

xeCJK 宏包

ctex.org

2013/11/11 v3.2.7*

目录

1	简介	1	5.5	字符类别处理	24
2	基本用法	2	5.6	字符输出规则	27
3	用户手册	2	5.7	全角右标点后的断行	38
3.1	宏包选项	2	5.8	段末孤字处理	40
3.2	字体设置与选择	5	5.9	增加 CJK 子分区	43
3.2.1	X _Y TeX 的字体名查找	7	5.10	标点处理	45
3.3	CJK 分区字体设置	8	5.11	后备字体	56
3.4	设置 CJK 字符范围	9	5.12	CJK 字体族声明方式	57
3.5	标点符号的处理	9	5.13	字体切换	63
3.5.1	设置特定标点符号的宽度和间距	9	5.14	数学字体设置	69
3.5.2	定义标点符号处理格式	10	5.15	抄录环境中的间距调整	70
3.6	其它	12	5.16	xeCJK 其它选项	74
4	已知问题和兼容性	12	5.17	xeCJK 初始化设置	75
5	xeCJK 代码实现	13	5.18	兼容性修补	77
5.1	运行环境检查	13	5.19	xeCJKfntef	83
5.2	内部工具	14	5.20	xeCJK-listings	91
5.3	功能开关	19	5.21	xunicode-addon	96
5.4	字符类别设定	19	5.22	xeCJK.cfg	103
			版本历史	104	
			代码索引	105	

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 [everypage](#) 和 [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 `indentfirst` 选项需要 `indentfirst` 宏包的支持。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](#) rev525.

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 \XeLaTeX 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, \dots }</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 `mktextlsr`,刷新文件名数据库以便 \TeX 系统能够找到它。

<hr/> indentfirst ★ <hr/>	<code>indentfirst = <true false></code>
Updated: 2012-11-22	是否使用 <code>indentfirst</code> 宏包,使得跟在章节标题后面的第一段首行也缩进。
<hr/>	
<hr/> xeCJKactive <hr/>	<code>xeCJKactive = <true false></code>
	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/>	
<hr/> CJKspace <hr/>	<code>CJKspace = <true false></code>
	缺省状态下, <code>xeCJK</code> 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/>	
<hr/> CJKmath ★ <hr/>	<code>CJKmath = <true false></code>
	是否支持在数学环境中直接输入 CJK 字符。使用这个选项后,可以直接在数学环境中输出 CJK 字符。 <code>url</code> 宏包将一个 URL 放在一个特殊的数学环境中排版,所以如果在 <code>\path</code> 等命令的路径参数中含有汉字,则需要启用这个选项,路径中的汉字才能显示。
<hr/>	
<hr/> CJKglue <hr/>	<code>CJKglue = {\hskip 0pt plus 0.08\baselineskip}</code>
	设置 CJK 文字之间插入的 <code>glue</code> ,上边是 <code>xeCJK</code> 的默认值。一般来说,除非有特殊需要(例如,改变文字间距等),否则不需要设置这个选项,使用默认值即可。如果要设置这个选项,为了行末的对齐,设置的 <code>glue</code> 最好有一定的弹性。
<hr/>	
<hr/> CJKecglue <hr/>	<code>CJKecglue = {\glue}</code>
	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 <code>glue</code> 最好也要用一定的弹性。请注意,这里设置的 <code>glue</code> 只影响 <code>xeCJK</code> 根据需要自动添加的空白,源文件中直接输入的 CJK 文字与西文之间的空格不受影响(直接输出)。有时候 <code>xeCJK</code> 可能不能正确地调整间距,需要手动加空格。
<hr/>	
<hr/> xCJKecglue <hr/>	<code>xCJKecglue = {\<true false glue>}</code>
	缺省状态下, <code>xeCJK</code> 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 <code>CJKecglue</code> 替换源文件中直接输入的 CJK 文字与西文之间的空格。
<hr/>	
<hr/> CheckSingle <hr/>	<code>CheckSingle = <true false></code>
Updated: 2013-06-26	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/>	
<hr/> PlainEquation <hr/>	<code>PlainEquation = <true false></code>
New: 2012-12-06	如果使用了 <code>\$\$...\$\$</code> 的形式来输入行间数学公式,就需要启用本选项,以便 <code>CheckSingle</code> 选项能够正确识别。推荐使用 <code>\[...\]</code> 的形式来输入行间数学公式。
<hr/>	
<hr/> NewLineCS <hr/>	<code>NewLineCS = { \par \[}</code>
NewLineCS+ NewLineCS-	设置造成断行的控制序列,以便 <code>CheckSingle</code> 选项能够正确识别。以上是 <code>xeCJK</code> 的初始设置。
New: 2012-12-04	
<hr/>	
<hr/> EnvCS <hr/>	<code>EnvCS = { \begin \end }</code>
EnvCS+ EnvCS-	设置 \LaTeX 环境开始和结束的控制序列,以便 <code>CheckSingle</code> 选项能够正确识别。以上是 <code>xeCJK</code> 的初始设置。
New: 2012-12-04	

InlineEnv	InlineEnv = { $\langle env_1 \rangle$, $\langle env_2 \rangle$, $\langle env_3 \rangle$, ...}
InlineEnv+	在使用 <code>CheckSingle</code> 选项的时候, <code>xeCJK</code> 会将 <code>CJK</code> 文字后接着的 <code>LaTeX</code> 环境的开始 <code>\begin{...}</code> 和结束 <code>\end{...}</code> 视为断行的地方, 如果有某些特殊的 <code>LaTeX</code> 环境没有造成断行, 可以使用这个选项来声明它, 以便 <code>CheckSingle</code> 能正确识别。
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 = { $\langle true false 数字 \rangle$ }
	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 = { $\langle true false 数字 \rangle$ }
	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 = { $\langle 数字 4 \rangle$ }
	设置伪粗体的默认粗细程度。
SlantFactor ☆	SlantFactor = { $\langle 数字 0.167 \rangle$ }
	设置伪斜体的粗细程度, 范围是 $-0.999 \sim 0.999$ 。
PunctStyle	PunctStyle = { $\langle quanjiao banjiao kaiming hangmobanjiao CCT plain ... \rangle$ }
Updated: 2012-11-10	设置标点处理格式。 <code>xeCJK</code> 中预先定义好的格式为
	<div> <div>quanjiao</div> <div>banjiao</div> <div>kaiming</div> <div>hangmobanjiao</div> <div>CCT</div> <div>plain</div> </div> <div> 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; 半角式: 所有标点占半个汉字宽度; 开明式: 句末点号用全角, 其他半角; 行末半角式: 所有标点占一个汉字宽度, 行首行末对齐; CCT 格式: 所有标点符号的宽度略小于一个汉字宽度; 原样(不调整标点间距)。 </div>
	可以使用 3.5.2 中介绍的 <code>\xeCJKDeclarePunctStyle</code> 定义新的标点格式。
KaiMingPunct ☆	KaiMingPunct = { $\langle . . ? ! \rangle$ }
KaiMingPunct+	设置开明(<code>kaiming</code>)标点处理格式时的句末点号, <code>KaiMingPunct</code> 后带的 <code>+</code> 与 <code>-</code> 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 <code>CJK</code> 文字之间的居中标点, <code>xeCJK</code> 会根据不同的标点处理格式, 调整居中标点与前后文字之间的空白, 保证其确实居中。对于行末出现的居中标点, 允许在其后面断行, 但禁止在它前面断行。
MiddlePunct-	
PunctWidth ☆	PunctWidth = { $\langle length \rangle$ }
	缺省状态下, <code>xeCJK</code> 会根据所选择的标点处理格式自动计算标点所占的宽度, 如果对缺省设置不满意, 可以通过这一选项来改变它。为了使得标点所占的宽度能够适应字体大小的变化, 这里设置的 <code>length</code> 的单位最好用 <code>em</code> 等相对距离单位, 而不建议使用诸如 <code>pt</code> 之类的绝对距离单位。这里的设置可用于除了 <code>plain</code> 以外的所有标点处理格式。同时, 这里的设置对所有的 <code>CJK</code> 标点都生效, 如果只要设置部分标点, 请使用 3.5.1 节的 <code>\xeCJKsetwidth</code> 。

<hr/> PunctBoundWidth ★ <hr/>	<code>PunctBoundWidth = {<length>}</code>
<hr/> New: 2013-08-22 <hr/>	与以上选项类似,但设置的是标点符号出现在行首/尾时的宽度。
<hr/> AllowBreakBetweenPuncts <hr/>	<code>AllowBreakBetweenPuncts = <true false></code>
	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。
<hr/> CheckFullRight <hr/>	<code>CheckFullRight = <true false></code>
<hr/> New: 2012-12-02 <hr/>	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。
<hr/> NoBreakCS NoBreakCS+ NoBreakCS- <hr/>	<code>NoBreakCS = { \footnote \footnotemark \nobreak }</code>
<hr/> New: 2012-12-02 <hr/>	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 <code>CheckFullRight</code> 选项,而是手工在这些控制序列前面加上 3.6 节介绍的 <code>\xeCJKnobreak</code> 。
<hr/> Verb <hr/>	<code>Verb = <true false env env+></code>
<hr/> Updated: 2013-05-29 <hr/>	<code>true</code> 表示在 <code>\verb</code> 命令或 <code>verbatim</code> 环境里不自动调整中英文之间的间距。 <code>env</code> 选项在 <code>verbatim</code> 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; <code>env</code> 选项不调整 <code>\verb</code> 里的间距, <code>env+</code> 选项还将正文里设置的间距应用到 <code>\verb</code> 里。这个选项对使用到 <code>\verbatim@font</code> 命令的情形均有效,更一般的情况可以使用 3.6 节介绍的 <code>\xeCJKVerbAddon</code> 。

3.2 字体设置与选择

<hr/> \setCJKmainfont ★ <hr/>	<code>\setCJKmainfont [<i></i>] {}</code>
	设置正文罗马族的 CJK 字体,影响 <code>\rmfamily</code> 和 <code>\textrm</code> 的字体。后面两个参数继承自 <code>fontspec</code> 宏包, <code></code> 表示字体属性选项, <code></code> 是字体名。字体名可以是字体族名,也可以是字体的文件名,查找字体名见 3.2.1 节;可用的字体属性选项参见 <code>fontspec</code> 宏包的文档。需要说明的是 xeCJK 修改了 <code>AutoFakeBold</code> 和 <code>AutoFakeSlant</code> 选项,以便配合全局伪粗体和伪斜体的设定。
<hr/> AutoFakeBold AutoFakeSlant <hr/>	<code>AutoFakeBold = {<true false 数字>}</code> <code>AutoFakeSlant = {<true false 数字>}</code>
	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。
<hr/> Mapping <hr/>	<code>Mapping = {<fullwidth-stop full-stop han-trad han-simp ...>}</code>
<hr/> New: 2013-06-07 <hr/>	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 <code>Mapping</code> 选项来使用它们。其中 <code>fullwidth-stop</code> 用于将正常句号“。”转换成全角实心句号“。”, <code>full-stop</code> 的作用相反。 <code>han-trad</code> 用于将简体中文转换成繁体中文, <code>han-simp</code> 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。
<hr/> \setCJKsansfont ★ <hr/>	<code>\setCJKsansfont [<i></i>] {}</code>
	设置正文无衬线族的 CJK 字体,影响 <code>\sffamily</code> 和 <code>\textsf</code> 的字体。
<hr/> \setCJKmonofont ★ <hr/>	<code>\setCJKmonofont [<i></i>] {}</code>
	设置正文等宽族的 CJK 字体,影响 <code>\ttfamily</code> 和 <code>\texttt</code> 的字体。
<hr/> \setCJKfamilyfont ★ <hr/>	<code>\setCJKfamilyfont {<family>} [<i></i>] {}</code>
	声明新的 CJK 字体族 <code><family></code> 并指定字体。

<code>\CJKfamily</code>
Updated: 2012-10-27

```
\CJKfamily {⟨family⟩}
\CJKfamily+ {⟨family⟩}
\CJKfamily- {⟨family⟩}
```

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。

`\newCJKfontfamily` ★

```
\newCJKfontfamily [⟨family⟩] \⟨font-switch⟩ [⟨font features⟩] {⟨font name⟩}
```

声明新的 CJK 字体族 $\langle family \rangle$ 并指定字体, 并定义 $\langle font-switch \rangle$, 在文档中可以使用它来切换 CJK 字体族。可以不必指定 $\langle family \rangle$, 这时候 $\langle family \rangle$ 将等于 $\langle font-switch \rangle$ 。事实上, `\newCJKfontfamily` 是 `\setCJKfamilyfont` 和 `\CJKfamily` 的合并。例如

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

等价于

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

`\CJKfontspec`

```
\CJKfontspec [⟨font features⟩] {⟨font name⟩}
```

在文档中随机定义新的 CJK 字体族, 并马上使用它。

`\defaultCJKfontfeatures` ★

```
\defaultCJKfontfeatures {⟨font features⟩}
```

全局设置 CJK 字体族的默认选项。例如, 使用

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

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

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

`\addCJKfontfeatures`

Updated: 2013-06-30

```
\addCJKfontfeatures {⟨font features⟩}
\addCJKfontfeatures * {⟨font features⟩}
\addCJKfontfeatures [⟨block1, block2, ...⟩] {⟨font features⟩}
\addCJKfontfeatures * [⟨block1, block2, ...⟩] {⟨font features⟩}
```

临时增加当前使用的 CJK 字体的选项。第一条命令, 仅对当前 CJK 主分区字体有效; 第二条对主分区和其它分区的字体都有效; 第三条仅对可选参数中指定的分区有效; 第四条对主分区和可选参数中指定的分区有效。例如, 使用

```
\addCJKfontfeatures{Scale=1.1}
```

可以将文档中当前使用的 CJK 主分区字体放大为 1.1。

`\CJKrmdefault`

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`。

`\CJKsfdefault`

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`。

`\CJKttdefault`

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`。

`\CJKfamilydefault`

Updated: 2013-01-01

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

```
\renewcommand\familydefault{\sfdefault}
```

就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。

```
\setCJKmathfont ★ \setCJKmathfont [<font features>] {<font name>}
```

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。

```
\setCJKfallbackfamilyfont ★ \setCJKfallbackfamilyfont {<family>} [<font features>] {<font name>}
```

设置 CJK 字体族 <family> 的备用字体。例如,使用

```
\setCJKmainfont{SimSun}
\setCJKfallbackfamilyfont{\CJKrmdefault}{SimSun-ExtB}
```

可以将 SimSun-ExtB 作为 SimSun 的备用字体。

```
FallBack FallBack = { [<font features>] {<font name>} }
```

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

```
\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>} ,
  .....
}
```

Updated: 2013-06-30

\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。若当前字体族缺字,并没有备用字体,则尝试使用 \CJKfamilydefault 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,Itálico,Курсив,
İtalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta cursiva,tučné kurzíva,fed kursiv,
Fett Kursiv,Έντονα Πλάγια,Bold Italic,Negrita Cursiva,
Lihavoitu Kursivoi,Gras Italique,Félkövér dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrito Itálico,
```

Полужирный Курсив, Tučná kurzíva, Fet Kursiv, Kalın İtalik,
 Krepko poševno, nghiêng đậm, Lodi etzana
 Times New Roman: style=Negreta, tučné, fed, Fett, Έντονα, Bold, Negrita,
 Lihavoitu, Gras, Félkövér, Grassetto, Vet, Halvfet, Pogrubiona, Negrito,
 Полужирный, Fet, Kalın, Krepko, đậm, 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 * {\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

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。


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

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

例如,可以使用

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

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

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

```
\xeCJKCancelSubCJKBlock    {\block_1, block_2, ...}
\xeCJKCancelSubCJKBlock*   {\block_1, block_2, ...}
```

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

```
\xeCJKRestoreSubCJKBlock   {\block_1, block_2, ...}
\xeCJKRestoreSubCJKBlock*  {\block_1, block_2, ...}
```

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

3.4 设置 CJK 字符范围

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

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

```
\xeCJKResetCharClass    ★ 用于恢复 xeCJK 对各个字符类别的初始化设置。
```

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

```
\normalspacedchars     \normalspacedchars {\char list}
```

在 *<char list>* 中出现的字符两端不自动添加空格,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。对于标点符号的相关设置,只能在导言区中进行。

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 plain 以外的所有标点处理格式。

```
\xeCJKsetwidth    ★ \xeCJKsetwidth  {\标点列表} {\length}
                   \xeCJKsetwidth* {\标点列表} {\length}
```

Updated: 2013-08-22

<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。带星号的命令,设置标点符号出现在行首/尾时的宽度。

<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
对个别情况进行特殊调整,可以使用这个命令。例如,

