2540, 2611, 2613
		\l__xeCJK_tmppc_int	29, 456, 459
		\l__xeCJK_ulem_hook_used_bool	3203, 3205, 3234, 3470, 3476, 3478, 3482
		\l__xeCJK_ulem_skip_punct_bool	3207, 3238, 3240, 3241
		\l__xeCJK_underdot_output_tl	3480, 3500, 3507, 3510
		\l__xeCJK_verb_addon_bool	2684, 2686, 2738
		\l__xeCJK_verb_case_int	2643, 2644, 2645, 2646, 2648, 2653
		\l__xeCJK_xecglue_bool	560, 620, 686, 691, 696, 704, 2705, 2706
		\l_keys_choice_tl	1917, 1925
		\l_keys_key_tl	2845, 2846, 2889, 2890, 2891
		\l_keys_value_tl	1778, 1779, 1780, 1921, 1929, 2023, 2032, 2814
		\l_peek_token	197, 204, 222, 589, 731, 733, 737, 978, 1058, 1063, 1064, 1065, 1066, 1083, 1091, 1103, 1123, 1141
		\l_xeCJK_current_font_tl	1390, 1394, 2304, 2305, 2309, 2311
		\l_xeCJK_family_tl	1858, 1870, 2305, 2312, 2339, 2342, 2343, 2345, 2349, 2352, 2382, 2389, 2390, 2398, 2398, 2506, 2512, 2533, 2742, 2750, 2761

L

\l__keys_module_tl	2851
\l__peek_search_token	188, 204
\l__xeCJK_add_font_prop	2039, 2122, 2124, 2225
\l__xeCJK_add_min_bound_to_margin_bool	1452, 1575
\l__xeCJK_auto_fake_bold_bool	2019, 2022, 2052, 2176, 2191
\l__xeCJK_auto_fake_slant_bool	2028, 2031, 2053, 2178
\l__xeCJK_ccglue_skip	672, 675, 3220, 3229, 3321, 3461
\l__xeCJK_check_single_cs_case_tl	1142, 1151, 1166, 1173
\l__xeCJK_current_coord_tl	2311, 2313, 2320, 2322, 2341, 2353, 2777, 2780
\l__xeCJK_different_align_margin_dim	1462
\l__xeCJK_different_align_ratio_fp	1463
\l__xeCJK_ecglue_skip	623, 681, 698, 703, 3227, 3231
\l__xeCJK_embolden_factor_fp	2023, 2054, 2130
\l__xeCJK_enabled_global_setting_bool	1439, 1484, 1586
\l__xeCJK_enabled_kerning_bool	1455, 1591
\l__xeCJK_env_cs_case_tl	1167, 1170, 1174
\l__xeCJK_env_cs_seq	1170
\l__xeCJK_fallback_family_tl	1893, 1895, 1899, 1900, 1902, 1903, 1904, 1905
\l__xeCJK_fallback_first_bool	1881
\l__xeCJK_family_default_init_tl	2548, 2902, 2904, 2913
\l__xeCJK_family_name_tl	1951, 2062, 2072, 2073, 2237, 2239, 2246, 2259, 2265, 2273, 2276
\l__xeCJK_fill_width_dim	3534, 3539, 3547, 3552
\l__xeCJK_fixed_margin_ratio_fp	1447
\l__xeCJK_fixed_margin_width_dim	1446
\l__xeCJK_fixed_punct_ratio_fp	1441
\l__xeCJK_fixed_punct_width_dim	1440
\l__xeCJK_font_features_clist	2065, 2067, 2069
\l__xeCJK_font_name_bf_tl	2153, 2155
\l__xeCJK_font_name_tl	1895, 1896, 1985, 1986, 2064, 2104, 2237
\l__xeCJK_font_options_clist	2063, 2066, 2235, 2239
\l__xeCJK_fontspec_family_tl	2102, 2105
\l__xeCJK_fontspec_options_clist	2045, 2069, 2103, 2221, 2226
\l__xeCJK_inline_env_case_tl	1156, 1202, 1204, 1206
\l__xeCJK_inline_env_seq	1180, 1187, 1188, 1195, 1199, 1203
\l__xeCJK_kerning_margin_minimum_dim	1466, 1615
\l__xeCJK_kerning_margin_ratio_fp	1465, 1661
\l__xeCJK_kerning_margin_width_dim	1464, 1659, 1660
\l__xeCJK_kerning_total_ratio_fp	1458, 1627, 1636
\l__xeCJK_kerning_total_width_dim	1457, 1624, 1625
\l__xeCJK_listings_env_bool	2673, 2680, 3568