```
\xeCJKsetkern{:}{ “}{0.3em}
```

将设置冒号与左双引号之间的空白宽度为 0.3 em。

3.5.2 定义标点符号处理格式

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

Updated: 2013-08-22

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

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

Updated: 2013-08-22

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 `nan`。

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

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

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

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

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

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

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

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

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

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

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

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

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

`middle-margin-ratio` *⟨real⟩* 设置居中标点的两边空白宽度之和与两边实际两边边界宽度之和的比例。默认值是与
`fixed-margin-ratio` 的值相同。

下面选项设置标点符号出现在行首或者行尾时的宽度或比例。

`bound-punct-width` *⟨length⟩* 设置标点符号出现在行首/尾时的宽度。默认值是 `\maxdimen`。

`bound-punct-ratio` *⟨real⟩* 设置标点符号出现在行首/尾时的输出宽度与实际宽度的比例。默认值是 `nan`。

`bound-margin-width` *⟨length⟩* 设置标点符号出现在行首/尾时的左/右空白宽度。默认值是 `\maxdimen`。

`bound-margin-ratio` *⟨real⟩* 设置标点符号出现在行首/尾时的左/右空白宽度与相应实际边界宽度的比例。默认值
是 0。

`enabled-hanging` *⟨boolean⟩* 当以上选项的计算结果得到的宽度小于标点符号的实际边界宽度时,是否允许标点符
号悬挂出页面边界。默认值是 `false`。

`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>* 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>* 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 **nan**。

different-align-margin *<length>* 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>* 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 **nan**。

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      = nan ,
    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 ,
    bound-punct-width      = 0 em ,
    enabled-hanging        = 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;当标点出现在行首或行尾时宽度为零,允许悬挂出页面边界;使用相邻两个标点的实际边界中的较小值作为它们之间的空白宽度,并且最小的空白宽度是 0.1 em。再例如,使用

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

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

3.6 其它

<code>\xeCJKVerbAddon</code>
<code>\xeCJKOffVerbAddon</code>
Updated: 2013-06-29

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。如果这两个空格的宽度小于当前 CJK 正常文字的宽度,将对 CJK 字体进行适当地缩小。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,将禁止在 CJK 字符类之间自动换行,这与西文在抄录环境中的情况是一致的。所以不应该单独使用,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。

<code>\xeCJKnobreak</code>
New: 2012-12-03

……汉字。`\xeCJKnobreak\footnote{脚注}`
`\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。

<code>\xeCJKShipoutHook</code>
New: 2013-11-09

`xeCJK` 在正文中的一些特殊设置(汉字下加点、在 `verbatim` 或 `lstlisting` 环境中分页)可能会影响到 `TEX` 的输出例行程序(output routine)中的内容(比如页眉和页脚)。`\xeCJKShipoutHook` 用于恢复正文中的普通设置。`xeCJK` 已经处理了页眉和页脚的情况,其它的就需要根据情况自行调用。比如若使用 `eso-pic` 或者 `atbegshi` 实现文字水印,并且正文中使用了以上所列的特殊形式,就需要在命令 `\AtBeginShipout` 的参数的前面使用 `\xeCJKShipoutHook`。

4 已知问题和兼容性

`XYTEX` 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。可以在 `xeCJK` 之后载入 `CJKnumb` 宏包,实现数字的中文化。

`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 `XYLATEX` 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 `XYTEX` 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 `LATEX3` 的语法来编写。`LATEX3` 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 `XYTEX` 的 `\XeTeXinterchartoks` 机制,与使用相同机制的宏包(例如 `polyglossia`)可能会存在大小不一的冲突。`xeCJK` 虽然为此作了一些处理,但与它们共同使用时应该小心。

5 xeCJK 代码实现

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

5.1 运行环境检查

xeCJK 必须使用 XeTeX 引擎的支持。

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

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

```
11 \prg_new_conditional:Nppn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
12 {
13   \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
14   { \prg_return_true: } { \prg_return_false: }
15 }
16 \tl_const:Nx \c__xeCJK_package_ext_tl { \@pkgextension }
```

(End definition for \xeCJK_if_package_loaded:n.)

下面这些 CJK 系列宏包不应该被使用。

```
17 \msg_new:nnn { xeCJK } { incompatible-package }
18 {
19   The~#1'~package~and~xeCJK~are~incompatible.\\
20   Please~do~not~use~it.
21 }
22 \msg_new:nnn { xeCJK } { after-package }
23 {
24   The~#1'~package~and~xeCJK~are~incompatible.\\
25   Please~load~it~after~xeCJK.
26 }
27 \clist_map_inline:nn { CJKfntef , CJKnumb }
28 {
29   \xeCJK_if_package_loaded:nT {#1}
30   { \msg_error:nnn { xeCJK } { after-package } {#1} }
31 }
32 \clist_map_inline:nn { CJKulem , CJKvert , CJKpunct , CJKutf8 , CJK }
33 {
34   \xeCJK_if_package_loaded:nTF {#1}
35   { \msg_error:nnn { xeCJK } { incompatible-package } {#1} }
36   { \tl_const:cn { ver@ #1 . \c__xeCJK_package_ext_tl } { 9999/99/99 } }
37 }
```

应该使用较新版本的 expl3 宏包。

```
38 \msg_new:nnn { xeCJK } { 13-too-old }
39 {
40   Support~package~#1'~too~old. \\
41   Please~update~an~up~to~date~version~of~the~bundles\\
42   `l3kernel'~and~`l3packages'\\
43   using~your~TeX~package~manager~or~from~CTAN.\\
44   \str_if_eq:nnT {#1} { expl3 } { Loading~xeCJK~will~abort! }
45 }
46 \ifpackagelater { expl3 } { 2013/07/20 } { }
47 { \msg_critical:nnn { xeCJK } { 13-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。

```
48 \RequirePackage { xtemplate }
49 \ifpackagelater { xtemplate } { 2012/11/10 } { }
50 { \msg_error:nnn { xeCJK } { 13-too-old } { xtemplate } }
51 \RequirePackage { xparse , l3keys2e }
52 \RequirePackage { everypage }
```


5.2 内部工具

分配临时变量。

```
53 \tl_new:N \l__xeCJK_tmp_tl
54 \int_new:N \l__xeCJK_tmp_int
55 \box_new:N \l__xeCJK_tmp_box
56 \dim_new:N \l__xeCJK_tmp_dim
57 \bool_new:N \l__xeCJK_tmp_bool
58 \skip_new:N \l__xeCJK_tmp_skip
59 \clist_new:N \l__xeCJK_tmp_clist
```

各种信息函数的缩略形式。

```
\__xeCJK_msg_new:nn
\__xeCJK_error:n 60 \cs_new_protected_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:nx 61 \cs_new_protected_nopar:Npn \__xeCJK_msg_new:nnn { \msg_new:nnnn { xeCJK } }
\__xeCJK_warning:nx 62 \cs_new_protected_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_info:nxx 63 \cs_new_protected_nopar:Npn \__xeCJK_error:nx { \msg_error:nxx { xeCJK } }
64 \cs_new_protected_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
65 \cs_new_protected_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nxx { xeCJK } }
66 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
67 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxxx { \msg_warning:nnxxx { xeCJK } }
68 \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: 69 \cs_new_protected_nopar:Npn \xeCJK_allow_break: { \tex_penalty:D \c_zero }
70 \cs_new_protected_nopar:Npn \xeCJK_no_break: { \tex_penalty:D \c_ten_thousand }
```

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

在 `\document` 前后加上各种钩子。

```
\__xeCJK_at_end_preamble:n
\__xeCJK_after_preamble:n 71 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
\__xeCJK_after_end_preamble:n 72 \tl_new:N \g__xeCJK_after_preamble_hook_tl
73 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl
74 \cs_new_protected:Npn \__xeCJK_at_end_preamble:n #1
75 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
76 \cs_new_protected:Npn \__xeCJK_after_preamble:n #1
77 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
78 \cs_new_protected:Npn \__xeCJK_after_end_preamble:n #1
79 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
80 \xeCJK_if_package_loaded:nTF { etoolbox }
81 {
82   \AtEndPreamble { \g__xeCJK_at_end_preamble_hook_tl }
83   \AfterPreamble { \g__xeCJK_after_preamble_hook_tl }
84   \AfterEndPreamble { \g__xeCJK_after_end_preamble_hook_tl }
85 }
86 {
87   \AtBeginDocument { \g__xeCJK_after_preamble_hook_tl }
88   \cs_new_protected_nopar:Npn \__xeCJK_document_left_hook:
89   { \group_end: \g__xeCJK_at_end_preamble_hook_tl \group_begin: }
90   \cs_new_protected_nopar:Npn \__xeCJK_document_right_hook:
91   { \scan_stop: \g__xeCJK_after_end_preamble_hook_tl \tex_ignorespaces:D }
92   \cs_gset_nopar:Npx \document
93   {
94     \__xeCJK_document_left_hook:
95     \exp_not:o { \document }
96     \__xeCJK_document_right_hook:
97   }
98 }
```

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

`\xeCJKShipoutHook`

在 `\shipout` 盒子里加钩子, 可以影响到页眉页脚。`\AtBeginDvi` 将参数保存在盒子中, 而 `atbegshi` 的 `\AtBeginShipout` 在 `\shipout` 盒子构建好之后才起作用, 所以它们都影响不到页眉页脚。

```
99 \AddEverypageHook { \xeCJKShipoutHook }
100 \NewDocumentCommand \xeCJKShipoutHook { }
101 {
102   \bool_if:NF \l__xeCJK_shipout_hook_bool
103   {
```



```

104         \bool_set_true:N \l__xeCJK_shipout_hook_bool
105         \tl_use:N \l__xeCJK_shipout_hook_tl
106     }
107 }
108 \cs_new_protected:Npn \__xeCJK_add_to_shipout:n #1
109 { \tl_put_right:Nn \l__xeCJK_shipout_hook_tl {#1} }
110 \tl_new:N \l__xeCJK_shipout_hook_tl
111 \bool_new:N \l__xeCJK_shipout_hook_bool
(End definition for \xeCJKShipoutHook. This function is documented on page 12.)

```

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

```

112 \cs_new_nopar:Npn \xeCJK_reverse:nnn #1#2#3
113 { \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 114 \cs_new_protected_nopar:Npn \xeCJK_tl_remove_outer_braces:N #1
115 { \tl_set:Nx #1 { \exp_args:NV \xeCJK_tl_remove_outer_braces:n #1 } }
116 \cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
117 {
118     \exp_last_unbraced:Nf
119     \__xeCJK_tl_remove_outer_braces:w { \tl_trim_spaces:n {#1} } \s__stop
120 }
121 \cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \s__stop
122 {
123     \bool_if:nTF { \tl_if_single_p:n {#1} && ! ( \tl_if_head_is_N_type_p:n {#1} ) }
124     { \xeCJK_tl_remove_outer_braces:n {#1} }
125     { \tl_trim_spaces:n {#1} }
126 }
(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 127 \cs_new_protected:Npn \xeCJK_cs_clear:N #1
128 { \cs_set_eq:NN #1 \prg_do_nothing: }
129 \cs_new_protected:Npn \xeCJK_cs_gclear:N #1
130 { \cs_gset_eq:NN #1 \prg_do_nothing: }
(End definition for \xeCJK_cs_clear:N and \xeCJK_cs_gclear:N.)

```

\xeCJK_swap_cs:NN 交换 #1 和 #2 的意义。

```

131 \cs_new_protected:Npn \xeCJK_swap_cs:NN #1#2
132 {
133     \cs_set_eq:NN \__xeCJK_swap_cs_aux:w #1
134     \cs_set_eq:NN #1 #2
135     \cs_set_eq:NN #2 \__xeCJK_swap_cs_aux:w
136     \cs_undefine:N \__xeCJK_swap_cs_aux:w
137 }
(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。

```

138 \cs_new_protected_nopar:Npn \xeCJK_font_gset_to_current:c #1
139 {
140     \exp_after:wN \cs_gset_eq:NN
141     \cs:w #1 \exp_after:wN \cs_end: \tex_the:D \tex_font:D
142 }
(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 143 \prg_new_conditional:Npnn \xeCJK_glyph_if_exist:N #1 { p , T , F , TF }
144 {
145     \etex_iffontchar:D \tex_font:D `#1 \exp_stop_f:
146     \prg_return_true: \else: \prg_return_false: \fi:
147 }
(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。

```

148 \tl_const:Nn \c_xeCJK_space_skip_tl
149 {
150   \int_compare:nNnTF \g__xeCJK_spacefactor_int = \c_one_thousand
151   {
152     \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
153     {
154       \tex_fontdimen:D \c_two \tex_font:D
155       plus \tex_fontdimen:D \c_three \tex_font:D
156       minus \tex_fontdimen:D \c_four \tex_font:D
157     }
158     { \tex_spaceskip:D }
159   }
160   {
161     \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
162     {
163       \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
164       { \__xeCJK_space_skip_scale:nnn { \tex_fontdimen:D \c_two \tex_font:D } }
165       {
166         \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
167         {
168           \__xeCJK_space_skip_scale:nnn
169           {
170             \tex_fontdimen:D \c_two \tex_font:D +
171             \tex_fontdimen:D \c_seven \tex_font:D
172           }
173         }
174         { \tex_xspaceskip:D \use_none:nn }
175       }
176       { \tex_fontdimen:D \c_three \tex_font:D }
177       { \tex_fontdimen:D \c_four \tex_font:D }
178     }
179     {
180       \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
181       { \__xeCJK_space_skip_scale:nnn { \tex_spaceskip:D } }
182       {
183         \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
184         {
185           \__xeCJK_space_skip_scale:nnn
186           { \tex_spaceskip:D + \tex_fontdimen:D \c_seven \tex_font:D }
187         }
188         { \tex_xspaceskip:D \use_none:nn }
189       }
190       { \etex_gluestretch:D \tex_spaceskip:D }
191       { \etex_glueshrink:D \tex_spaceskip:D }
192     }
193   }
194 }
195 \cs_new_nopar:Npn \__xeCJK_space_skip_scale:nnn #1#2#3
196 {
197   \dim_eval:n {#1}
198   plus \fp_eval:n { \g__xeCJK_spacefactor_int / 1000 } #2
199   minus
200   \int_div_truncate:nn
201   { 1000 * \tex_number:D #3 } { \g__xeCJK_spacefactor_int } sp
202 }
203 \int_new:N \g__xeCJK_spacefactor_int
204 \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。

```

205 \cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2
206 {
207   \hbox_set:Nn \l__xeCJK_tmp_box { #1 \scan_stop: \exp_after:wN } \exp_after:wN
208   \skip_set:Nn \exp_after:wN #2 \exp_after:wN { \skip_use:N \tex_lastskip:D }
209 }
(End definition for \xeCJK_glue_to_skip:nN.)

```

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

```
\xeCJK_if_blank_x:n:NTF
210 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
211 {
212   \if_case:w \pdfTeX_strcmp:D { } {#1} \exp_stop_f:
213   \prg_return_true:
214   \else:
215     \if_case:w \pdfTeX_strcmp:D { ~ } {#1} \exp_stop_f:
216     \prg_return_true: \else: \prg_return_false: \fi:
217   \fi:
218 }
```

(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:nNnn` 稍微快一点点。

```
\_xeCJK_int_until_do:wn
219 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
220 { \_xeCJK_int_until_do:wn \use_none:n { \reverse_if:N \if_int_compare:w #1#2 } }
221 \cs_new_protected:Npn \_xeCJK_int_until_do:wn \use_none:n #1
222 { #1 \exp_after:wN \_xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
223 \int_new:N \l_xeCJK_begin_int
224 \int_new:N \l_xeCJK_end_int
```

(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。

```
225 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
226 {
227   \cs_set_eq:NN \l_peek_search_token #1 \scan_stop:
228   \tl_set:Nn \_xeCJK_peek_catcode_true:w { \group_align_safe_end: #2 }
229   \tl_set:Nn \_xeCJK_peek_catcode_false:w { \group_align_safe_end: #3 }
230   \bool_set_false:N \l_xeCJK_peek_ignore_spaces_bool
231   \group_align_safe_begin:
232   \peek_after:Nw \_xeCJK_peek_catcode_ignore_spaces_branches:w
233 }
234 \cs_new_protected_nopar:Npn \_xeCJK_peek_catcode_ignore_spaces_branches:w
235 {
236   \if_meaning:w \l_peek_token \c_space_token
237     \bool_set_true:N \l_xeCJK_peek_ignore_spaces_bool
238     \exp_after:wN \peek_after:Nw
239     \exp_after:wN \_xeCJK_peek_catcode_ignore_spaces_branches:w
240     \tex_romannumeral:D 0
241   \else:
242     \if_catcode:w
243       \exp_not:N \l_peek_token \exp_not:N \l_peek_search_token
244       \exp_after:wN \exp_after:wN
245       \exp_after:wN \_xeCJK_peek_catcode_true:w
246     \else:
247       \exp_after:wN \exp_after:wN
248       \exp_after:wN \_xeCJK_peek_catcode_false:w
249     \fi:
250   \fi:
251 }
252 \tl_new:N \_xeCJK_peek_catcode_true:w
253 \tl_new:N \_xeCJK_peek_catcode_false:w
254 \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。

```
255 \cs_new_protected:Npn \xeCJK_peek_after_ignore_spaces:nw #1
256 {
257   \tl_set:Nn \_xeCJK_peek_after_do:w { \group_align_safe_end: #1 }
258   \group_align_safe_begin:
259   \peek_after:Nw \_xeCJK_peek_ignore_spaces_branches:w
260 }
261 \cs_new_protected_nopar:Npn \_xeCJK_peek_ignore_spaces_branches:w
262 {
263   \if_meaning:w \l_peek_token \c_space_token
264     \exp_after:wN \peek_after:Nw
265     \exp_after:wN \_xeCJK_peek_ignore_spaces_branches:w
```

```

266     \tex_romannumeral:D 0
267   \else:
268     \exp_after:wN \__xeCJK_peek_after_do:w
269   \fi:
270 }

```

(End definition for `\xeCJK_peek_after_ignore_spaces:nu`)

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

```

271 \cs_new_nopar:Npn \xeCJK_token_value_class:N #1
272 { \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, X_gT_EX 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`。目前, 0.9999.0 版以后的 X_gT_EX 的 `\meaning` 对于超出 BMP 的字符, 会返回两个字符, 分别对应于其 UTF-16 编码的首尾代理。⁵

```

273 \cs_new_nopar:Npn \xeCJK_token_value_charcode:N #1
274 { \exp_after:wN \__xeCJK_token_value_charcode:w \token_to_meaning:N #1 \q_stop }
275 \fp_compare:nNnTF { \int_use:N \xetex_XeTeXversion:D \XeTeXrevision } > { 0.9998 }
276 {
277   \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
278   {
279     \int_eval:n
280     {
281       \tl_if_empty:nTF {#4}
282       { `#3 }
283       { ( `#3 - "D800 ) * "400 + ( `#4 - "DC00 ) + "10000 }
284     }
285   }
286   \cs_new_eq:NN \xeCJK_xetex_mathcode:w \Umathcode
287 }
288 {
289   \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
290   { \int_eval:n { \tl_if_empty:nTF {#4} { `#3 } { "20000 } } }
291   \cs_new_eq:NN \xeCJK_xetex_mathcode:w \XeTeXmathcode
292 }

```

(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:NTF
293 \prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
294 {
295   \if_cs_exist:w \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 } \cs_end:
296   \prg_return_true: \else: \prg_return_false: \fi:
297 }
298 \cs_new_nopar:Npn \__xeCJK_CJK_class_tl:n #1
299 { c__xeCJK_CJK_class_ \int_eval:n {#1} _tl }
300 \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:NTTF
301 \prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
302 {
303   \if_int_compare:w
304   \xeCJK_token_value_class:N #1 = \xeCJK_token_value_class:N #2 \exp_stop_f:
305   \prg_return_true: \else: \prg_return_false: \fi:
306 }

```

(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>。

⁵参见 <http://tug.org/pipermail/xetex/2013-June/024543.html>。

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。

```
307 \keys_define:nn { xecjk / options }
308 {
309   xecjkActive .choice: ,
310   xecjkActive / true .code:n = { \makexecjkActive } ,
311   xecjkActive / false .code:n = { \makexecjkInactive } ,
312   xecjkActive .default:n = { true }
313 }
```

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

```
\makexecjkActive
\makexecjkInactive 314 \NewDocumentCommand \makexecjkActive { } { \XeTeXinterchartokenstate = \c_one }
315 \NewDocumentCommand \makexecjkInactive { } { \XeTeXinterchartokenstate = \c_zero }
```

(End definition for `\makexecjkActive` and `\makexecjkInactive`.)

抑制 BOM。

```
316 \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`

```
317 \seq_new:N \g__xecjk_class_seq
318 \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。

```
319 \cs_new_protected_nopar:Npn \xecjk_new_class:n #1
320 {
321   \int_if_exist:cTF { \__xecjk_class_csname:n {#1} }
322   { \__xecjk_error:nx { class-already-defined } {#1} }
323   {
324     \exp_args:Nc \newXeTeXintercharclass { \__xecjk_class_csname:n {#1} }
325     \clist_new:c { g__xecjk_#1_range_clist }
326     \seq_gput_right:Nn \g__xecjk_class_seq {#1}
327     \seq_gput_right:Nv \g__xecjk_new_class_seq { \__xecjk_class_csname:n {#1} }
328   }
329 }
```

(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。

```
330 \cs_new_protected_nopar:Npn \xecjk_save_class:nn #1#2
331 {
332   \int_if_exist:cTF { \__xecjk_class_csname:n {#1} }
333   { \__xecjk_error:nx { class-already-defined } {#1} }
334   {
335     \int_const:cn { \__xecjk_class_csname:n {#1} } {#2}
336     \clist_new:c { g__xecjk_#1_range_clist }
337     \seq_gput_right:Nn \g__xecjk_class_seq {#1}
338   }
339 }
```

(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。

```
340 \cs_new_nopar:Npn \__xecjk_class_csname:n #1 { c__xecjk_#1_class_int }
341 \cs_new_eq:cN { \__xecjk_class_csname:n { Others } } \l__xecjk_tmp_int
342 \__xecjk_msg_new:nn { class-already-defined }
343 {
344   XeTeX~character~class~`#1'~has~been~already~defined.\\\
345   Please~take~another~name. \\
346 }
```

(End definition for `_xeCJK_class_cname:n`.)

`xeCJK` 需要以下字符类别用于字符输出。其中 `Default`、`CJK`、`FullLeft`、`FullRight`、`Boundary` 为 `XYTeX` 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)⁶ 需要 `XYTeX` 0.9999.0 以上的版本⁷和相关字体的支持。

类别	说明	例子
<code>Default</code>	西文一般符号	abc123
<code>CJK</code>	<code>CJK</code> 表意符号	汉字あいう
<code>FullLeft</code>	全角左标点	(《: “
<code>FullRight</code>	全角右标点	, 。)》”
<code>HalfLeft</code>	半角左标点	([{
<code>HalfRight</code>	半角右标点	,.?)]}
<code>NormalSpace</code>	前后原始间距的符号	/
<code>Boundary</code>	边界	空格
<code>IVS</code>	异体字选择符	“回字有四样写法”

`Default` 这五类是 `XYTeX` 预定义的类别。

```

      CJK
FullLeft  \xeCJK_save_class:nn { Default } { \c_zero }
FullRight \xeCJK_save_class:nn { CJK } { \c_one }
Boundary  \xeCJK_save_class:nn { FullLeft } { \c_two }
          \xeCJK_save_class:nn { FullRight } { \c_three }
          \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
NormalSpace
IVS
          \xeCJK_new_class:n { HalfLeft }
          \xeCJK_new_class:n { HalfRight }
          \xeCJK_new_class:n { NormalSpace }
          \xeCJK_new_class:n { IVS }

```

(End definition for *HalfLeft* and others.)

```

\_xeCJK\_HalfLeft\_chars\_clist 西文半角左/右标点和前后原始间距的字符类。
\_xeCJK\_HalfRight\_chars\_clist
\_xeCJK\_NormalSpace\_chars\_clist
          \clist_const:Nn \c\_xeCJK\_HalfLeft\_chars\_clist
          { "28 , "2D , "5B , "60 , "7B }
          \clist_const:Nn \c\_xeCJK\_HalfRight\_chars\_clist
          { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
          \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 `XYTeX` 的脚本 `unicode-char-prep.pl` 和 Unicode 数据库⁸。

```

\_xeCJK\_OP\_chars\_clist
          \c\_xeCJK\_OP\_chars\_clist
          {
            "2018 , "201C ,
            "2329 , "3008 , "300A , "300C , "300E , "3010 , "3014 , "3016 , "3018 , "301A ,
            "301D , "FE17 , "FE35 , "FE37 , "FE39 , "FE3B , "FE3D , "FE3F , "FE41 , "FE43 ,
            "FE47 , "FE59 , "FE5B , "FE5D , "FF08 , "FF3B , "FF5B , "FF5F , "FF62
          }

```

以下代码的第一行是中西文共用的左引号。

```

          \clist_const:Nn \c\_xeCJK\_OP\_chars\_clist
          {
            "2018 , "201C ,
            "2329 , "3008 , "300A , "300C , "300E , "3010 , "3014 , "3016 , "3018 , "301A ,
            "301D , "FE17 , "FE35 , "FE37 , "FE39 , "FE3B , "FE3D , "FE3F , "FE41 , "FE43 ,
            "FE47 , "FE59 , "FE5B , "FE5D , "FF08 , "FF3B , "FF5B , "FF5F , "FF62
          }

```

(End definition for `\c_xeCJK_OP_chars_clist`.)

⁶<http://www.unicode.org/reports/tr37/>

⁷<http://tug.org/pipermail/xetex/2013-March/024118.html>

⁸<http://www.unicode.org/reports/tr14/>

\c__xeCJK_PR_chars_clist Prefix Numeric (PR)

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

```

368 \clist_const:Nn \c__xeCJK_PR_chars_clist
369 { "20A9 , "FE69 , "FF04 , "FFE1 , "FFE5 , "FFE6 }

```

(End definition for \c__xeCJK_PR_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。

```

370 \clist_const:Nx \c__xeCJK_FullLeft_chars_clist
371 {
372     \c__xeCJK_OP_chars_clist ,
373     \c__xeCJK_PR_chars_clist
374 }

```

(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。

```

375 \clist_const:Nn \c__xeCJK_CL_chars_clist
376 {
377     "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2027 , "2500 ,
378     "232A , "3001 , "3002 , "3009 , "300B , "300D , "300F , "3011 , "3015 , "3017 ,
379     "3019 , "301B , "301E , "301F , "FE11 , "FE12 , "FE18 , "FE36 , "FE38 , "FE3A ,
380     "FE3C , "FE3E , "FE40 , "FE42 , "FE44 , "FE48 , "FE50 , "FE52 , "FE5A , "FE5C ,
381     "FE5E , "FF09 , "FF0C , "FF0E , "FF3D , "FF5D , "FF60 , "FF61 , "FF63 , "FF64
382 }

```

(End definition for \c__xeCJK_CL_chars_clist.)

\c__xeCJK_NS_chars_clist Nonstarter (NS)

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

```

383 \clist_const:Nn \c__xeCJK_NS_chars_clist
384 {
385     "3005 , "301C , "303B , "303C , "309B , "309C , "309D , "309E , "30A0 , "30FB ,
386     "30FD , "30FE , "A015 , "FE54 , "FE55 , "FF1A , "FF1B , "FF65 , "FF9E , "FF9F
387 }

```

(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	?	
--	--------	---	--------	---	--------	---	--------	---	--------	---	--------	---	--

```

388 \clist_const:Nn \c__xeCJK_EX_chars_clist
389 { "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	;	
--	--------	---	--------	---	--------	---	--

```

390 \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	ヤ	U+FF6D	ユ	U+FF6E	ヨ
U+FF6F	ツ	U+FF70	ー										

```
391 \clist_const:Nn \c__xeCJK_CJ_chars_clist
392 {
393     "3041 , "3043 , "3045 , "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E ,
394     "3095 , "3096 , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
395     "30E7 , "30EE , "30F5 , "30F6 , "30FC , "31F0 , "31F1 , "31F2 , "31F3 , "31F4 ,
396     "31F5 , "31F6 , "31F7 , "31F8 , "31F9 , "31FA , "31FB , "31FC , "31FD , "31FE ,
397     "31FF , "FF67 , "FF68 , "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F ,
398     "FF70
399 }
```

(End definition for \c__xeCJK_CJ_chars_clist.)

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

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

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

(End definition for \c__xeCJK_PO_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。

```
401 \clist_const:Nx \c__xeCJK_FullRight_chars_clist
402 {
403     \c__xeCJK_CL_chars_clist ,
404     \c__xeCJK_NS_chars_clist ,
405     \c__xeCJK_EX_chars_clist ,
406     \c__xeCJK_IS_chars_clist ,
407     \c__xeCJK_CJ_chars_clist ,
408     \c__xeCJK_PO_chars_clist
409 }
```

(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。

```
410 \clist_const:Nn \c__xeCJK_CJK_chars_clist
411 {
```

- Hangul Jamo (谚文字母)

```
412     "1100 -> "11FF ,
```

- CJK Radicals Supplement (中日韩部首补充)

```
413     "2E80 -> "2EFF ,
```

- Kangxi Radicals (康熙字典部首)

```
414     "2F00 -> "2FDF ,
```

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

```
415     "2FF0 -> "2FFF ,
```

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

```
416     "3000 -> "303F ,
```

- Hiragana (日文平假名)

```
417     "3040 -> "309F ,
```

- Katakana (日文片假名)

```
418     "30A0 -> "30FF ,
```

- Bopomofo (注音字母)
419 "3100 -> "312F ,
- Hangul Compatibility Jamo (谚文兼容字母)
420 "3130 -> "318F ,
- Kanbun (象形字注释标志)
421 "3190 -> "319F ,
- Bopomofo Extended (注音字母扩展)
422 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)
423 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)
424 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
425 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
426 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
427 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
428 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
429 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
430 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
431 "A490 -> "A4CF ,
- Hangul Jamo Extended-A (谚文扩展 A)
432 "A960 -> "A97F ,
- Hangul Syllables (谚文音节)
433 "AC00 -> "D7AF ,
- Hangul Jamo Extended-B (谚文扩展 B)
434 "D7B0 -> "D7FF ,
- CJK Compatibility Ideographs (中日韩兼容表意文字)
435 "F900 -> "FAFF ,
- Vertical Forms (竖排形式)
436 "FE10 -> "FE1F ,
- CJK Compatibility Forms (中日韩兼容形式)
437 "FE30 -> "FE4F ,

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

```
438      "FF00 -> "FFE0 ,
```

- Kana Supplement (日文假名增补)

```
439      "1B000 -> "1B0FF ,
```

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

```
440      "1F200 -> "1F2FF ,
```

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

```
441      "20000 -> "2A6DF ,
```

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

```
442      "2A700 -> "2B73F ,
```

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

```
443      "2B740 -> "2B81F ,
```

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

```
444      "2F800 -> "2FA1F
```

```
445    }
```

(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。

```
446 \clist_const:Nn \c__xeCJK_IVS_chars_clist
```

```
447 {
```

- 日文假名浊点

```
448      "3099 -> "309A ,
```

- Variation Selectors (异体字选择符)

```
449      "FE00 -> "FE0F ,
```

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

```
450      "E0100 -> "E01EF
```

```
451    }
```

(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

```
452 \seq_new:N \g__xeCJK_base_class_seq
```

```
453 \seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq
```

```
454 \seq_new:N \g__xeCJK_non_CJK_class_seq
```

```
455 \seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq
```

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

```
457 \seq_new:N \g__xeCJK_CJK_class_seq
```

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

```
459 {
```

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

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

```
462 }
```

```
463 \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。

```
464 \cs_new_nopar:Npn \xeCJK_class_num:n #1 { \use:c { \__xeCJK_class_csname:n {#1} } }
```

(End definition for \xeCJK_class_num:n.)

`\xeCJKDeclareCharClass`

```
465 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
466 {
467   \xeCJK_declare_char_class:nx {#2} {#3}
468   \IfBooleanT {#1} { \xeCJKResetPunctClass }
469 }
```

(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
470 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
471 {
472   \clist_set:Nn \l__xeCJK_tmp_clist {#2}
473   \clist_gconcat:ccN
474   { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmp_clist
475   \clist_map_inline:Nn \l__xeCJK_tmp_clist
476   {
477     \str_if_eq:nnF {##1} { -> }
478     {
479       \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
480       { \xeCJK_class_num:n {#1} }
481     }
482   }
483   \xeCJK_set_char_class:nnn { "3099 } { "309A } { \xeCJK_class_num:n { IVS } }
484 }
485 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw
486 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
487 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
488 \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`

```
489 \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
490 {
491   \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
492   {
493     \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
494     \int_set_eq:NN #3 #4
495   }
496   {
497     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
498     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
499   }
500 }
```

(End definition for `__xeCJK_check_num_range:nnNN`.)

```
501 \int_set:Nn \l__xeCJK_begin_int { "AC00 }
502 \int_set:Nn \l__xeCJK_end_int { "D7A3 }
503 \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
504 {
505   \char_set_catcode_letter:n { \l__xeCJK_begin_int }
506   \int_incr:N \l__xeCJK_begin_int
507 }
```

`\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。

```
508 \cs_new_protected_nopar:Npn \xeCJK_set_char_class:nnn #1#2#3
509 {
510   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
511   \int_set:Nn \l__xeCJK_tmp_int {#3}
512   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
513   {
514     \XeTeXcharclass \l__xeCJK_begin_int = \l__xeCJK_tmp_int
515     \int_incr:N \l__xeCJK_begin_int
516   }
517 }
```

(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。

```
518 \cs_new_protected_nopar:Npn \_xeCJK_set_char_class_eq:nn #1#2
519 {
520   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#2} }
521   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
522   { \XeTeXcharclass ##1 = \l__xeCJK_tmp_int }
523 }
(End definition for \_xeCJK_set_char_class_eq:nn.)
```

`\normalspacedchars` 声明前后不加间距的字符。

```
524 \NewDocumentCommand \normalspacedchars { m }
525 {
526   \tl_map_inline:nn {#1}
527   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
528 }
(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。

```
529 \NewDocumentCommand \xeCJKResetPunctClass { }
530 {
531   \xeCJK_declare_char_class:nV { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
532   \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
533   \xeCJK_declare_char_class:nV { FullLeft } \c__xeCJK_FullLeft_chars_clist
534   \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
535 }
(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。

```
536 \NewDocumentCommand \xeCJKResetCharClass { }
537 {
538   \xeCJK_declare_char_class:nV { CJK } \c__xeCJK_CJK_chars_clist
539   \xeCJK_declare_char_class:nV { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
540   \xeCJK_declare_char_class:nV { IVS } \c__xeCJK_IVS_chars_clist
541   \xeCJKResetPunctClass
542 }
(End definition for \xeCJKResetCharClass. This function is documented on page 9.)
```

设置字符类别。

```
543 \xeCJKResetCharClass
```

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

```
544 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
545 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }
546 \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。

```
547 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2
548 { \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。

```
549 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2
550 { \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。

```
551 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
552 {
553   \xeCJK_inter_class_toks:nnx {#1} {#2}
554   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
555 }
556 \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。

```

557 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
558 {
559   \xeCJK_inter_class_toks:nnx {#1} {#2}
560   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
561 }
562 \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。

```

563 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
564 {
565   \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
566   \tl_if_empty:NF \l__xeCJK_tmp_tl
567   { \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmp_tl } }
568 }
(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。

```

569 \cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
570 {
571   \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
572   \tl_if_empty:NF \l__xeCJK_tmp_tl
573   {
574     \tl_replace_all:Nnn \l__xeCJK_tmp_tl {#3} {#4}
575     \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmp_tl }
576   }
577 }
(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。

```

578 \cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
579 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
580 \cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
581 { \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:

```

582 \cs_new_protected_nopar:Npn \xeCJK_class_group_begin:
583 {
584   \c_group_begin_token
585   \bool_set_true:N \l__xeCJK_CJK_group_bool
586 }
587 \bool_new:N \l__xeCJK_CJK_group_bool
588 \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。