\l__xeCJK_listings_letter_bool	3639, 3648, 3655, 3658, 3665, 3668, 3671, 3679, 3682
\l__xeCJK_margin_minimum_dim	1454, 1508
\l__xeCJK_middle_margin_ratio_fp	1451, 1538
\l__xeCJK_middle_margin_width_dim	1450, 1535, 1536
\l__xeCJK_middle_punct_ratio_fp	1445
\l__xeCJK_middle_punct_width_dim	1444
\l__xeCJK_min_bound_to_kerning_bool	1456, 1617

\l_xeCJK_punct_style_tl	1390, 1392, 1698, 1710, 1757, 1761, 1775, 1779	InlineEnv+	4
\LA@space	3286	InlineEnv-	4
PlainEquation	3	\nobreak	2876
SlantFactor	4	\NoBreakCS	1015
AllowBreakBetweenPuncts	5	\nobreakspace	2942, 2942, 2944, 2945
\LocalConfig	2801	\normalfont	2450, 2456, 2461
\LongPunct	1320	\NormalSpace	304
\lst@AddToHook	3561, 3562, 3563, 3568	\normalspacedchars	9, 469, 469
\lst@Append	3634, 3675, 3686	EnvCS	3
\lst@AppendLetter	3666	EnvCS+	3
\lst@AppendOther	3677	EnvCS-	3
\lst@arg	3726, 3737	O	
\lst@FillOutputBox	3701	NoBreakCS	5
\lst@ifec	3715	NoBreakCS+	5
\lst@ifflexible	3690, 3697	NoBreakCS-	5
\lst@ifletter	3642, 3661, 3674, 3685	LocalConfig	2
\lst@ifNextCharActive	3721	LongPunct	4
\lst@InlineGJ	3734	LongPunct+	4
\lst@InlineGJEnd	3739	LongPunct-	4
\lst@InsideConvert@	3728	\or:	2656, 2663
\lst@lastother	3662, 3686	\orig@text@composite	2950, 2956, 2959
\lst@length	3633	P	
\lst@letterfalse	3659, 3661, 3683, 3685	\par	2874
\lst@lettertrue	3640, 3646, 3672, 3674	\PassOptionsToPackage	2833, 2840, 2846, 3194, 3195, 3557
\lst@numberstyle	3573	\pdfstringdefDisableCommands	3122
\lst@Output	3643, 3657, 3661, 3670, 3681, 3685	\pdftex_strcmp:D	175, 178
\lst@OutputOther	3645, 3674	\peek_after:Nw	193, 199, 218, 223
\lst@ProcessLetter	3622	\peek_catcode:Ntf	1055, 1131
\lst@ProcessOther	3623	\peek_meaning_remove:Ntf	586
\lst@whitespacefalse	3638, 3654	\penalty	3217, 3224
M		\Pifont	3114
\makexxeCJKactive	262, 266, 266	\PlainEquation	1207
\makexxeCJKinactive	263, 266, 267, 3098, 3100, 3109, 3115, 3125	\prg_do_nothing:	495
EmboldenFactor	4	\prg_new_conditional:Npnn	46, 106, 173, 245, 253, 647, 1375, 1383, 2161, 2166, 2210
\mddefault	2615	\prg_new_protected_conditional:Npnn	2370
\MiddlePunct	1320	\prg_return_false:	49, 109, 179, 248, 257, 650, 1378, 1386, 2164, 2170, 2213, 2375
\mode_if_vertical:T	3537	\prg_return_true:	49, 109, 176, 179, 248, 257, 650, 1378, 1386, 2164, 2170, 2213, 2375
\msg_critical:nn	10	\ProcessKeysOptions	2883
\msg_critical:nnn	20	\ProcessOptions	3196, 3558
\msg_error:nn	37	\prop_clear:N	2039, 2081
\msg_error:nnn	23	\prop_get:Nn	1882
\msg_error:nnx	38	\prop_get:NnNF	2544
\msg_info:nnx	42	\prop_get:NnNT	2111, 2275, 2295
\msg_info:nnxx	43	\prop_get:No	1878
\msg_new:nnn	3, 11, 36	\prop_get:NoNF	2532
\msg_redirect_module:nnn	2830, 2831, 2837, 2838	\prop_get:NVNF	1894
\msg_warning:nn	39	\prop_get:NVNT	2258, 2264, 2511, 2611
\msg_warning:nnx	40	\prop_get:NVNTF	2506
\msg_warning:nnxx	41	\prop_get:NxNF	2535
N		\prop_gpop:NnNT	2113