```

589 \AtEndOfPackage
590 {
591   \seq_map_inline:Nn \g__xeCJK_class_seq
592   {
593     \str_if_eq:nnTF {#1} { IVS }
594     { \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } { CJK } }
595     {
596       \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } {#1}
597       \str_if_eq:nnF {#1} { CJK }
598       { \xeCJK_copy_inter_class_toks:nnnn {#1} { IVS } {#1} { CJK } }
599     }
600   }
601 }
602 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
603 {
604   \xeCJK_inter_class_toks:nnn {#1} { CJK }
605   {
606     \xeCJK_class_group_begin:
607     \xeCJK_select_font:
608     \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
609     \xeCJK_clear_Boundary_and_CJK_toks:
610     \CJKsymbol
611   }
612   \xeCJK_inter_class_toks:nnn { CJK } {#1} { \xeCJK_class_group_end: }
613 }
614 \clist_map_inline:nn { Default , HalfLeft }
615 {
616   \xeCJK_inter_class_toks:nnn { Boundary } {#1} { \xeCJK_Boundary_and_Default: }
617   \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue }
618 }

```

`\xeCJK_Boundary_and_Default:`

```

619 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_Default:
620 {
621   \bool_if:nTF
622   {
623     \l__xeCJK_xecglue_bool &&
624     \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
625     \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
626   }
627   {
628     \tex_unskip:D
629     \bool_if:nTF
630     {
631       \xeCJK_if_last_node_p:n { CJK } ||
632       \xeCJK_if_last_node_p:n { CJK-space }
633     }
634     { \xeCJK_remove_node: \CJKecglue } { ~ }
635   }
636   {
637     \bool_if:nTF
638     {
639       \xeCJK_if_last_node_p:n { CJK } ||
640       \xeCJK_if_last_node_p:n { CJK-nobreak }
641     }
642     { \xeCJK_remove_node: \CJKecglue }
643     {
644       \xeCJK_if_last_node:nT { CJK-space }
645       { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
646     }
647   }
648 }

```

(End definition for `\xeCJK_Boundary_and_Default:`)

```

649 \clist_map_inline:nn { Default , HalfRight }
650 {

```

```

651 \xeCJK_inter_class_toks:nnn {#1} { Boundary }
652 {
653   \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
654   \peek_meaning_remove:NTF \tex_italiccorrection:D
655   { \tex_italiccorrection:D { \xeCJK_make_node:n { default } } }
656   {
657     \token_if_space:NTF \l_peek_token
658     { { \xeCJK_make_node:n { default-space } } }
659     { { \xeCJK_make_node:n { default } } }
660   }
661 }
662 \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue }
663 }

664 \xeCJK_inter_class_toks:nnn { Boundary } { NormalSpace }
665 { \xeCJK_Boundary_and_NormalSp: }

```

\xeCJK_Boundary_and_NormalSp:

```

666 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_NormalSp:
667 {
668   \bool_if:nTF
669   {
670     \l__xeCJK_xecglue_bool &&
671     \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
672     \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
673   }
674   {
675     \tex_unskip:D
676     \bool_if:nTF
677     {
678       \xeCJK_if_last_node_p:n { CJK } ||
679       \xeCJK_if_last_node_p:n { CJK-space }
680     }
681     { \xeCJK_remove_node: \CJKecglue } { ~ }
682   }
683   {
684     \xeCJK_if_last_node:nT { CJK-space }
685     { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
686   }
687 }

```

(End definition for \xeCJK_Boundary_and_NormalSp:.)

```

688 \xeCJK_inter_class_toks:nnn { NormalSpace } { Boundary }
689 {
690   \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
691   \peek_meaning_remove:NTF \tex_italiccorrection:D
692   { \tex_italiccorrection:D { \xeCJK_make_node:n { normalspace } } }
693   {
694     \token_if_space:NTF \l_peek_token
695     { { \xeCJK_make_node:n { default-space } } }
696     { { \xeCJK_make_node:n { normalspace } } }
697   }
698 }

699 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
700 {
701   \xeCJK_check_for_glue:
702   \xeCJK_class_group_begin:
703   \xeCJK_clear_Boundary_and_CJK_toks:
704   \xeCJK_select_font:
705   \CJKsymbol
706 }

```

\xeCJK_check_for_glue:

```

707 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
708 {
709   \bool_if:nTF
710   { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
711   { \xeCJK_remove_node: \CJKglue }

```

```

712 {
713   \xeCJK_if_last_node:nTF { CJK-nobreak }
714   { \xeCJK_remove_node: \xeCJK_no_break: \CJKglue }
715   {
716     \bool_if:nTF
717     {
718       \xeCJK_if_last_node_p:n { default } ||
719       \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
720     }
721     { \xeCJK_remove_node: \CJKecglue }
722     {
723       \bool_if:nT
724       {
725         \l__xeCJK_xecglue_bool &&
726         \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
727         ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
728           \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_xecglue_skip )
729       }
730       {
731         \tex_unskip:D
732         \bool_if:nTF
733         {
734           \xeCJK_if_last_node_p:n { default-space } ||
735           \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
736           \xeCJK_if_last_node_p:n { default }
737         }
738         { \xeCJK_remove_node: \CJKecglue }
739         {
740           \bool_if:nTF
741           {
742             \xeCJK_if_last_node_p:n { CJK } ||
743             \xeCJK_if_last_node_p:n { CJK-space }
744           }
745           {
746             \xeCJK_remove_node:
747             \bool_if:NTF \l__xeCJK_reserve_space_bool
748             { ~ } { \CJKglue }
749           }
750           { ~ }
751         }
752       }
753     }
754   }
755 }
756 }

```

(End definition for `\xeCJK_check_for_glue:`)

`\xeCJK_if_last_node_p:n`

```

\xeCJK_if_last_node:nTF 757 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
758 {
759   \if_dim:w \use:c { c__xeCJK_#1_node_dim } = \tex_lastkern:D
760   \prg_return_true: \else: \prg_return_false: \fi:
761 }

```

(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 762 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
763 {
764   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
765   { \dim_gset:cn } { \dim_const:cn }
766   { c__xeCJK_#1_node_dim } {#2}
767 }
768 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
769 {
770   \tex_kern:D - \use:c { c__xeCJK_#1_node_dim }
771   \tex_kern:D \use:c { c__xeCJK_#1_node_dim }
772 }
773 \cs_new_protected_nopar:Npn \xeCJK_remove_node:

```

```

774 { \tex_unkern:D \tex_unkern:D }
775 \xeCJK_def_node:nn { CJK } { 11 sp }
776 \xeCJK_def_node:nn { CJK-space } { 12 sp }
777 \xeCJK_def_node:nn { default } { 13 sp }
778 \xeCJK_def_node:nn { default-space } { 14 sp }
779 \xeCJK_def_node:nn { CJK-nobreak } { 15 sp }
780 \xeCJK_def_node:nn { normalspace } { 16 sp }
(End definition for \xeCJK_def_node:nn and \xeCJK_make_node:n)

```

CJKglue CJK 文字之间插入的 glue。

```

781 \keys_define:nn { xeCJK / options }
782 {
783   CJKglue .code:n =
784   {
785     \cs_set_protected_nopar:Npn \CJKglue {#1}
786     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
787   }
788 }
789 \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

```

790 \keys_define:nn { xeCJK / options }
791 {
792   CJKecglue .code:n =
793   {
794     \cs_set_protected_nopar:Npn \CJKecglue {#1}
795     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
796   } ,
797   xCJKecglue .choice: ,
798   xCJKecglue / true .code:n =
799   {
800     \bool_set_true:N \l__xeCJK_xecglue_bool
801     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
802   } ,
803   xCJKecglue / false .code:n =
804   {
805     \bool_set_false:N \l__xeCJK_xecglue_bool
806     \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_tl
807   } ,
808   xCJKecglue / unknown .code:n =
809   {
810     \bool_set_true:N \l__xeCJK_xecglue_bool
811     \cs_set_protected_nopar:Npn \CJKecglue {#1}
812     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
813     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
814   } ,
815   xCJKecglue .default:n = { true }
816 }
817 \skip_new:N \l__xeCJK_ecglue_skip
818 \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。

```

819 \keys_define:nn { xeCJK / options }
820 {
821   CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
822   space .meta:n = { CJKspace = true } ,
823   nospace .meta:n = { CJKspace = false }
824 }
(End definition for CJKspace. This function is documented on page 3.)

825 \xeCJK_inter_class_toks:nnn { CJK } { Boundary } { \xeCJK_CJK_and_Boundary:w }

```

\xeCJK_CJK_and_Boundary:w 当边界是 \relax 的时候,它可能是由 \csname ... \endcsname 的形式产生的,这样就可能出现⁹。原来是都在未定义控制序列前都加上 \exp_not:N,现在是采用分组结束后手工恢复的方式。

```

826 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_Boundary:w
827 {
828   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
829   {
830     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
831     { \xeCJK_class_group_end: \xeCJK_space_or_xecglue: }
832     { \xeCJK_class_group_end: \CJKecglue }
833   }
834   {
835     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
836     {
837       \bool_if:nTF
838       {
839         \token_if_macro_p:N \l_peek_token ||
840         ( \l__xeCJK_reserve_space_bool && \token_if_letter_p:N \l_peek_token )
841       }
842       {
843         \xeCJK_class_group_end: { \xeCJK_make_node:n { CJK-space } }
844         \xeCJK_space_or_xecglue:
845       }
846       { \xeCJK_class_group_end: { \xeCJK_make_node:n { CJK-space } } }
847     }
848     {
849       \token_if_eq_meaning:NNTF \l_peek_token \scan_stop:
850       { \__xeCJK_CJK_and_Boundary_relax:N }
851       { \__xeCJK_CJK_and_Boundary_aux: }
852     }
853   }
854 }
855 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_Boundary_aux:
856 { \xeCJK_class_group_end: { \xeCJK_make_node:n { CJK } } }
857 \cs_new_protected:Npn \__xeCJK_CJK_and_Boundary_relax:N #1
858 {
859   \__xeCJK_CJK_and_Boundary_aux:
860   \token_if_eq_meaning:NNTF #1 \scan_stop:
861   {#1} { \cs_set_eq:NN #1 \scan_stop: #1 }
862 }

```

(End definition for \xeCJK_CJK_and_Boundary:w)

\xeCJK_ignore_spaces:w

```

863 \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
864 {
865   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
866   {
867     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
868     { \xeCJK_space_or_xecglue: } { \CJKecglue }
869   }
870   {
871     \bool_if:NT \l__xeCJK_peek_ignore_spaces_bool
872     {
873       \xeCJK_if_last_node:nT { CJK }
874       { \xeCJK_remove_node: { \xeCJK_make_node:n { CJK-space } } }
875       \bool_if:nT
876       {
877         \token_if_macro_p:N \l_peek_token ||
878         ( \l__xeCJK_reserve_space_bool && \token_if_letter_p:N \l_peek_token )
879       }
880       { \xeCJK_space_or_xecglue: }
881     }
882   }
883 }

```

(End definition for \xeCJK_ignore_spaces:w)

```

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

```

⁹参见 <http://bbs.ctex.org/forum.php?mod=viewthread&tid=71563>。

\xeCJK_CJK_and_CJK:N

```
885 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }
(End definition for \xeCJK_CJK_and_CJK:N)

886 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
887 { \xeCJK_FullLeft_and_CJK: \CJKsymbol }
888 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
889 { \xeCJK_FullRight_and_CJK: \CJKsymbol }
890 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
891 {
892   \clist_map_inline:nn { FullLeft , FullRight }
893   {
894     \xeCJK_inter_class_toks:nnx {#1} {##1}
895     { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
896     \xeCJK_inter_class_toks:nnc {##1} {#1} { xeCJK_##1_and_Default: }
897   }
898 }

899 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
900 { \xeCJK_Boundary_and_FullLeft:N }
901 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
902 { \xeCJK_Boundary_and_FullRight:N }
```

\xeCJK_FullRight_and_Boundary:

```
903 \xeCJK_app_inter_class_toks:nnn { FullLeft } { Boundary } { \tex_ignorespaces:D }
904 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
905 { \xeCJK_FullRight_and_Boundary: }
(End definition for \xeCJK_FullRight_and_Boundary:.)
```

\xeCJK_FullRight_and_Boundary:

```
906 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Boundary:
907 { \xeCJK_FullRight_and_Default: \tex_ignorespaces:D }
(End definition for \xeCJK_FullRight_and_Boundary:.)

908 \clist_map_inline:nn { CJK , FullLeft , FullRight }
909 {
910   \clist_map_inline:nn { FullLeft , FullRight }
911   { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
912 }
```

__xeCJK_punct_bound_rule:NN 用于抹去标点符号的全部左/右空白。

```
913 \cs_new_protected_nopar:Npn __xeCJK_punct_bound_rule:NN #1#2
914 {
915   \tex_vrule:D
916   width - \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
917   depth \c_zero_dim
918   height \c_zero_dim \scan_stop:
919 }
(End definition for \__xeCJK_punct_bound_rule:NN.)
```

__xeCJK_punct_rule:NN 用于减少标点符号的左/右空白。

```
920 \cs_new_protected_nopar:Npn \__xeCJK_punct_rule:NN #1#2
921 {
922   \tex_vrule:D
923   width \__xeCJK_use_punct_dim:nnn { rule } {#1} {#2}
924   depth \c_zero_dim
925   height \c_zero_dim \scan_stop:
926 }
(End definition for \__xeCJK_punct_rule:NN.)
```

__xeCJK_punct_offset:NN 经过以上 \vrule 处理后, 标点输出边界与实际边界的距离。

```
927 \cs_new_protected_nopar:Npn \__xeCJK_punct_offset:NN #1#2
928 { \__xeCJK_punct_kern:n { - \__xeCJK_use_punct_dim:nnn { offset } {#1} {#2} } }
929 \cs_new_protected_nopar:Npn \__xeCJK_punct_kern:n #1
930 { \tex_kern:D #1 \exp_stop_f: }
(End definition for \__xeCJK_punct_offset: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。

```
931 \cs_new_protected_nopar:Npn \_xeCJK_punct_glue:NN #1#2
932 {
933   \_xeCJK_punct_hskip:n
934   {
935     \_xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
936     plus \_xeCJK_use_punct_dim:nnn { plus } {#1} {#2}
937     minus \_xeCJK_use_punct_dim:nnn { minus } {#1} {#2}
938   }
939 }
940 \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。

```
941 \cs_new_protected_nopar:Npn \_xeCJK_punct_kern:NN #1#2
942 { \tex_kern:D \_xeCJK_use_punct_dim:nnn { kern } {#1} {#2} \exp_stop_f: }
(End definition for \_xeCJK_punct_kern:NN.)
```

`\g_xeCJK_last_punct_tl` 用于记录当前的标点符号。

```
943 \tl_new:N \g_xeCJK_last_punct_tl
(End definition for \g_xeCJK_last_punct_tl.)
```

`\xeCJK_FullLeft_and_CJK:`

```
944 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_CJK:
945 {
946   \_xeCJK_punct_if_middle:NTF \g_xeCJK_last_punct_tl
947   {
948     \_xeCJK_punct_bound_rule:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
949     \xeCJK_no_break:
950     \_xeCJK_punct_glue:NN \c_xeCJK_left_tl \g_xeCJK_last_punct_tl
951   }
952   { \xeCJK_no_break: }
953 }
(End definition for \xeCJK_FullLeft_and_CJK:.)
```

`\xeCJK_FullLeft_and_Default:`

```
954 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_Default:
955 {
956   \_xeCJK_punct_if_middle:NTF \g_xeCJK_last_punct_tl
957   {
958     \_xeCJK_punct_bound_rule:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
959     \xeCJK_class_group_end: \xeCJK_no_break:
960     \_xeCJK_punct_glue:NN \c_xeCJK_left_tl \g_xeCJK_last_punct_tl
961   }
962   { \xeCJK_class_group_end: \xeCJK_no_break: \_xeCJK_zero_glue: }
963 }
964 \cs_new_protected_nopar:Npn \_xeCJK_zero_glue:
965 { \skip_horizontal:N \c_zero_skip }
(End definition for \xeCJK_FullLeft_and_Default:.)
```

`\xeCJK_FullRight_and_CJK:`

```
966 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_CJK:
967 {
968   \_xeCJK_punct_rule:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
969   \_xeCJK_punct_offset:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
970   \_xeCJK_punct_glue:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
971   \CJKglue
972 }
(End definition for \xeCJK_FullRight_and_CJK:.)
```

`\xeCJK_FullRight_and_Default:`

```
973 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Default:
974 {
975   \_xeCJK_punct_rule:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
976   \xeCJK_class_group_end:
977   \_xeCJK_punct_offset:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
978   \_xeCJK_punct_glue:NN \c_xeCJK_right_tl \g_xeCJK_last_punct_tl
979 }
```

(End definition for \xeCJK_FullRight_and_Default:.)

\xeCJK_Default_and_FullLeft:nN

```

980 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullLeft:nN #1#2
981 {
982   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#2}
983   \__xeCJK_Default_and_FullLeft_glue:N {#2}
984   \xeCJK_class_group_begin:
985   \xeCJK_select_font:
986   \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
987   \xeCJK_clear_Boundary_and_CJK_toks:
988   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
989   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2}
990   \CJKpunctsymbol {#2}
991 }
992 \cs_new_protected_nopar:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
993 {
994   \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
995   \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
996 }

```

(End definition for \xeCJK_Default_and_FullLeft:nN.)

\xeCJK_CJK_and_FullLeft:N

```

997 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
998 {
999   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
1000   \__xeCJK_CJK_and_FullLeft_glue:N {#1}
1001   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1002   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
1003   \CJKpunctsymbol {#1}
1004 }
1005 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
1006 {
1007   \CJKglue
1008   \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
1009   \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
1010 }

```

(End definition for \xeCJK_CJK_and_FullLeft:N.)