indentfirst	3	\prop_gput:Nnn	2369
\newCJKfontfamily	6, 2468, 2470	\prop_gput:NnV	2296
\NewDocumentCommand	266, 267, 417, 437, 469, 474, 481, 1017, 1210, 1218, 1227, 1313, 1316, 1790, 1804, 1883, 2377, 2447, 2453, 2458, 2468, 2470, 2476, 2495, 2498, 2599, 2669, 2681, 2914, 2919, 2921, 2923, 2924, 2925, 2927, 2928, 2929, 2931, 2933, 2935, 2937, 3142	\prop_gput:NVV	2236, 2238, 2261, 2268
\NewLineCS	1162	\prop_gput:Nxx	2361
\newXeTeXintercharclass	276	\prop_if_empty:Nf	2410
InlineEnv	4	\prop_if_empty:Ntf	2562
		\prop_map_break:n	2576
		\prop_map_function:NN	2138

<code>\prop_map_inline:Nn</code>	2007, 2040, 2084, 2217, 2225, 2575	<code>\skip_horizontal:n</code>	784, 2726, 2858, 3458, 3465
<code>\prop_new:N</code>	33, 2001, 2228, 2229, 2230	<code>\skip_if_eq:nnTF</code>	115, 124, 129, 146
<code>\prop_put:Nnn</code>	2002, 2003, 2004, 2005, 2006, 2093, 2095, 2122	<code>\skip_if_eq:p:nn</code>	562, 622, 623
<code>\prop_put_if_new:Nnn</code>	2124	<code>\skip_new:N</code>	32, 675, 703, 2763
<code>\ProvideTextCommandDefault</code>	2944	<code>\skip_set:Nn</code>	171, 2703, 2729, 3466
<code>\PunctStyle</code>	1766	<code>\skip_set_eq:NN</code>	1025, 3461
<code>\punctstyle</code>	2923, 2923	<code>\skip_use:N</code>	171, 2703
<code>\PunctWidth</code>	1320	<code>\skip_vertical:n</code>	3516
Q			
<code>\q_stop</code>	85, 86, 89, 233, 236, 241	<code>\SlantedFeatures</code>	2001
<code>\quiet</code>	2826	<code>\SlantedFont</code>	2001
R			
<code>\r</code>	3101, 3102	<code>\SlantFactor</code>	1914
<code>\RenewDocumentCommand</code>	2995, 3114, 3468, 3486	<code>\SplitArgument</code>	438
<code>\RenewDocumentEnvironment</code>	3532	<code>\str_case:nnn</code>	3000
<code>\RequirePackage</code>	21, 24, 2884, 2886, 3154, 3161, 3164, 3172, 3181, 3197, 3198, 3199, 3559, 3560	<code>\str_case_x:nnn</code>	1161, 2428, 2439, 2552
<code>\reverse_if:N</code>	183, 3674, 3690, 3697	<code>\str_case_x:non</code>	1155
<code>\rmdefault</code>	2554, 3002	<code>\str_if_eq:nnF</code>	429, 537, 578, 594, 2328
S			
<code>\scan_stop:</code>	72, 170, 188, 774, 1159, 1541, 1563, 1573, 1640, 1663, 2772, 3180, 3196, 3558	<code>\str_if_eq:nnT</code>	17, 2150
<code>\selectfont</code>	2366, 2786, 2798	<code>\str_if_eq:nnTF</code>	533, 1273, 1955, 1993, 2175, 2331, 2949, 3270
<code>\seq_clear:N</code>	2046	<code>\str_if_eq_p:nn</code>	1738, 1739, 2169, 2182, 2184
<code>\seq_count:N</code>	3053	<code>\str_if_eq_x:nnTF</code>	81, 234, 2568
<code>\seq_gclear:c</code>	1349	<code>\str_if_eq_x_p:nn</code>	3106, 3107
<code>\seq_gput_right:cn</code>	1353, 1363	<code>\sw@slant</code>	3014
<code>\seq_gput_right:Nn</code>	278, 289, 412, 1294, 1794, 1942, 2414	T	
<code>\seq_gput_right:Nv</code>	279	<code>\tex_baselineskip:D</code>	2858
<code>\seq_gremove_all:cn</code>	1372	<code>\tex_char:D</code>	3180
<code>\seq_gset_eq:NN</code>	405	<code>\tex_font:D</code>	104, 108, 117, 118, 119, 127, 133, 134, 139, 140, 149, 1727, 2754, 2755, 2779, 2787, 2794
<code>\seq_gset_from_clist:Nn</code>	407	<code>\tex_fontdimen:D</code>	117, 118, 119, 127, 133, 134, 139, 140, 149, 2754, 2779, 2787, 2794
<code>\seq_if_in:cnF</code>	1360	<code>\tex_futurelet:D</code>	3686
<code>\seq_if_in:NnF</code>	999, 1187, 2412, 3057	<code>\tex_global:D</code>	2635
<code>\seq_map_function:NN</code>	524, 2509	<code>\tex_hbox:D</code>	3539, 3552
<code>\seq_map_inline:cn</code>	1347	<code>\tex_hfil:D</code>	3535
<code>\seq_map_inline:Nn</code>	531, 746, 1012, 1203, 1269, 1285, 1732, 2047, 2234, 2244, 3065, 3266, 3268, 3592, 3607	<code>\tex_hss:D</code>	3520, 3702
<code>\seq_new:c</code>	1344	<code>\tex_iftrue:D</code>	3153
<code>\seq_new:N</code>	269, 270, 404, 406, 409, 988, 1199, 1209, 1797, 1939, 2420	<code>\tex_ignorespaces:D</code>	72, 759, 763, 2396, 2479, 2501, 2917, 3483, 3496, 3540, 3553
<code>\seq_put_right:Nn</code>	999, 1188, 1954	<code>\tex_indent:D</code>	3537
<code>\seq_remove_all:Nn</code>	1004, 1195	<code>\tex_italiccorrection:D</code>	586, 587
<code>\seq_set_from_clist:Nn</code>	1180	<code>\tex_kern:D</code>	660, 661, 786, 3519
<code>\seq_set_split:Nnn</code>	993	<code>\tex_lastkern:D</code>	649, 3464
<code>\seq_use:Nnnn</code>	1788	<code>\tex_lastskip:D</code>	171, 562, 622, 623, 1025, 1027
<code>\setCJKfallbackfamilyfont</code>	6, 7, 1883, 1883	<code>\tex_let:D</code>	3135
<code>\setCJKfamilyfont</code>	5, 2434, 2468, 2468	<code>\tex_noindent:D</code>	3571
<code>\setCJKmainfont</code>	5, 2430, 2447, 2447, 2452, 2463	<code>\tex_number:D</code>	164
<code>\setCJKmathfont</code>	6, 2464, 2599, 2599	<code>\tex_output:D</code>	2715, 2720, 3480, 3501, 3502, 3574, 3579
<code>\setCJKmonofont</code>	5, 2432, 2447, 2458, 2466	<code>\tex_penalty:D</code>	44, 45, 3217, 3224
<code>\setCJKromanfont</code>	2452, 2467	<code>\tex_rightskip:D</code>	2703, 2729
<code>\setCJKsansfont</code>	5, 2431, 2447, 2453, 2465	<code>\tex_romannumeral:D</code>	201, 225
<code>\SetSymbolFont</code>	2616	<code>\tex_space:D</code>	3216, 3223
<code>\sfdefault</code>	2555, 3003	<code>\tex_spacefactor:D</code>	585
<code>\shapedefault</code>	2615, 2617	<code>\tex_spaceskip:D</code>	115, 121, 124, 144, 149, 153, 154
<code>\silent</code>	2826	<code>\tex_the:D</code>	104, 493, 2720, 3501, 3579
<code>\skip_add:Nn</code>	1027	<code>\tex_unkern:D</code>	727, 3021, 3027, 3033
<code>\skip_gset:Nn</code>	2760	<code>\tex_unskip:D</code>	565, 626, 1028, 3543, 3550
<code>\skip_horizontal:N</code>	809, 1032, 2724, 3218, 3225, 3321	<code>\tex_vbox:D</code>	3538
		<code>\tex_vrule:D</code>	771
		<code>\tex_vss:D</code>	3522
		<code>\tex_xspaceskip:D</code>	129, 137, 146, 151
		<code>\textcentereddot</code>	3093
		<code>\textellipsis</code>	3093
		<code>\textemdash</code>	3093