\xeCJK_Boundary_and_FullLeft:N

```

1011 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullLeft:N #1
1012 {
1013   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
1014   \__xeCJK_Boundary_and_FullLeft_glue:N {#1}
1015   \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
1016   \xeCJK_class_group_begin:
1017   \xeCJK_select_font:
1018   \xeCJK_clear_Boundary_and_CJK_toks:
1019   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1020   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
1021   \CJKpunctsymbol {#1}
1022 }

```

(End definition for \xeCJK_Boundary_and_FullLeft:N.)

__xeCJK_Boundary_and_FullLeft_glue:N

\etex_lastnodetype:D 为 1 表示 hlist node, 在这里用来判断是否位于段首。基于正常情况下, \TeX 会在段落开头插入宽度为 \parindent 的水平盒子用于缩进。-1 表示 empty list, 常出现在盒子的起始位置, 在段落前使用 \noindent 就是这种情况。11 表示 glue node, 这里判断的目的是当全角左标点出现在 \LaTeX 表格的非 p 列行首时, 能够对齐到单元格的边界。判断基于标准 \LaTeX 表格的列格式 (\@tabclassz) 定义中, 在 l 列和 r 列前为了防止 \tabcolsep 被无意 \unskip 掉, 都加了 \hskip1sp, 而 c 列前则有 \hfil。13 表示 penalty node, 这里判断的目的是全角左标点出现在 \LaTeX 列表环境的 \item 后面时, 能对齐到边界。判断基于 \item 的内部定义 \@item 对 \everypar 进行了修改, 在这里起到影响作用的是 \box\@labels \penalty\z@。enumitem 宏包修改了 description 环境中使用的 \item (\enit@postlabel@i), 在这里起到影响作用的是 \penalty\z@ \hskip\labelsep。以上判断都比较粗略, 暂时也没有想起更好的办法。

```

1023 \cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1

```

```

1024 {
1025   \int_case:nnTF { \etex_lastnodetype:D }
1026   {
1027     { \c_one }
1028     {
1029       \box_set_to_last:N \l__xeCJK_tmp_box
1030       \bool_if:nTF
1031       {
1032         \int_compare_p:nNn \etex_lastnodetype:D = \c_minus_one &&
1033         \dim_compare_p:nNn { \box_wd:N \l__xeCJK_tmp_box } = \tex_parindent:D
1034       }
1035       { \box_use_clear:N \l__xeCJK_tmp_box \use_none:n }
1036       { \box_use_clear:N \l__xeCJK_tmp_box \use:n }
1037     }
1038     { \c_minus_one } { \__xeCJK_zero_glue: \use_none:n }
1039     { \c_eleven }
1040     {
1041       \bool_if:nTF
1042       {
1043         ! ( \skip_if_finite_p:n { \tex_lastskip:D } ) ||
1044         \skip_if_eq_p:nn { \tex_lastskip:D } { 1 sp }
1045       }
1046       { \__xeCJK_zero_glue: \use_none:n }
1047       {
1048         \skip_if_eq:nnTF { \tex_lastskip:D } { \labelsep }
1049         {
1050           \tex_unskip:D
1051           \bool_if:nTF
1052           {
1053             \int_compare_p:nNn \etex_lastnodetype:D = \c_thirteen &&
1054             \int_compare_p:nNn \tex_lastpenalty:D = \c_zero
1055           }
1056           { \skip_horizontal:n { \labelsep } \use_none:n }
1057           { \skip_horizontal:n { \labelsep } \use:n }
1058         }
1059         { \use:n }
1060       }
1061     }
1062     { \c_thirteen }
1063     {
1064       \int_compare:nNnTF \tex_lastpenalty:D = \c_zero
1065       {
1066         \tex_unpenalty:D
1067         \int_compare:nNnTF \etex_lastnodetype:D = \c_one
1068         { \tex_penalty:D \c_zero \use_none:n }
1069         { \tex_penalty:D \c_zero \use:n }
1070       }
1071       { \use:n }
1072     }
1073   }
1074   { { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} } }
1075   { { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} } }
1076 }

```

(End definition for `__xeCJK_Boundary_and_FullLeft_glue:N`.)

`\xeCJK_Default_and_FullRight:nN`

```

1077 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullRight:nN #1#2
1078 {
1079   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#2}
1080   \__xeCJK_Default_and_FullRight_glue:N {#2}
1081   \xeCJK_class_group_begin:
1082   \xeCJK_select_font:
1083   \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
1084   \xeCJK_clear_Boundary_and_CJK_toks:
1085   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
1086   \xeCJK_FullRight_symbol:N {#2}
1087 }

```

(End definition for `\xeCJK_Default_and_FullRight:nN`.)

\xeCJK_Boundary_and_FullRight:N

```
1088 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullRight:N #1
1089 {
1090   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
1091   \__xeCJK_Default_and_FullRight_glue:N {#1}
1092   \xeCJK_class_group_begin:
1093   \xeCJK_select_font:
1094   \xeCJK_clear_Boundary_and_CJK_toks:
1095   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1096   \xeCJK_FullRight_symbol:N {#1}
1097 }
```

(End definition for \xeCJK_Boundary_and_FullRight:N.)

\xeCJK_CJK_and_FullRight:N

```
1098 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
1099 {
1100   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
1101   \__xeCJK_CJK_and_FullRight_glue:N {#1}
1102   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1103   \xeCJK_FullRight_symbol:N {#1}
1104 }
```

(End definition for \xeCJK_CJK_and_FullRight:N.)

_xeCJK_CJK_and_FullRight_glue:N

_xeCJK_Default_and_FullRight_glue:N

```
1105 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
1106 {
1107   \__xeCJK_punct_if_long:NTF {#1}
1108   { \CJKglue }
1109   {
1110     \__xeCJK_punct_if_middle:NTF {#1}
1111     {
1112       \xeCJK_no_break:
1113       \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
1114       \__xeCJK_punct_bound_rule:NN \c__xeCJK_left_tl {#1}
1115     }
1116     { \xeCJK_no_break: }
1117   }
1118 }
1119 \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

```
1120 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
1121 {
1122   \xeCJK_no_break:
1123   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
1124   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
1125   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
1126   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1127   \CJKpunctsymbol {#1}
1128 }
```

(End definition for \xeCJK_FullLeft_and_FullLeft:N.)

\xeCJK_FullLeft_and_FullRight:N

```
1129 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
1130 {
1131   \xeCJK_no_break:
1132   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
1133   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
1134   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
1135   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1136   \xeCJK_no_break:
1137   \xeCJK_FullRight_symbol:N {#1}
1138 }
```

(End definition for \xeCJK_FullLeft_and_FullRight:N.)

```

\xeCJK_FullRight_and_FullLeft:N
1139 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
1140 {
1141   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
1142   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
1143   \xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
1144   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1145   \CJKpunctsymbol {#1}
1146 }
(End definition for \xeCJK_FullRight_and_FullLeft:N.)

```

```

\_xeCJK_punct_nobreak_kern:NN
1147 \cs_new_protected_nopar:Npn \_xeCJK_punct_nobreak_kern:NN #1#2
1148 {
1149   \_xeCJK_punct_kern:NN #1#2
1150   \xeCJK_no_break:
1151 }
1152 \cs_new_eq:NN \xeCJK_punct_kern:NN \_xeCJK_punct_nobreak_kern:NN
(End definition for \_xeCJK_punct_nobreak_kern:NN.)

```

```

\_xeCJK_punct_breakable_kern:NN
1153 \cs_new_protected_nopar:Npn \_xeCJK_punct_breakable_kern:NN #1#2
1154 {
1155   \_xeCJK_punct_rule:NN \c__xeCJK_right_tl #1
1156   \_xeCJK_punct_breakable_kern:n
1157   { \_xeCJK_use_punct_dim:nnn { break_kern } {#1} {#2} }
1158   \_xeCJK_punct_rule:NN \c__xeCJK_left_tl #2
1159 }
1160 \cs_new_eq:NN \_xeCJK_punct_breakable_kern:n \skip_horizontal:n
(End definition for \_xeCJK_punct_breakable_kern:NN.)

```

```

\xeCJK_FullRight_and_FullRight:N
1161 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
1162 {
1163   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
1164   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
1165   \_xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
1166   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
1167   \xeCJK_no_break:
1168   \xeCJK_FullRight_symbol:N {#1}
1169 }
(End definition for \xeCJK_FullRight_and_FullRight:N.)

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```

1170 \keys_define:nn { xecjk / options }
1171 {
1172   CheckFullRight .choice: ,
1173   CheckFullRight / true .code:n =
1174   {
1175     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
1176     {
1177       \cs_set_eq:NN \_xeCJK_save_FullRight_check: \xeCJK_FullRight_and_Boundary:
1178       \cs_set_eq:NN \_xeCJK_save_FullRight_symbol:N \xeCJK_FullRight_symbol:N
1179       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
1180       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \xeCJK_check_FullRight_symbol:Nw
1181     }
1182   } ,
1183   CheckFullRight / false .code:n =
1184   {
1185     \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
1186     {
1187       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \_xeCJK_save_FullRight_check:
1188       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \_xeCJK_save_FullRight_symbol:N
1189     }
1190   }

```

```

1190     } ,
1191     CheckFullRight      .default:n = { true }
1192   }
(End definition for CheckFullRight. This function is documented on page 5.)

```

\xeCJK_FullRight_symbol:N

```

1193 \cs_new_nopar:Npn \xeCJK_FullRight_symbol:N { \CJKpunctsymbol }
(End definition for \xeCJK_FullRight_symbol:N.)

```

\xeCJK_check_FullRight:

```

1194 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight:
1195 {
1196   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1197   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1198   \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1199   \group_align_safe_begin:
1200   \tl_case:NoTF \l_peek_token
1201     { \l__xeCJK_no_break_cs_case_tl }
1202     { \group_align_safe_end: \xeCJK_no_break: }
1203     { \group_align_safe_end: }
1204   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1205   \xeCJK_class_group_end:
1206 }
1207 \cs_generate_variant:Nn \tl_case:NnTF { No }
(End definition for \xeCJK_check_FullRight:.)

```

\xeCJK_check_FullRight_symbol:Nw

```

1208 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight_symbol:Nw #1
1209 { \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

```

1210 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
1211 {
1212   \tl_new:N #2
1213   \seq_new:N #3
1214   \keys_define:nn { xeCJK / options }
1215   {
1216     #1 .code:n =
1217     {
1218       \seq_set_split:Nnn #3 { } {##1}
1219       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1220     } ,
1221     #1+ .code:n =
1222     {
1223       \tl_map_inline:nn {##1}
1224       { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
1225       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1226     } ,
1227     #1- .code:n =
1228     {
1229       \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {####1} }
1230       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
1231     }
1232   }
1233 }
1234 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
1235 {
1236   \tl_clear:N #1
1237   \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
1238   #4
1239 }
(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。

```

1240 \xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
1241 \l__xeCJK_no_break_cs_case_tl \l__xeCJK_no_break_cs_seq { } { }

```


(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。

```
1242 \NewDocumentCommand \xeCJKnobreak { }
1243 {
1244   \bool_set_true:N \l__xeCJK_tmp_bool
1245   \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
1246   {
1247     \bool_if:NTF \l__xeCJK_tmp_bool
1248     {
1249       \bool_set_false:N \l__xeCJK_tmp_bool
1250       \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
1251     }
1252     { \skip_add:Nn \l__xeCJK_tmp_skip \tex_lastskip:D }
1253     \tex_unskip:D
1254   }
1255   \xeCJK_no_break:
1256   \bool_if:NF \l__xeCJK_tmp_bool { \skip_horizontal:N \l__xeCJK_tmp_skip }
1257 }
```

(End definition for *\xeCJKnobreak*. This function is documented on page 12.)

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```
1258 \keys_define:nn { xeCJK / options }
1259 {
1260   CheckSingle .choice: ,
1261   CheckSingle / true .code:n =
1262   {
1263     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1264     {
1265       \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
1266       \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1267     }
1268   } ,
1269   CheckSingle / false .code:n =
1270   {
1271     \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
1272     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
1273   } ,
1274   CheckSingle .default:n = { true } ,
1275   CJKchecksingle .meta:n = { CheckSingle = true }
1276 }
```

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

\xeCJK_check_single:Nw

```
1277 \cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
1278 {
1279   \peek_catcode:NTF \c_catcode_letter_token
1280   { \xeCJK_check_single:NNw #1 }
1281   {
1282     \group_align_safe_begin:
1283     \token_if_other:NTF \l_peek_token
1284     { \group_align_safe_end: \xeCJK_check_single:NNw #1 }
1285     {
1286       \group_align_safe_end:
1287       \bool_if:nTF
1288       {
1289         \str_if_eq_x_p:nn { \token_get_arg_spec:N \l_peek_token } { } &&
1290         \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
1291         ( \exp_after:wN \token_if_other_p:N \l_peek_token ||
1292           \exp_after:wN \token_if_letter_p:N \l_peek_token )
1293       }
1294       { \exp_after:wN \xeCJK_check_single:NNw \exp_after:wN #1 }
1295       { \__xeCJK_check_single_save:N #1 }
1296     }
1297   }
1298 }
```

(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。

```

1299 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
1300 {
1301   \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
1302   {
1303     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1304     {
1305       \bool_if:NTF \l__xeCJK_reserve_space_bool
1306       { \__xeCJK_check_single_save:N #1 #2 ~ }
1307       { \__xeCJK_check_single_space:NN #1#2 }
1308     }
1309     { \__xeCJK_check_single_save:N #1 #2 }
1310   }
1311   {
1312     \group_align_safe_begin:
1313     \token_if_other:NTF \l_peek_token
1314     {
1315       \group_align_safe_end:
1316       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1317       { \__xeCJK_check_single_space:NN #1#2 }
1318       { \__xeCJK_check_single_save:N #1 #2 }
1319     }
1320     {
1321       \token_if_cs:NTF \l_peek_token
1322       {
1323         \group_align_safe_end:
1324         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1325         { \xeCJK_check_single_cs:NNn #1#2 { ~ } }
1326         { \xeCJK_check_single_cs:NNn #1#2 { } }
1327       }
1328       {
1329         \group_align_safe_end:
1330         \bool_if:NTF
1331         {
1332           \l__xeCJK_plain_equation_bool &&
1333           \token_if_math_toggle_p:N \l_peek_token
1334         }
1335         {
1336           \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1337           { \xeCJK_check_single_equation:NNnNw #1 #2 { ~ } }
1338           { \xeCJK_check_single_equation:NNnNw #1 #2 { } }
1339         }
1340         {
1341           \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1342           { \__xeCJK_check_single_save:N #1 #2 ~ }
1343           { \__xeCJK_check_single_save:N #1 #2 }
1344         }
1345       }
1346     }
1347   }
1348 }

```

(End definition for `\xeCJK_check_single:NNw`)

`__xeCJK_check_single_space:NN`

```

1349 \cs_new_protected_nopar:Npn \__xeCJK_check_single_space:NN #1#2
1350 {
1351   \xeCJK_if_CJK_class:NTF #2
1352   {
1353     \xeCJK_if_CJK_class:NTF \l_peek_token
1354     { \__xeCJK_check_single_save:N #1 #2 }
1355     { \__xeCJK_check_single_save:N #1 #2 ~ }
1356   }
1357   { \__xeCJK_check_single_save:N #1 #2 ~ }
1358 }

```

(End definition for `__xeCJK_check_single_space:NN`)

\xeCJK_check_single_equation:NNnNw

```
1359 \cs_new_protected_nopar:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
1360 {
1361   \peek_catcode:NTF \c_math_toggle_token
1362   {
1363     \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1364     \xeCJK_make_node:n { CJK-nobreak } #2 #4
1365   }
1366   { \__xeCJK_check_single_save:N #1 #2#3#4 }
1367 }
```

(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:NnF` 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。

```
1368 \cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
1369 {
1370   \group_align_safe_begin:
1371   \tl_case:NnF \l_peek_token
1372   { \l__xeCJK_check_single_cs_case_tl }
1373   { \group_align_safe_end: \use_iii:nnn }
1374   { \xeCJK_check_single_env:nnNn }
1375   {
1376     \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1377     \xeCJK_make_node:n { CJK-nobreak } #2
1378   }
1379   { \__xeCJK_check_single_save:N #1 #2#3 }
1380 }
1381 \tl_new:N \l__xeCJK_check_single_cs_case_tl
1382 \cs_generate_variant:Nn \tl_case:NnF { No }
```

(End definition for \xeCJK_check_single_cs:NNn.)

\xeCJK_check_single_env:nnNn

```
1383 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
1384 {
1385   \group_align_safe_begin:
1386   \str_case:x:noTF {#4}
1387   { \l__xeCJK_inline_env_case_tl }
1388   { \group_align_safe_end: #2 }
1389   { \group_align_safe_end: #1 }
1390   #3 {#4}
1391 }
1392 \cs_generate_variant:Nn \str_case:x:nnTF { no }
```

(End definition for \xeCJK_check_single_env:nnNn.)