\textperiodcentered	3093
\textquotedblleft	3094
\textquotedblright	3094
\textquoteleft	3094
\textquoteright	3094
\textvisiblespace	2768
\tipaencoding	3100
\tl_case:Nnn	982
\tl_case:Non	978, 1141
\tl_clear:c	2042
\tl_clear:N	92, 1011, 1202, 1896
\tl_concat:NNN	1166, 1173
\tl_const:cn	413
\tl_const:Nn	111, 1318, 1319, 1783, 2601, 3188
\tl_const:Nx	51, 2613, 2885
\tl_gclear:N	93
\tl_gput_right:Nn	56, 58, 60
\tl_gset:cn	1314, 1317
\tl_gset:cx	1400, 1402, 1404, 2749
\tl_gset:Nn	2572, 2576, 2809, 2899, 2900, 2901
\tl_gset:Nx	830, 840, 853, 876, 888, 897, 919, 928, 937, 946, 2550, 2814, 2904
\tl_gset_eq:cN	2488
\tl_gset_eq:NN	2913
\tl_if_blank:nTF	1470, 1947, 2086, 2172
\tl_if_blank:VT	1977
\tl_if_blank:VTF	1984
\tl_if_blank:vTF	2163
\tl_if_empty:Nf	511, 517
\tl_if_empty:nTF	242, 2955
\tl_if_eq:NNT	2548
\tl_if_eq:NNTF	1698, 1757
\tl_if_exist:cF	1696, 1755, 2484, 2742
\tl_if_exist:cTF	48
\tl_if_exist:Nf	2899, 2900, 2901
\tl_if_exist:NTF	2908
\tl_if_exist_p:c	1587
\tl_if_head_eq_meaning:nNTF	1998
\tl_if_head_eq_meaning:VNTF	1974
\tl_if_head_is_group_p:n	88
\tl_if_single_p:n	88
\tl_if_single_token_p:n	1064
\tl_map_inline:nn	471, 998, 1004, 1315, 2288, 3091
\tl_map_inline:xn	1314, 1350, 1358, 1369
\tl_new:c	1352, 1362
\tl_new:N	25, 26, 52, 53, 54, 787, 987, 1151, 1206, 2304, 2398, 2818, 2902, 3510, 3613
\tl_put_left:Nn	2772, 3100, 3573
\tl_put_right:Nn	1012, 1204, 1899, 1951, 2651
\tl_put_right:Nv	3726
\tl_put_right:Nx	3594
\tl_replace_all:Nnn	519, 1999
\tl_replace_all:Nnv	1986
\tl_replace_once:Nnn	3013, 3014, 3237, 3239
\tl_set:cn	3137
\tl_set:Nn	189, 190, 216, 1893, 1995, 2062, 2064, 2305, 3176, 3183, 3184
\tl_set:Nx	83, 510, 516, 1476, 1477, 1779, 1972, 2246, 2311, 2320, 2341, 2364, 2365, 2389, 2472, 2483, 2486, 2605, 2608, 2777, 2784, 2785, 2997, 3500, 3584
\tl_set_eq:cc	3138, 3139
\tl_set_eq:cN	1988
\tl_set_eq:NN	1775, 1979, 1983, 1985, 1994, 2390, 2537, 2567
\tl_set_rescan:Nnn	3724, 3737
\tl_to_lowercase:n	3744
\tl_to_str:n	2483
\tl_trim_spaces:n	90
\tl_use:c	1244, 1396, 1398, 1589
\token_get_arg_spec:N	1063
\token_if_active:Nf	3709
\token_if_cs:Nf	1091, 3712
\token_if_letter:Nf	3621, 3627
\token_if_letter_p:N	1066
\token_if_macro_p:N	731
\token_if_math_toggle_p:N	1103
\token_if_other:Nf	1058, 1083
\token_if_other_p:N	733, 1065
\token_if_space:Nf	589
\token_to_meaning:N	233, 234
\token_to_str:N	234, 1253, 1304, 2430, 2431, 2432, 2434, 2441, 2442, 2443, 2524, 2896, 2897, 2942, 2945, 2957
\TrimSpaces	417, 1211, 1790, 1804
\ttdefault	2556, 3004
U	
\UL@hook	3200
\UL@leaders	3461, 3466
\UL@leadtype	3208, 3244, 3246
\UL@skip	3461, 3466
\UL@start	3297, 3314, 3333, 3345, 3358, 3379, 3407, 3429, 3443, 3457, 3476, 3482
\UL@stop	3297, 3314, 3330, 3341, 3354, 3367, 3401, 3404, 3426, 3439, 3457, 3470, 3478
\ULon	3237, 3238, 3239, 3240
\Umathcode	234, 238
PunctStyle	4
PunctWidth	4