NewLineCS

```
1393 \xeCJK_cs_case_keys_define:nnNnn { NewLineCS }
1394 \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
1395 { \group_align_safe_end: \use_ii:nnn }
1396 {
1397   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1398   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1399 }
```

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

EnvCS

```
1400 \xeCJK_cs_case_keys_define:nnNnn { EnvCS }
1401 \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
1402 { \group_align_safe_end: \use:n }
```

```

1403 {
1404   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1405   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1406 }

```

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

InlineEnv

```

1407 \keys_define:nn { xeCJK / options }
1408 {
1409   InlineEnv .code:n =
1410   {
1411     \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
1412     \__xeCJK_update_inline_env_case_tl:
1413   } ,
1414   InlineEnv+ .code:n =
1415   {
1416     \clist_map_inline:nn {#1}
1417     {
1418       \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
1419       { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
1420     }
1421     \__xeCJK_update_inline_env_case_tl:
1422   } ,
1423   InlineEnv- .code:n =
1424   {
1425     \clist_map_inline:nn {#1}
1426     { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
1427     \__xeCJK_update_inline_env_case_tl:
1428   }
1429 }
1430 \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:

```

1431 \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
1432 {
1433   \tl_clear:N \l__xeCJK_inline_env_case_tl
1434   \seq_map_inline:Nn \l__xeCJK_inline_env_seq
1435   { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { } } }
1436 }
1437 \tl_new:N \l__xeCJK_inline_env_case_tl

```

(End definition for __xeCJK_update_inline_env_case_tl:.)

PlainEquation

```

1438 \keys_define:nn { xeCJK / options }
1439 { 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

```

1440 \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。

```

1441 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
1442 { s > { \TrimSpaces } m > { \TrimSpaces } m }
1443 {
1444   \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
1445   \IfBooleanT {#1} { \xeCJKResetPunctClass }
1446 }
1447 \@onlypreamble \xeCJKDeclareSubCJKBlock

```

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

`\xeCJKCancelSubCJKBlock`
`\xeCJKRestoreSubCJKBlock`

取消和恢复对 CJK 子区的声明。

```

1448 \bool_new:N \l__xeCJK_sub_cancel_bool
1449 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
1450 {
1451   \bool_if:NF \l__xeCJK_sub_cancel_bool
1452   {
1453     \bool_set_true:N \l__xeCJK_sub_cancel_bool
1454     \__xeCJK_sub_restore_or_cancel:x {#2}
1455     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1456   }
1457 }
1458 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
1459 {
1460   \bool_if:NT \l__xeCJK_sub_cancel_bool
1461   {
1462     \bool_set_false:N \l__xeCJK_sub_cancel_bool
1463     \__xeCJK_sub_restore_or_cancel:x {#2}
1464     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1465   }
1466 }

```

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

`__xeCJK_sub_restore_or_cancel:n`

```

1467 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
1468 {
1469   \clist_map_inline:nn {#1}
1470   {
1471     \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
1472     {
1473       \xeCJK_declare_char_class:nx
1474       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
1475       { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
1476     }
1477     { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
1478   }
1479 }
1480 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
1481 \__xeCJK_msg_new:nn { SubBlock-undefined }
1482 {
1483   The~CJK~sub~block~`#1'~is~undefined.\\\
1484   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
1485   to~declare~it.
1486 }

```

(End definition for `__xeCJK_sub_restore_or_cancel:n`.)

`\xeCJK_declare_sub_char_class:nnn`

```

1487 \cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
1488 {
1489   \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
1490   {
1491     \xeCJK_new_class:n { #1/#2 }
1492     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
1493     \xeCJK_new_sub_key:n {#2}
1494   }
1495   \xeCJK_declare_char_class:nn { #1/#2 } {#3}
1496 }
1497 \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`

```

1498 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
1499 {
1500   \seq_map_inline:Nn \g__xeCJK_base_class_seq
1501   {
1502     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
1503     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
1504     \str_if_eq:nnTF {##1} { CJK }

```

```

1505     {
1506       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
1507       { \_xeCJK_switch_font:nn {#1} {#2} }
1508     }
1509     {
1510       \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
1511       { \CJKsymbol }
1512       { \_xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
1513     }
1514   }
1515   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
1516   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
1517   {
1518     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
1519     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
1520     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
1521     { \_xeCJK_switch_font:nn {#2} {##1} }
1522     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
1523     { \_xeCJK_switch_font:nn {##1} {#2} }
1524   }
1525   \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
1526   \_xeCJK_save_CJK_class:n { #1/#2 }
1527   \clist_map_inline:nn { CJK , FullLeft , FullRight }
1528   {
1529     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
1530     { \_xeCJK_switch_font:nn {#2} {#1} }
1531   }
1532 }
(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。

```

1533 \cs_if_exist:NF \XeTeXglyphbouds
1534 {
1535   \_xeCJK_msg_new:nn { XeTeX-too-old }
1536   {
1537     \token_to_str:N \XeTeXglyphbouds \ is~not~defined.\
1538     CJK~punctuation~kerning~will~not~be~available.\
1539     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
1540   }
1541   \_xeCJK_error:n { XeTeX-too-old }
1542   \AtEndOfPackage
1543   {
1544     \keys_define:nn { xeCJK / options }
1545     {
1546       PunctStyle / unknown .code:n =
1547       { \_xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
1548     }
1549     \seq_gclear:N \g__xeCJK_punct_style_seq
1550     \xeCJKsetup { PunctStyle = plain }
1551   }
1552 }

```

`\xeCJKsetwidth` 手动设置参数中的标点符号的宽度。

```

1553 \NewDocumentCommand \xeCJKsetwidth { s m m }
1554 {
1555   \IfBooleanTF {#1}
1556   {
1557     \tl_map_inline:xn {#2}
1558     { \tl_gset:cn { g__xeCJK_punct_bound_width/##1/tl } {#3} }
1559   }
1560   {
1561     \tl_map_inline:xn {#2}
1562     { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#3} }
1563   }

```

```

1564 }
1565 \@onlypreamble \xeCJKsetwidth
1566 \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。

```

1567 \NewDocumentCommand \xeCJKsetkern { m m m }
1568 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
1569 \@onlypreamble \xeCJKsetkern
(End definition for \xeCJKsetkern. This function is documented on page 10.)

```

```

\c__xeCJK_left_tl
\c__xeCJK_right_tl
1570 \tl_const:Nn \c__xeCJK_left_tl { left }
1571 \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 1572 \keys_define:nn { xeCJK / options }
LongPunct    1573 {
MiddlePunct  1574   AllowBreakBetweenPuncts .choice: ,
PunctWidth   1575   AllowBreakBetweenPuncts / true .code:n =
PunctBoundWidth 1576   {
1577       \bool_set_true:N \l__xeCJK_punct_breakable_bool
1578       \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
1579   } ,
1580   AllowBreakBetweenPuncts / false .code:n =
1581   {
1582       \bool_set_false:N \l__xeCJK_punct_breakable_bool
1583       \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_nobreak_kern:NN
1584   } ,
1585   AllowBreakBetweenPuncts .default:n = { true } ,
1586   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
1587   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
1588   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
1589   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
1590   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
1591   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
1592   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
1593   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
1594   MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
1595   PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
1596   PunctBoundWidth .tl_gset:N = \g__xeCJK_punct_bound_width_tl ,
1597   PunctWidth .value_required: ,
1598   PunctBoundWidth .value_required:
1599 }
1600 \bool_new:N \l__xeCJK_punct_breakable_bool
(End definition for AllowBreakBetweenPuncts and others. These functions are documented on page 5.)

```

相关选项定义的辅助函数。

```

1601 \clist_new:N \g__xeCJK_special_punct_clist
1602 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
1603 \cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
1604 \cs_new_nopar:Npn \__xeCJK_special_punct_tl:n #1#2 { g__xeCJK_special_punct_#1_#2_tl }
1605 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1606 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
1607 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1608 {
1609   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
1610   { \cs_undefine:c { \__xeCJK_special_punct_tl:n {#1} {##1} } }
1611   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
1612   \tl_map_inline:xn {#2}
1613   {
1614     \tl_new:c { \__xeCJK_special_punct_tl:n {#1} {##1} }
1615     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1616   }
1617 }
1618 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1619 {

```



```

1620 \tl_map_inline:xn {#2}
1621 {
1622   \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
1623   {
1624     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} } {##1} }
1625     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1626   }
1627 }
1628 }
1629 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1630 {
1631   \tl_map_inline:xn {#2}
1632   {
1633     \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} } {##1} }
1634     \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1635   }
1636 }

```

判断一个标点符号是否为全角右标点和长标点符号。

```

1637 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
1638 {
1639   \if_int_compare:w \xeCJK_token_value_class:N #1 = \xeCJK_class_num:n { FullRight }
1640   \prg_return_true: \else: \prg_return_false: \fi:
1641 }
1642 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1643 {
1644   \exp_args:Nc
1645   \prg_new_conditional:Npnn { \__xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1646   {
1647     \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
1648     \prg_return_true: \else: \prg_return_false: \fi:
1649   }
1650 }

```

一些用于记录的辅助函数。

```

1651 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
1652 { c_\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
1653 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn #1#2#3
1654 { c_\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/#3/tl }
1655 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
1656 { \use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
1657 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
1658 { \use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
1659 \cs_new_protected_nopar:Npn \__xeCJK_save_punct_dim:nnn #1#2#3
1660 { \tl_const:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } { \dim_eval:n {#3} } }
1661 \cs_new_protected_nopar:Npn \__xeCJK_save_punct_dim:nnnn #1#2#3#4
1662 { \tl_const:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } { \dim_eval:n {#4} } }

```

定义标点处理模板。

```

1663 \DeclareObjectType { xeCJK / punctuation } { \c_zero }
1664 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_zero }
1665 {
1666   enabled-global-setting : boolean = true ,
1667   fixed-punct-width      : length  = \c_max_dim ,
1668   fixed-punct-ratio      : real    = \c_one_fp ,
1669   mixed-punct-width      : length  = \KeyValue { fixed-punct-width } ,
1670   mixed-punct-ratio      : real    = \KeyValue { fixed-punct-ratio } ,
1671   middle-punct-width      : length  = \KeyValue { fixed-punct-width } ,
1672   middle-punct-ratio      : real    = \KeyValue { fixed-punct-ratio } ,
1673   fixed-margin-width      : length  = \c_max_dim ,
1674   fixed-margin-ratio      : real    = \c_one_fp ,
1675   mixed-margin-width      : length  = \KeyValue { fixed-margin-width } ,
1676   mixed-margin-ratio      : real    = \KeyValue { fixed-margin-ratio } ,
1677   middle-margin-width      : length  = \KeyValue { fixed-margin-width } ,
1678   middle-margin-ratio      : real    = \KeyValue { fixed-margin-ratio } ,
1679   bound-punct-width      : length  = \c_max_dim ,
1680   bound-punct-ratio      : real    = \c_nan_fp ,
1681   bound-margin-width      : length  = \c_max_dim ,
1682   bound-margin-ratio      : real    = \c_zero_fp ,

```

```

1683     enabled-hanging           : boolean = false ,
1684     add-min-bound-to-margin  : boolean = false ,
1685     optimize-margin          : boolean = false ,
1686     margin-minimum           : length  = \c_zero_dim ,
1687     enabled-kerning          : boolean = true  ,
1688     min-bound-to-kerning     : boolean = false ,
1689     kerning-total-width      : length  = \c_max_dim ,
1690     kerning-total-ratio      : real    = 0.75 ,
1691     optimize-kerning         : boolean = false ,
1692     same-align-margin        : length  = \c_max_dim ,
1693     same-align-ratio         : real    = \c_nan_fp ,
1694     different-align-margin   : length  = \c_max_dim ,
1695     different-align-ratio    : real    = \c_nan_fp ,
1696     kerning-margin-width     : length  = \c_max_dim ,
1697     kerning-margin-ratio     : real    = \c_one_fp ,
1698     kerning-margin-minimum   : length  = \c_zero_dim
1699 }

1700 \DeclareTemplateCode { xeCJK / punctuation } { basic } { \c_zero }
1701 {
1702     enabled-global-setting    = \l__xeCJK_enabled_global_setting_bool ,
1703     fixed-punct-width        = \l__xeCJK_fixed_punct_width_dim ,
1704     fixed-punct-ratio        = \l__xeCJK_fixed_punct_ratio_fp ,
1705     mixed-punct-width        = \l__xeCJK_mixed_punct_width_dim ,
1706     mixed-punct-ratio        = \l__xeCJK_mixed_punct_ratio_fp ,
1707     middle-punct-width       = \l__xeCJK_middle_punct_width_dim ,
1708     middle-punct-ratio       = \l__xeCJK_middle_punct_ratio_fp ,
1709     fixed-margin-width       = \l__xeCJK_fixed_margin_width_dim ,
1710     fixed-margin-ratio       = \l__xeCJK_fixed_margin_ratio_fp ,
1711     mixed-margin-width       = \l__xeCJK_mixed_margin_width_dim ,
1712     mixed-margin-ratio       = \l__xeCJK_mixed_margin_ratio_fp ,
1713     middle-margin-width      = \l__xeCJK_middle_margin_width_dim ,
1714     middle-margin-ratio      = \l__xeCJK_middle_margin_ratio_fp ,
1715     bound-punct-width        = \l__xeCJK_bound_punct_width_dim ,
1716     bound-punct-ratio        = \l__xeCJK_bound_punct_ratio_fp ,
1717     bound-margin-width       = \l__xeCJK_bound_margin_width_dim ,
1718     bound-margin-ratio       = \l__xeCJK_bound_margin_ratio_fp ,
1719     enabled-hanging          = \l__xeCJK_enabled_hanging_bool ,
1720     add-min-bound-to-margin  = \l__xeCJK_add_min_bound_to_margin_bool ,
1721     optimize-margin          = \l__xeCJK_optimize_margin_bool ,
1722     margin-minimum           = \l__xeCJK_margin_minimum_dim ,
1723     enabled-kerning          = \l__xeCJK_enabled_kerning_bool ,
1724     min-bound-to-kerning     = \l__xeCJK_min_bound_to_kerning_bool ,
1725     kerning-total-width      = \l__xeCJK_kerning_total_width_dim ,
1726     kerning-total-ratio      = \l__xeCJK_kerning_total_ratio_fp ,
1727     optimize-kerning         = \l__xeCJK_optimize_kerning_bool ,
1728     same-align-margin        = \l__xeCJK_same_align_margin_dim ,
1729     same-align-ratio         = \l__xeCJK_same_align_ratio_fp ,
1730     different-align-margin   = \l__xeCJK_different_align_margin_dim ,
1731     different-align-ratio    = \l__xeCJK_different_align_ratio_fp ,
1732     kerning-margin-width     = \l__xeCJK_kerning_margin_width_dim ,
1733     kerning-margin-ratio     = \l__xeCJK_kerning_margin_ratio_fp ,
1734     kerning-margin-minimum   = \l__xeCJK_kerning_margin_minimum_dim
1735 }
1736 { \AssignTemplateKeys }

```

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

```

1737 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
1738 {
1739     \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }
1740     {
1741         \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1742         {
1743             \__xeCJK_save_punct_dim:nnnn { rule } {#1} {#2} { \c_zero_dim }
1744             \__xeCJK_save_punct_dim:nnnn { glue } {#1} {#2} { \c_zero_dim }
1745             \__xeCJK_save_punct_dim:nnnn { plus } {#1} {#2} { \c_zero_dim }
1746             \__xeCJK_save_punct_dim:nnnn { minus } {#1} {#2} { \c_zero_dim }
1747             \__xeCJK_save_punct_dim:nnnn { offset } {#1} {#2} { \c_zero_dim }
1748             \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }

```

```

1749 \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
1750 }
1751 {
1752   { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
1753   \dim_set:Nn \l__xeCJK_bound_dim
1754   { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
1755   \dim_set:Nn \l__xeCJK_reverse_bound_dim
1756   {
1757     \__xeCJK_use_punct_dim:nnn { bound }
1758     { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl }
1759     {#2}
1760   }
1761   \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
1762   \xeCJK_punct_margin_process:NN {#1} {#2}
1763   \xeCJK_punct_offset_process:NN {#1} {#2}
1764 }
1765 }
1766 }
1767 \dim_new:N \l__xeCJK_bound_dim
1768 \dim_new:N \l__xeCJK_reverse_bound_dim
(End definition for \xeCJK_get_punct_bounds:NN.)

```

\xeCJK_get_punct_kerning:NN 标点挤压。

```

1769 \cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
1770 {
1771   \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
1772   {
1773     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1774     {
1775       \__xeCJK_save_punct_dim:nnnn { kern } {#1} {#2} { \c_zero_dim }
1776       \__xeCJK_save_punct_dim:nnnn { break_kern } {#1} {#2} { \c_zero_dim }
1777     }
1778     {
1779       \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
1780       \xeCJK_punct_kerning_process:NN {#1} {#2}
1781     }
1782   }
1783 }
1784 \cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
(End definition for \xeCJK_get_punct_kerning:NN.)