\Unicode	3179, 3183, 3184
\use:c	416, 649, 660, 661, 1556, 1557, 1559, 1569, 1570, 1651, 1652, 1654, 1917, 1925, 2358, 2618, 2761
\use:n	1171, 1490, 1494, 1521, 1531, 1622, 1655, 1661, 2569, 2719, 2903, 3010, 3028, 3034, 3578
\use:x	2746, 3145
\use_i:nn	1157, 2373, 3289
\use_ii:nn	1204, 1656, 3287
\use_iii:nnn	1164
\use_iv:nnn	1143
\use_none:n	183, 184, 185, 1487, 1491, 1660, 2101, 3036
\use_none:nn	137, 151, 1652, 2381
\use_none:nnnnn	2609
\usefont	3115
\UseInstance	1709, 1760
\usepackage	2896
AutoFakeBold	4, 5
AutoFakeSlant	4, 5
AutoFallBack	4
V	
\vbox_gset_to_ht:Nnn	3514
\Verb	2640
\verbatim@font	2651
X	
\xCJKecglue	676
\xe@alloc@intercharclass	3053, 3055

\xeCJK@family	2390	\xeCJK_FullLeft_and_CJK:	743, 788, 788, 3262
\xeCJK@fix@penalty	3012, 3012, 3013, 3014	\xeCJK_FullLeft_and_Default:	798, 798, 3261, 3415
\xeCJK@italiccorr	3013, 3015, 3015	\xeCJK_FullLeft_and_FullLeft:N	913, 913
\xeCJK@setfont	2325	\xeCJK_FullLeft_and_FullRight:N	922, 922
\xeCJK@text@composite	2947, 2947, 2960	\xeCJK_FullRight_and_Boundary:	759, 761, 762, 762, 955, 957, 959, 965, 967
\xeCJK_add_font_features:n	2504, 2504, 2521	\xeCJK_FullRight_and_CJK:	745, 810, 810, 3264
\xeCJK_add_font_features:x	2500	\xeCJK_FullRight_and_Default:	763, 816, 816, 3263
\xeCJK_allow_break:	44, 44, 1324	\xeCJK_FullRight_and_FullLeft:N	932, 932
\xeCJK_app_inter_class_toks:nnn	502, 502, 507, 579, 759	\xeCJK_FullRight_and_FullRight:N	941, 941
\xeCJK_app_inter_class_toks:nxx	3069	\xeCJK_FullRight_symbol:N	879, 891, 900, 930, 948, 958, 960, 968, 973, 973
\xeCJK_Boundary_and_FullLeft:N	756, 846, 846	\xeCJK_get_inter_class_toks:nn	492, 492, 499, 505, 510, 516, 3070, 3072, 3074, 3080, 3617
\xeCJK_Boundary_and_FullRight:N	758, 881, 881	\xeCJK_get_punct_bounds:NN	824, 838, 848, 870, 883, 895, 916, 925, 934, 943, 976, 1694, 1694, 3399
\xeCJK_calc_punct_dimen:f	1705	\xeCJK_get_punct_kerning:NN	1753, 1753, 1765
\xeCJK_calc_punct_dimen:N	1715, 1715, 1747	\xeCJK_get_punct_kerning:oN	917, 926, 935, 944
\xeCJK_check_for_glue:	599, 605, 605, 2694, 2711, 2727	\xeCJK_glue_to_skip:nN	168, 168, 672, 681, 698, 3214, 3221
\xeCJK_check_FullRight:	955, 959, 965, 974, 974	\xeCJK_glyph_bounds:NN	1718, 1720, 1748, 1748
\xeCJK_check_FullRight_symbol:Nw	960, 983, 983	\xeCJK_glyph_if_exist:N	106
\xeCJK_check_single:NNw	1056, 1059, 1068, 1073, 1073	\xeCJK_glyph_if_exist:NTF	106, 1856, 1865
\xeCJK_check_single:Nw	1039, 1042, 1047, 1053, 1053	\xeCJK_glyph_if_exist_p:N	106
\xeCJK_check_single_cs:NNn	1095, 1096, 1138, 1138	\xeCJK_gset_mathcode:nnnn	2625, 2630, 2630
\xeCJK_check_single_env:nnNn	1144, 1152, 1152	\xeCJK_hook_for_ulem:	3200, 3201, 3201
\xeCJK_check_single_equation:NNnNw	1107, 1108, 1129, 1129	\xeCJK_if_blank_x:n	173
\xeCJK_CJK_and_CJK:N	740, 741, 741, 1039, 1041, 1042, 1047, 1048, 3265	\xeCJK_if_blank_x:nT	3073, 3079
\xeCJK_CJK_and_FullLeft:N	836, 836	\xeCJK_if_blank_x:nTF	173, 445, 1489, 2212, 2379
\xeCJK_CJK_and_FullRight:N	893, 893	\xeCJK_if_blank_x_p:n	173, 443, 1063
\xeCJK_class_group_begin:	527, 527, 546, 600, 826, 850, 872, 885, 3305, 3315	\xeCJK_if_CJK_class:N	245
\xeCJK_class_group_end:	527, 528, 552, 713, 803, 806, 819, 980, 3296, 3313, 3353, 3387, 3401, 3404, 3425, 3438	\xeCJK_if_CJK_class:NTF	245, 1121, 1123
\xeCJK_class_num:n	416, 416, 432, 435, 465, 472, 490, 493, 1377, 2734	\xeCJK_if_CJK_class_p:N	245
\xeCJK_clear_Boundary_and_CJK_toks:	523, 523, 549, 601, 829, 852, 875, 887, 3306, 3316	\xeCJK_if_last_node:n	647
\xeCJK_clear_inter_class_toks:nn	494, 494, 526, 548, 828, 874	\xeCJK_if_last_node:nT	575
\xeCJK_copy_inter_class_toks:nnnn	508, 508, 534, 536, 538, 1271, 1272, 1284, 1287, 1288, 3067, 3068, 3076, 3082	\xeCJK_if_last_node:nTF	574, 647, 3019, 3025, 3031
\xeCJK_cs_case_keys_define:nnNnn	985, 985, 1015, 1162, 1169	\xeCJK_if_last_node_p:n	568, 569, 608, 613, 629, 631, 637, 638, 647
\xeCJK_cs_clear:N	92, 92, 3125, 3126, 3127, 3210, 3211, 3244, 3247, 3701	\xeCJK_if_package_loaded:n	46
\xeCJK_cs_gclear:N	92, 93	\xeCJK_if_package_loaded:nF	2832, 2839, 3158
\xeCJK_declare_char_class:nn	422, 422, 440, 1264	\xeCJK_if_package_loaded:nT	2983, 3112, 3120, 3163, 3171
\xeCJK_declare_char_class:nV	476, 477, 478, 479, 483, 484, 485	\xeCJK_if_package_loaded:nTF	46, 61, 2844, 2967, 2973, 3160
\xeCJK_declare_char_class:nx	419, 1242	\xeCJK_if_package_loaded_p:n	46, 3134
\xeCJK_declare_sub_char_class:nnn	1256, 1256, 1266	\xeCJK_if_same_class:NN	253
\xeCJK_declare_sub_char_class:nxx	1213	\xeCJK_if_same_class:NNTF	253, 1629
\xeCJK_def_node:nn	652, 652, 663, 664, 665, 666	\xeCJK_if_same_class_p:NN	253
\xeCJK_Default_and_FullLeft:nN	822, 822	\xeCJK_if_ulem_patch:TF	3284, 3284, 3294, 3311, 3328, 3339, 3351, 3365, 3385, 3395, 3413, 3422, 3435, 3450, 3456
\xeCJK_Default_and_FullRight:nN	868, 868	\xeCJK_ignore_spaces:w	715, 717, 717, 2695, 2712, 2728, 3043, 3530
\xeCJK_fallback_loop:Nn	1858, 1860, 1860, 1867	\xeCJK_int_until_do:nn	182, 182, 457, 2633
\xeCJK_fallback_test_glyph:N	1840, 1843, 1848, 1854, 1854	\xeCJK_inter_class_toks:nnc	752, 767
\xeCJK_family_if_exist:x	2370	\xeCJK_inter_class_toks:nnn	489, 489, 491, 495, 544, 552, 556, 583, 597, 711, 740, 742, 744, 755, 757, 760, 3272, 3274, 3278, 3597, 3599, 3601, 3603, 3605, 3609, 3616
\xeCJK_family_if_exist:xF	2343, 2565	\xeCJK_inter_class_toks:nxx	498, 504, 512, 520, 750
\xeCJK_family_if_exist:xT	1870, 2273, 2286	\xeCJK_make_node:n	587, 590, 591, 652, 658, 714, 728, 3022, 3028, 3034, 3529
\xeCJK_family_if_exist:xTF	1862, 2370, 2387, 2404, 2571, 2604, 2607	\xeCJK_new_class:n	271, 271, 304, 305, 306, 307, 1260
\xeCJK_family_if_exist_use:x	2382, 2385, 2402, 2402	\xeCJK_new_sub_key:n	1262, 1939, 1940, 2000