```

\xeCJK_punct_margin_process:NN

```

1785 \cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
1786 {
1787   \dim_set:Nn \l__xeCJK_tmp_dim
1788   {
1789     \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1790     {
1791       \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
1792       { \use_none:n }
1793       {
1794         \tl_if_empty:NTF \g__xeCJK_punct_width_tl
1795         { \use:n }
1796         { \g__xeCJK_punct_width_tl \use_none:n }
1797       }
1798     }
1799     { \use:n }
1800   }
1801   \__xeCJK_punct_if_middle:NNTF {#2}
1802   { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
1803   {
1804     \__xeCJK_punct_if_mixed_width:NNTF {#2}
1805     { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
1806     { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }
1807   }
1808 }
1809 }
1810 \dim_set:Nn \l__xeCJK_tmp_dim

```

```

1811 {
1812   \dim_max:nn
1813   { \l__xeCJK_margin_minimum_dim }
1814   {
1815     \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1816     {
1817       \__xeCJK_punct_if_middle:NTF {#2}
1818       {
1819         (
1820           \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1821         ) / \c_two
1822       }
1823       {
1824         \bool_if:NTF \l__xeCJK_optimize_margin_bool
1825         {
1826           \dim_max:nn
1827           { \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
1828           { \use:n }
1829           {
1830             \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim
1831             - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1832           }
1833         }
1834       }
1835     }
1836   }
1837   \bool_if:NTF \l__xeCJK_optimize_margin_bool
1838   { \dim_min:nn { \l__xeCJK_bound_dim } }
1839   { \use:n }
1840   {
1841     \__xeCJK_punct_if_middle:NTF {#2}
1842     {
1843       \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
1844       { \l__xeCJK_middle_margin_width_dim }
1845       {
1846         \fp_use:N \l__xeCJK_middle_margin_ratio_fp
1847         \etex_dimexpr:D
1848         ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / \c_two
1849         \scan_stop:
1850       }
1851     }
1852     {
1853       \__xeCJK_punct_if_mixed_width:NTF {#2}
1854       { \__xeCJK_margin_width_or_ratio:n { mixed } }
1855       { \__xeCJK_margin_width_or_ratio:n { fixed } }
1856     }
1857   }
1858 }
1859 }
1860 }
1861 \__xeCJK_save_punct_dim:nnnn { glue } {#1} {#2} { \l__xeCJK_tmp_dim }
1862 \__xeCJK_save_punct_dim:nnnn { plus } {#1} {#2}
1863 {
1864   \dim_max:nn { \c_zero_dim }
1865   {
1866     \__xeCJK_punct_if_middle:NTF {#2}
1867     {
1868       ( \__xeCJK_use_punct_dim:nn { width } {#2} -
1869         \__xeCJK_use_punct_dim:nn { dimen } {#2} ) / \c_two
1870       - \l__xeCJK_tmp_dim
1871     }
1872     { \l__xeCJK_bound_dim - \l__xeCJK_tmp_dim }
1873   }
1874 }
1875 \__xeCJK_save_punct_dim:nnnn { minus } {#1} {#2}
1876 {
1877   \dim_max:nn { \c_zero_dim }
1878   {
1879     \__xeCJK_punct_if_middle:NTF {#2}

```

```

1880         { .5 \l__xeCJK_tmp_dim }
1881         { \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim }
1882     }
1883 }
1884 }

```

(End definition for \xeCJK_punct_margin_process:NN.)

\xeCJK_punct_offset_process:NN

```

1885 \cs_new_protected_nopar:Npn \xeCJK_punct_offset_process:NN #1#2
1886 {
1887     \dim_set:Nn \l__xeCJK_tmp_dim
1888     {
1889         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1890         {
1891             \cs_if_exist_use:cTF { g__xeCJK_punct_bound_width/#2/tl }
1892             { \use_none:n }
1893             {
1894                 \tl_if_empty:NTF \g__xeCJK_punct_bound_width_tl
1895                 { \use:n }
1896                 { \g__xeCJK_punct_bound_width_tl \use_none:n }
1897             }
1898         }
1899         { \use:n }
1900         { \__xeCJK_punct_width_or_ratio:nN { bound } {#2} }
1901     }
1902     \dim_set:Nn \l__xeCJK_tmp_dim
1903     {
1904         \bool_if:NTF \l__xeCJK_enabled_hanging_bool
1905         { \use:n }
1906         { \dim_max:nn { \l__xeCJK_margin_minimum_dim } }
1907         {
1908             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1909             {
1910                 \__xeCJK_punct_if_middle:NTF {#2}
1911                 {
1912                     \l__xeCJK_tmp_dim
1913                     - ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} )
1914                     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1915                 }
1916                 {
1917                     \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim
1918                     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1919                 }
1920             }
1921             {
1922                 \bool_if:NTF \l__xeCJK_optimize_margin_bool
1923                 { \dim_min:nn { \l__xeCJK_bound_dim } }
1924                 { \use:n }
1925                 { \__xeCJK_margin_width_or_ratio:n { bound } }
1926             }
1927         }
1928     }
1929     \__xeCJK_save_punct_dim:nnnn { offset } {#1} {#2} { \l__xeCJK_tmp_dim }
1930     \__xeCJK_save_punct_dim:nnnn { rule } {#1} {#2}
1931     { - \l__xeCJK_bound_dim + \l__xeCJK_tmp_dim }
1932 }

```

(End definition for \xeCJK_punct_offset_process:NN.)

__xeCJK_punct_width_or_ratio:nN

```

1933 \cs_new_nopar:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
1934 {
1935     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
1936     { \use:c { l__xeCJK_#1_punct_width_dim } }
1937     {
1938         \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } ? \c_zero_fp
1939         { \c_max_dim }
1940         {
1941             \fp_use:c { l__xeCJK_#1_punct_ratio_fp }

```

```

1942         \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
1943     }
1944 }
1945 }
(End definition for \__xeCJK_punct_width_or_ratio:nN.)

```

_xeCJK_margin_width_or_ratio:n

```

1946 \cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
1947 {
1948     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
1949     { \use:c { l__xeCJK_#1_margin_width_dim } }
1950     {
1951         \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
1952         \etex_dimexpr:D \l__xeCJK_bound_dim \scan_stop:
1953     }
1954     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
1955     { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
1956 }
(End definition for \__xeCJK_margin_width_or_ratio:n.)

```

\xeCJK_punct_kerning_process:NN

```

1957 \cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
1958 {
1959     \dim_set:Nn \l__xeCJK_tmp_dim
1960     {
1961         \bool_if:nTF
1962         {
1963             \l__xeCJK_enabled_global_setting_bool &&
1964             \tl_if_exist_p:c { g__xeCJK_punct/kern/#1/#2/tl }
1965         }
1966         { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }
1967         {
1968             \bool_if:NTF \l__xeCJK_enabled_kerning_bool
1969             { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
1970             { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
1971         }
1972     }
1973     \__xeCJK_save_punct_dim:nnnn { kern } {#1} {#2}
1974     {
1975         \l__xeCJK_tmp_dim
1976         - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1977         - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} )
1978     }
1979     \__xeCJK_punct_if_right:NF {#2}
1980     {
1981         \__xeCJK_punct_if_right:NT {#1}
1982         {
1983             \__xeCJK_save_punct_dim:nnnn { break_kern } {#1} {#2}
1984             {
1985                 \l__xeCJK_tmp_dim
1986                 - ( \__xeCJK_use_punct_dim:nnn { offset } \c__xeCJK_right_tl {#1} )
1987                 - ( \__xeCJK_use_punct_dim:nnn { offset } \c__xeCJK_left_tl {#2} )
1988             }
1989         }
1990     }
1991 }
(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。

```

1992 \cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
1993 {
1994     \dim_eval:n
1995     {
1996         \__xeCJK_use_punct_dim:nnn
1997         { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
1998         { \c__xeCJK_right_tl } {#1} +
1999         \__xeCJK_use_punct_dim:nnn

```

```

2000         { \_xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
2001         { \c\_xeCJK_left_tl } {#2}
2002     }
2003 }

```

(End definition for _xeCJK_original_kerning_margin:NN.)

_xeCJK_calc_kerning_margin:NN

```

2004 \cs_new_nopar:Npn \_xeCJK_calc_kerning_margin:NN #1#2
2005 {
2006     \dim_max:nn
2007     { \l\_xeCJK_kerning_margin_minimum_dim }
2008     {
2009         \bool_if:NTF \l\_xeCJK_min_bound_to_kerning_bool
2010         { \_xeCJK_punct_min_bound:NN {#1} {#2} }
2011         {
2012             \bool_if:NTF \l\_xeCJK_optimize_kerning_bool
2013             { \dim_max:nn { \_xeCJK_punct_min_bound:NN {#1} {#2} } }
2014             { \use:n }
2015             {
2016                 \dim_compare:nNnTF \l\_xeCJK_kerning_total_width_dim < \c_max_dim
2017                 { \_xeCJK_calc_kerning_margin:nNN \l\_xeCJK_kerning_total_width_dim }
2018                 {
2019                     \fp_compare:nNnTF \l\_xeCJK_kerning_total_ratio_fp ? \c_zero_fp
2020                     {
2021                         \xeCJK_if_same_class:NNTF {#1} {#2}
2022                         { \_xeCJK_kerning_width_or_ratio:nNN { same } }
2023                         { \_xeCJK_kerning_width_or_ratio:nNN { different } }
2024                     }
2025                     {
2026                         \_xeCJK_calc_kerning_margin:nNN
2027                         {
2028                             \fp_use:N \l\_xeCJK_kerning_total_ratio_fp
2029                             \etex_dimexpr:D
2030                                 \_xeCJK_use_punct_dim:nn { width } {#1} +
2031                                 \_xeCJK_use_punct_dim:nn { width } {#2}
2032                             \scan_stop:
2033                         }
2034                     }
2035                 }
2036                 {#1} {#2}
2037             }
2038         }
2039     }
2040 }

```

(End definition for _xeCJK_calc_kerning_margin:NN.)

_xeCJK_kerning_width_or_ratio:nNN

```

2041 \cs_new_nopar:Npn \_xeCJK_kerning_width_or_ratio:nNN #1#2#3
2042 {
2043     \dim_compare:nNnTF { \use:c { \l\_xeCJK_#1_align_margin_dim } } < \c_max_dim
2044     { \use:c { \l\_xeCJK_#1_align_margin_dim } }
2045     {
2046         \fp_compare:nNnTF { \use:c { \l\_xeCJK_#1_align_ratio_fp } } ? \c_zero_fp
2047         {
2048             \dim_compare:nNnTF \l\_xeCJK_kerning_margin_width_dim < \c_max_dim
2049             { \l\_xeCJK_kerning_margin_width_dim \use_none:n }
2050             { \fp_use:N \l\_xeCJK_kerning_margin_ratio_fp \use:n }
2051         }
2052         { \fp_use:c { \l\_xeCJK_#1_align_ratio_fp } \use:n }
2053         { \etex_dimexpr:D \_xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
2054     }
2055 }

```

(End definition for _xeCJK_kerning_width_or_ratio:nNN.)

_xeCJK_punct_min_bound:NN

```

2056 \cs_new_nopar:Npn \_xeCJK_punct_min_bound:NN #1#2
2057 {

```



```

2058 \dim_max:nn
2059 {
2060   \dim_min:nn
2061   { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} }
2062   { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
2063 }
2064 {
2065   \dim_min:nn
2066   { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} }
2067   { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
2068 }
2069 }
(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。

```

2070 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:nnN #1#2#3
2071 {
2072   \dim_eval:n
2073   {
2074     (#1)
2075     - ( \__xeCJK_use_punct_dim:nnn
2076         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
2077         { \c__xeCJK_left_tl } {#2} )
2078     - ( \__xeCJK_use_punct_dim:nnn
2079         { \__xeCJK_punct_if_right:NTF {#3} { glue } { bound } }
2080         { \c__xeCJK_right_tl } {#3} )
2081     - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
2082     - ( \__xeCJK_use_punct_dim:nn { dimen } {#3} )
2083   }
2084 }
(End definition for \__xeCJK_calc_kerning_margin:nnN.)

```

\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。

```

2085 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
2086 {
2087   \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_left_tl {#1}
2088   { \xeCJK_glyph_bounds:NN \c_one {#1} }
2089   \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_right_tl {#1}
2090   { \xeCJK_glyph_bounds:NN \c_three {#1} }
2091   \dim_set:Nn \l__xeCJK_tmp_dim
2092   {
2093     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} ) +
2094     ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
2095   }
2096   \__xeCJK_save_punct_dim:nnn { width } {#1}
2097   { \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
2098   \__xeCJK_save_punct_dim:nnn { dimen } {#1}
2099   { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim }
2100   \__xeCJK_punct_if_long:NT {#1}
2101   {
2102     \__xeCJK_save_punct_dim:nnnn { kern } {#1} {#1}
2103     {
2104       \str_case:nnTF {#1}
2105       { { ^^^^2025 } { } { ^^^^2026 } { } }
2106       { \c_zero_dim }
2107       { - \l__xeCJK_tmp_dim }
2108     }
2109   }
2110 }
2111 \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。

```

2112 \cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
2113 {
2114   \dim_use:N \XeTeXglyphbounds #1 ~

```

```

2115 \XeTeXcharglyph \xeCJK_token_value_charcode:N #2 \exp_stop_f:
2116 }
(End definition for \xeCJK_glyph_bounds:NN)

```

PunctStyle

```

2117 \keys_define:nn { xecjk / options }
2118 {
2119   PunctStyle .choice: ,
2120   PunctStyle .default:n = { quanjiao } ,
2121   PunctStyle / halfwidth .meta:n = { PunctStyle = banjiao } ,
2122   PunctStyle / fullwidth .meta:n = { PunctStyle = quanjiao } ,
2123   PunctStyle / mixedwidth .meta:n = { PunctStyle = kaiming } ,
2124   PunctStyle / marginkerning .meta:n = { PunctStyle = hangmoban_jiao } ,
2125   PunctStyle / plain .code:n =
2126     { \tl_set_eq:NN \l_xecjk_punct_style_tl \c__xecjk_punct_style_plain_tl } ,
2127   PunctStyle / unknown .code:n =
2128     {
2129       \IfInstanceExistTF { xecjk / punctuation } { \l_keys_value_tl }
2130       { \tl_set:Nx \l_xecjk_punct_style_tl { \l_keys_value_tl } }
2131       { \__xecjk_error:nx { punct-style-unknown } { \l_keys_value_tl } }
2132     }
2133 }
2134 \tl_new:N \l_xecjk_punct_style_tl
2135 \tl_const:Nn \c__xecjk_punct_style_plain_tl { plain }
2136 \__xecjk_msg_new:nn { punct-style-unknown }
2137 {
2138   Punctuation~style~"#1"~is~unknown. \\\
2139   The~available~styles~are~listed~as~follow.\\
2140   "plain,~\seq_use:Nnnn \g__xecjk_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
2141 }
(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。

```

2142 \NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
2143 {
2144   \IfInstanceExistTF { xecjk / punctuation } {#1}
2145     { \__xecjk_warning:nx { punct-style-already-defined } {#1} }
2146     { \seq_gput_right:Nx \g__xecjk_punct_style_seq {#1} }
2147   \exp_args:Nnx \DeclareInstance { xecjk / punctuation } {#1} { basic } {#2}
2148 }
2149 \seq_new:N \g__xecjk_punct_style_seq
2150 \__xecjk_msg_new:nn { punct-style-already-defined }
2151 {
2152   Punctuation~style~"#1"~is~already~defined!. \\\
2153   The~existing~style~of~"#1"~will~be~overwritten.\\
2154 }
2155 \@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。

```

2156 \NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
2157 {
2158   \IfInstanceExistTF { xecjk / punctuation } {#1}
2159     { \exp_args:Nnx \EditInstance { xecjk / punctuation } {#1} {#2} }
2160     { \__xecjk_error:nx { punct-style-unknown } {#1} }
2161 }
2162 \@onlypreamble \xeCJKEditPunctStyle
(End definition for \xeCJKEditPunctStyle. This function is documented on page 10.)
默认设置即为全角格式。

```

```

2163 \xeCJKDeclarePunctStyle { quanjiao } { }
2164 \xeCJKDeclarePunctStyle { hangmoban_jiao } { enabled-kerning = false }
2165 \xeCJKDeclarePunctStyle { banjiao }
2166 {
2167   fixed-punct-ratio = 0.5 ,
2168   optimize-margin = true ,
2169   kerning-total-ratio = 0.5 ,

```

```

2170     optimize-kerning      = true
2171 }

2172 \xeCJKDeclarePunctStyle { kaiming }
2173 {
2174     fixed-punct-ratio    = 0.5 ,
2175     mixed-punct-ratio    = 0.8 ,
2176     optimize-margin      = true ,
2177     kerning-total-ratio  = 0.5 ,
2178     optimize-kerning     = true
2179 }

2180 \xeCJKDeclarePunctStyle { CCT }
2181 {
2182     fixed-punct-ratio    = 0.7 ,
2183     optimize-margin      = true ,
2184     kerning-total-ratio  = 0.6 ,
2185     optimize-kerning     = true
2186 }

```

5.11 后备字体

AutoFallBack 后备字体的宏包选项声明。

```

2187 \keys_define:nn { xeCJK / options }
2188 {
2189     AutoFallBack .choice: ,
2190     AutoFallBack / true .code:n =
2191     {
2192         \cs_if_eq:NNF \CJKsymbol \xeCJK_fallback_test_glyph:N
2193         {
2194             \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
2195             \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
2196         }
2197     } ,
2198     AutoFallBack / false .code:n =
2199     {
2200         \cs_if_eq:NNT \CJKsymbol \xeCJK_fallback_test_glyph:N
2201         { \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N }
2202     } ,
2203     AutoFallBack .default:n = { true } ,
2204     fallback .meta:n = { AutoFallBack = true }
2205 }

```

(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。

```

2206 \cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
2207 {
2208     \xeCJK_glyph_if_exist:NTF {#1}
2209     { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
2210     {
2211         \xeCJK_class_group_begin:
2212         \tl_set_eq:NN \l__xeCJK_fallback_family_tl \l_xeCJK_family_tl
2213         \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_fallback_family_tl/FallBack }
2214         \xeCJK_class_group_end:
2215     }
2216 }

```

(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。当前字体族没有备用字体时,使用 \CJKfamilydefault 的设置。

```

2217 \cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
2218 {
2219     \xeCJK_family_if_exist:XTF {#2}
2220     {
2221         \xeCJK_select_font:x {#2}
2222         \tl_set:Nx \l__xeCJK_fallback_family_tl {#2}
2223         \xeCJK_glyph_if_exist:NTF {#1}
2224         { \__xeCJK_fallback_save_CJKsymbol:N {#1} }

```

```

2225     { \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_fallback_family_tl/FallBack } }
2226   }
2227   {
2228     \str_if_eq_x:nnTF { \CJKfamilydefault } { \l__xeCJK_family_tl }
2229     {
2230       \__xeCJK_warning:nxxx { missing-glyph }
2231       { \l__xeCJK_fallback_family_tl } {#1}
2232       { \int_to_hexadecimal:n { `#1 } }
2233       \__xeCJK_fallback_save_CJKsymbol:N {#1}
2234     }
2235     {
2236       \tl_set:Nx \l__xeCJK_family_tl { \CJKfamilydefault }
2237       \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_family_tl }
2238     }
2239   }
2240 }
2241 \__xeCJK_msg_new:nn { missing-glyph }
2242 {
2243   CJKfamily~\__xeCJK_msg_family_map:n {#1}'~
2244   ( \prop_get:Nn \g__xeCJK_family_font_name_prop {#1} )~
2245   does~not~contain~glyph~`#2'~(U+#3).\
2246 }
(End definition for \xeCJK_fallback_loop:Nn.)

```

\setCJKfallbackfamilyfont