\xeCJK_font_gset_to_current:c	101, 101, 2313, 2322, 2353, 2780	\xeCJK_no_break:	44, 45, 793, 796, 803, 806, 908, 909, 915, 924, 929, 947, 1016, 1030, 1133, 1134, 1146, 1147, 1326, 3374, 3377, 3401, 3406
\xeCJK_fontspec:nn	2481, 2481, 2493		
\xeCJK_fontspec:VV	2517		
\xeCJK_fontspec:xx	2478		

\xeCJK_peek_after_ignore_spaces:nw	214, 214, 984	\xeCJKcaption	3140, 3142
\xeCJK_peek_catcode_ignore_spaces:NTF	186, 186, 719, 1075	\xeCJKDeclareCharClass	9, 417, 417
\xeCJK_pre_inter_class_toks:nnn	496, 496, 501, 595, 1275, 1289, 1291, 1298	\xeCJKDeclarePunctStyle	10, 1790, 1790, 1803, 1811, 1812, 1813, 1820, 1828
\xeCJK_pre_inter_class_toks:nxx	3071	\xeCJKDeclareSubCJKBBlock	8, 1210, 1210, 1216, 1253
\xeCJK_punct_kerning_process:NN	1472, 1578, 1578	\xeCJKdisablefallback	2933, 2935
\xeCJK_punct_margin_process:NN	1471, 1474, 1474	\xeCJKEditPunctStyle	10, 1804, 1804, 1810
\xeCJK_replace_inter_class_toks:nnnn	514, 514, 1279	\xeCJKenablefallback	2933, 2933
\xeCJK_reverse:nnn	80, 80, 1480, 2209	\xeCJKnobreak	11, 1017, 1017
\xeCJK_save_class:nn	282, 282, 299, 300, 301, 302, 303	\xeCJKnobreakbetweenpuncts	2929, 2931
\xeCJK_select_font:	547, 602, 827, 851, 873, 886, 1705, 2304, 2307, 2325, 2332, 2745, 3307	\xeCJKOffVerbAddon	11, 2669, 2681, 2696, 2718
\xeCJK_select_font:x	1864, 2304, 2316	\xeCJKplainchr	2923, 2924
\xeCJK_set_char_class:nnn	431, 435, 453, 453, 2939	\xeCJKResetCharClass	9, 481, 481, 488
\xeCJK_set_family:nnn	2058, 2058, 2078	\xeCJKResetPunctClass	9, 420, 474, 474, 486, 1214, 1224, 1233, 2940
\xeCJK_set_family:Vnn	2489	\xeCJKRestoreSubCJKBBlock	9, 1217, 1227
\xeCJK_set_family:Vvv	1903, 2250	\xeCJKsetcharclass	2937, 2937
\xeCJK_set_family:xxx	1888, 2449, 2455, 2460, 2469, 2474, 2600	\xeCJKsetecglue	2926
\xeCJK_set_family_fallback:nnN	1890, 1890, 1909	\xeCJKseteboldenfactor	2919, 2919
\xeCJK_set_family_fallback:xxN	1887	\xeCJKsetkern	9, 1316, 1316
\xeCJK_set_mathfont:	2582, 2602, 2602	\xeCJKsetslantfactor	2919, 2921
\xeCJK_set_visible_space_font:	2767, 2775, 2775	\xeCJKsetup	2, 1311, 2722, 2897, 2914, 2914, 2920, 2922, 2923, 2924, 2925, 2927, 2928, 2930, 2932, 2934, 2936, 3206
\xeCJK_swap_cs:NN	94, 94, 3250, 3252, 3254, 3256, 3258, 3260, 3261, 3262, 3263, 3264, 3265, 3748	\xeCJKsetwidth	9, 1313, 1313
\xeCJK_tl_remove_outer_braces:N	82, 82	\xeCJKVerbAddon	11, 2658, 2665, 2669, 2669, 2739
\xeCJK_tl_remove_outer_braces:n	82, 83, 84, 1972	\xetex_if_engine:F	10
\xeCJK_token_value_charcode:N	231, 232, 232, 1727, 1751	\XeTeXcharclass	231, 459, 467, 472, 2736
\xeCJK_token_value_class:N	230, 230, 247, 256, 1377	\XeTeXcharglyph	1751
\xeCJK_visible_space_fallback:	2764, 2764, 2773	\XeTeXdefaultencoding	3144, 3151
\xeCJK_xetex_mathcode:w	238, 243, 2635	\XeTeXglyphbounds	1304, 1308, 1750
\xeCJKactive	259	\XeTeXinterchartokenstate	266, 267, 3017
\xeCJKallowbreakbetweenpuncts	2929, 2929	\XeTeXinterchartoks	490, 493
\xeCJKCancelSubCJKBBlock	9, 1217, 1218	\XeTeXmathcode	243