```

2247 \NewDocumentCommand \setCJKfallbackfamilyfont { m O { } m }
2248 { \use:x { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} } }
(End definition for \setCJKfallbackfamilyfont. This function is documented on page 7.)

```

\xeCJK_set_family_fallback:nnn

```

2249 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnn #1#2#3
2250 {
2251   \group_begin:
2252   \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
2253   \prop_get:NVNF \g__xeCJK_family_font_name_prop
2254   \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
2255   { \tl_clear:N \l__xeCJK_font_name_tl }
2256   \clist_map_inline:nn {#3}
2257   {
2258     \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
2259     \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
2260     \clist_put_left:Nn \l__xeCJK_sub_font_options_clist {#2}
2261     \xeCJK_set_family:VVV \l__xeCJK_fallback_family_tl
2262     \l__xeCJK_sub_font_options_clist \l__xeCJK_sub_font_name_tl
2263   }
2264   \group_end:
2265 }
2266 \tl_new:N \l__xeCJK_fallback_family_tl
(End definition for \xeCJK_set_family_fallback:nnn.)

```

5.12 CJK 字体族声明方式

```

2267 \bool_new:N \g__xeCJK_auto_fake_bold_bool
2268 \bool_new:N \g__xeCJK_auto_fake_slant_bool
2269 \fp_new:N \g__xeCJK_embolden_factor_fp
2270 \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。

```

2271 \keys_define:nn { xeCJK / options }
2272 {
2273   AutoFakeBold .choices:nn = { true , false }
2274   { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,
2275   AutoFakeBold / unknown .code:n =
2276   {
2277     \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
2278     \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
2279   } ,

```

```

2280     AutoFakeBold .default:n = { true } ,
2281     AutoFakeSlant .choices:nn = { true , false }
2282     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
2283     AutoFakeSlant / unknown .code:n =
2284     {
2285         \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
2286         \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
2287     } ,
2288     AutoFakeSlant .default:n = { true } ,
2289     EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
2290     SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
2291     BoldFont .meta:n = { AutoFakeBold = true } ,
2292     boldfont .meta:n = { AutoFakeBold = true } ,
2293     SlantFont .meta:n = { AutoFakeSlant = true } ,
2294     slantfont .meta:n = { AutoFakeSlant = true }
2295 }

```

(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 2296 \seq_new:N \g__xeCJK_sub_key_seq
2297 \cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
2298 {
2299     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
2300     \keys_define:nn { xeCJK / features }
2301     {
2302         #1 .code:n =
2303         {
2304             \tl_if_blank:nTF {##1}
2305             {
2306                 \prop_clear:N \l__xeCJK_sub_key_prop
2307                 \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
2308                 \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
2309             }
2310             {
2311                 \str_if_eq:nnTF {##1} { * }
2312                 { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
2313                 { \__xeCJK_get_sub_features:nn {#1} {##1} }
2314             }
2315         } ,
2316         #1 .default:n = { }
2317     }
2318 }

```

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

__xeCJK_get_sub_features:nn

```

\__xeCJK_get_sub_features:w 2319 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
2320 {
2321     \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
2322     \clist_clear:N \l__xeCJK_sub_font_options_clist
2323     \exp_after:wN \__xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
2324     \q_mark [ \q_nil ] \q_mark \q_stop
2325     \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
2326     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
2327     { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
2328     \prop_put:Nnx \l__xeCJK_sub_key_prop {#1}
2329     {
2330         { \exp_not:V \l__xeCJK_sub_font_options_clist }
2331         { \exp_not:V \l__xeCJK_sub_font_name_tl }
2332     }
2333 }
2334 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
2335 {
2336     \quark_if_nil:nTF {#2}
2337     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
2338     {
2339         \tl_set:Nx \l__xeCJK_sub_font_name_tl
2340         { \xeCJK_tl_remove_outer_braces:n {#3} }
2341         \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl

```

```

2342         { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
2343         { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
2344     }
2345 }
2346 \tl_new:N \l__xeCJK_sub_family_name_tl
2347 \tl_new:N \l__xeCJK_sub_font_name_tl
2348 \clist_new:N \l__xeCJK_sub_font_options_clist
2349 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
2350 \cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
(End definition for \__xeCJK_get_sub_features:nn and \__xeCJK_get_sub_features:w)

```

FallBack

```

2351 \xeCJK_new_sub_key:n { FallBack }
(End definition for FallBack. This function is documented on page 7.)

```

BoldFont 调用字体的属性声明, 同 fontspec 宏包。

```

2352 \keys_define:nn { xeCJK / features }
2353 {
2354     BoldFont .tl_set:N = \l__xeCJK_font_name_bf_tl ,
2355     ItalicFont .tl_set:N = \l__xeCJK_font_name_it_tl
2356 }
(End definition for BoldFont and ItalicFont.)

```

AutoFakeBold

AutoFakeSlant

```

2357 \keys_define:nn { xeCJK / features }
2358 {
2359     AutoFakeBold .choice: ,
2360     AutoFakeBold / true .code:n =
2361     {
2362         \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
2363         \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
2364     } ,
2365     AutoFakeBold / false .code:n =
2366     { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
2367     AutoFakeBold / unknown .code:n =
2368     {
2369         \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
2370         \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
2371     } ,
2372     AutoFakeBold .default:n = { true } ,
2373     AutoFakeSlant .choice: ,
2374     AutoFakeSlant / true .code:n =
2375     {
2376         \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
2377         \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
2378     } ,
2379     AutoFakeSlant / false .code:n =
2380     { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
2381     AutoFakeSlant / unknown .code:n =
2382     {
2383         \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
2384         \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
2385     } ,
2386     AutoFakeSlant .default:n = { true }
2387 }
(End definition for AutoFakeBold and AutoFakeSlant.)

```

__xeCJK_set_family_initial:

```

2388 \cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
2389 {
2390     \int_gincr:N \g__xeCJK_family_int
2391     \prop_clear:N \l__xeCJK_sub_key_prop
2392     \tl_clear:N \l__xeCJK_font_name_bf_tl
2393     \tl_clear:N \l__xeCJK_font_name_it_tl
2394     \clist_clear:N \l__xeCJK_fontspec_options_clist
2395     \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
2396     \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool

```

```

2397 \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
2398 \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
2399 }
2400 \int_new:N \g__xeCJK_family_int
2401 \prop_new:N \l__xeCJK_sub_key_prop
2402 \clist_new:N \l__xeCJK_fontspec_options_clist
2403 \bool_new:N \l__xeCJK_auto_fake_bold_bool
2404 \bool_new:N \l__xeCJK_auto_fake_slant_bool
2405 \fp_new:N \l__xeCJK_embolden_factor_fp
2406 \fp_new:N \l__xeCJK_slant_factor_fp

```

(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。

```

2407 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
2408 {
2409   \group_begin:
2410   \__xeCJK_set_family_initial:
2411   \tl_set:Nn \l__xeCJK_family_name_tl {#1}
2412   \clist_set:Nn \l__xeCJK_font_options_clist {#2}
2413   \tl_set:Nn \l__xeCJK_font_name_tl {#3}
2414   \clist_concat:NNN \l__xeCJK_font_options_clist
2415   \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
2416   \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_options_clist
2417   \keys_set_known:nVN { xeCJK / features }
2418   \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
2419   \__xeCJK_parse_font_shape:
2420   \__xeCJK_check_family:V \l__xeCJK_family_name_tl
2421   \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
2422   \__xeCJK_save_family_info:
2423   \__xeCJK_set_sub_block_family:
2424   \group_end:
2425 }
2426 \tl_new:N \l__xeCJK_family_name_tl
2427 \tl_new:N \l__xeCJK_font_name_tl
2428 \clist_new:N \l__xeCJK_font_options_clist
2429 \cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , VVV , Voo }
2430 \cs_new_protected_nopar:Npn \xeCJK_set_family:xxx #1#2#3
2431 { \use:x { \xeCJK_set_family:nnn {#1} {#2} {#3} } }

```

(End definition for \xeCJK_set_family:nnn.)

__xeCJK_remove_duplicate_keys:N

```

2432 \cs_new_protected_nopar:Npn \__xeCJK_remove_duplicate_keys:N #1
2433 {
2434   \prop_clear:N \l__xeCJK_font_options_prop
2435   \keyval_parse:NNV \__xeCJK_prop_put_aux:n \__xeCJK_prop_put_aux:nn #1
2436   \clist_clear:N #1
2437   \prop_map_inline:Nn \l__xeCJK_font_options_prop
2438   {
2439     \tl_set:No \l__xeCJK_tmp_tl { \use_ii:nn ##2 }
2440     \tl_if_blank:VTF \l__xeCJK_tmp_tl
2441     { \clist_put_right:No #1 { \use_i:nn ##2 } }
2442     {
2443       \clist_put_right:Nx #1
2444       { \exp_not:o { \use_i:nn ##2 } = { \exp_not:V \l__xeCJK_tmp_tl } }
2445     }
2446   }
2447 }
2448 \prop_new:N \l__xeCJK_font_options_prop
2449 \cs_generate_variant:Nn \keyval_parse:NNn { NNv }
2450 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
2451 { \prop_put:Nnn \l__xeCJK_font_options_prop {#1} { {#1} { } } }
2452 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
2453 { \prop_put:Nnn \l__xeCJK_font_options_prop {#1} { {#1} {#2} } }

```

(End definition for __xeCJK_remove_duplicate_keys:N.)

__xeCJK_gset_family_cs:x

```

2454 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_cs:x #1

```



```

2455 {
2456   \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n {#1} }
2457   {
2458     \group_begin:
2459     \exp_not:n { \cs_set_eq:NN \__xeCJK_update_family:nn \use_none:nn }
2460     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
2461     { \exp_not:V \l__xeCJK_fontspec_options_clist }
2462     { \exp_not:V \l__xeCJK_font_name_tl }
2463     \__xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
2464     \group_end:
2465   }
2466 }
2467 \tl_new:N \l__xeCJK_fontspec_family_tl
(End definition for \__xeCJK_gset_family_cs:x)

```

__xeCJK_check_family:n

```

2468 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
2469 {
2470   \prop_gpop:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
2471   {
2472     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
2473     { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
2474     \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
2475   }
2476 }
2477 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
2478 \__xeCJK_msg_new:nn { CJKfamily-redef }
2479 { Redefining~CJKfamily~\__xeCJK_msg_family_map:n {#1}'~(#{2}). }
(End definition for \__xeCJK_check_family:n)

```

__xeCJK_parse_font_shape:

```

2480 \cs_new_protected_nopar:Npn \__xeCJK_parse_font_shape:
2481 {
2482   \tl_if_blank:VTF \l__xeCJK_font_name_bf_tl
2483   {
2484     \bool_if:NT \l__xeCJK_auto_fake_bold_bool
2485     {
2486       \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2487       { AutoFakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
2488     }
2489   }
2490   {
2491     \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2492     { BoldFont = { \exp_not:V \l__xeCJK_font_name_bf_tl } }
2493   }
2494   \tl_if_blank:VTF \l__xeCJK_font_name_it_tl
2495   {
2496     \bool_if:NT \l__xeCJK_auto_fake_slant_bool
2497     {
2498       \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2499       { AutoFakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
2500     }
2501   }
2502   {
2503     \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2504     { ItalicFont = { \exp_not:V \l__xeCJK_font_name_it_tl } }
2505   }
2506 }
(End definition for \__xeCJK_parse_font_shape:.)

```

\g__xeCJK_family_name_prop

```

\g__xeCJK_family_font_name_prop 2507 \prop_new:N \g__xeCJK_family_name_prop
\g__xeCJK_family_font_options_prop 2508 \prop_new:N \g__xeCJK_family_font_name_prop
2509 \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:

```
2510 \cs_new_protected_nopar:Npn \\_xeCJK_save_family_info:
2511 {
2512   \prop_gput:NVV \g__xeCJK_family_font_name_prop
2513   \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
2514   \prop_gput:NVV \g__xeCJK_family_font_options_prop
2515   \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
2516 }
(End definition for \\_xeCJK_save_family_info:.)
```

_xeCJK_set_sub_block_family:

```
2517 \cs_new_protected_nopar:Npn \\_xeCJK_set_sub_block_family:
2518 {
2519   \prop_map_inline:Nn \l__xeCJK_sub_key_prop
2520   {
2521     \tl_set:Nx \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/##1 }
2522     \quark_if_no_value:nTF {##2}
2523     { \_xeCJK_copy_sub_family:n {##1} }
2524     {
2525       \xeCJK_set_family:Voo \l__xeCJK_sub_family_name_tl
2526       { \use_i:nn ##2 } { \use_ii:nn ##2 }
2527     }
2528   }
2529 }
2530 \cs_new_protected_nopar:Npn \\_xeCJK_copy_sub_family:n #1
2531 {
2532   \_xeCJK_check_family:V \l__xeCJK_sub_family_name_tl
2533   \prop_get:NVNT \g__xeCJK_family_font_name_prop
2534   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
2535   {
2536     \prop_gput:NVV \g__xeCJK_family_font_name_prop
2537     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
2538   }
2539   \prop_get:NVNT \g__xeCJK_family_font_options_prop
2540   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
2541   {
2542     \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
2543     \prop_gput:NVV \g__xeCJK_family_font_options_prop
2544     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist
2545   }
2546   \cs_gset_protected_nopar:cpx
2547   { \_xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
2548   {
2549     \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }
2550     {
2551       \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
2552       { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_fontspec_family_tl
2553       {
2554         \_xeCJK_gset_family_nfss_cs:xx
2555         { \l__xeCJK_sub_family_name_tl }
2556         { \exp_not:N \l__xeCJK_fontspec_family_tl }
2557       }
2558     }
2559   }
2560 }
(End definition for \\_xeCJK_set_sub_block_family:.)
```

_xeCJK_copy_family:nn

```
2561 \cs_new_protected_nopar:Npn \\_xeCJK_copy_family:nn #1#2
2562 {
2563   \xeCJK_family_if_exist:xT {#2}
2564   {
2565     \tl_map_inline:nn
2566     {
2567       \g__xeCJK_family_name_prop
2568       \g__xeCJK_family_font_name_prop
2569       \g__xeCJK_family_font_options_prop
2570     }
2571   }
```

```

2571         {
2572             \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
2573             { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmp_tl }
2574         }
2575         \cs_gset_eq:cc
2576         { \__xeCJK_family_nfss_csname:n {#1} }
2577         { \__xeCJK_family_nfss_csname:n {#2} }
2578     }
2579 }
2580 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }
(End definition for \__xeCJK_copy_family:nn.)

```

5.13 字体切换

缓存当前字体的原始格式,以加速编译。

```

\l_xeCJK_current_font_tl
\__xeCJK_select_font:
\__xeCJK_select_font:x
2581 \tl_new:N \l_xeCJK_current_font_tl
2582 \tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
2583 \cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
2584 \cs_new_protected_nopar:Npn \xeCJK_select_font:
2585 {
2586     \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
2587     {
2588         \tl_set:Nx \l__xeCJK_current_coor_tl { \l_xeCJK_current_font_tl }
2589         \__xeCJK_family_use:x { \l_xeCJK_family_tl }
2590         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2591     }
2592 }
2593 \cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
2594 {
2595     \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
2596     {
2597         \tl_set:Nx \l__xeCJK_current_coor_tl { \__xeCJK_font_csname:n {#1} }
2598         \__xeCJK_family_use:x {#1}
2599         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2600     }
2601 }
2602 \tl_new:N \l__xeCJK_current_coor_tl
2603 \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.)

```

两个 CJK 分区之间的字体切换。

```

\__xeCJK_switch_font:nn
2604 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
2605 {
2606     \str_if_eq:nnF {#1} {#2}
2607     {
2608         \__xeCJK_info:nxx { CJK-block } {#1} {#2}
2609         \str_if_eq:nnTF {#2} { CJK }
2610         { \xeCJK_select_font: }
2611         { \__xeCJK_block_select_font:n {#2} }
2612     }
2613 }
2614 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~`#1'~to~`#2'. }
(End definition for \__xeCJK_switch_font:nn.)

```

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。

```

\__xeCJK_block_select_font:n
2615 \cs_new_protected_nopar:Npn \__xeCJK_block_select_font:n #1
2616 {
2617     \cs_if_exist_use:cF { \__xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
2618     {
2619         \tl_set:Nx \l__xeCJK_current_coor_tl
2620         { \__xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
2621         \xeCJK_family_if_exist:xF { \l_xeCJK_family_tl/#1 }
2622         {
2623             \__xeCJK_copy_family:xx { \l_xeCJK_family_tl/#1 }
2624             {
2625                 \cs_if_exist:cTF

```

```

2626             { \_xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
2627             { \CJKfamilydefault/#1 } { \l_xeCJK_family_tl }
2628         }
2629     }
2630     \_xeCJK_family_use:x { \l_xeCJK_family_tl/#1 }
2631     \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coors_tl }
2632 }
2633 }

```

(End definition for _xeCJK_block_select_font:n)

```

\_xeCJK_family_csname:n
  \_xeCJK_family_nfss_csname:n 2634 \cs_new_nopar:Npn \_xeCJK_family_csname:n #1 { xeCJK/family/#1 }
\_xeCJK_family_use:x 2635 \cs_new_nopar:Npn \_xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
\_xeCJK_gset_family_nfss_cs:xx 2636 \cs_new_nopar:Npn \_xeCJK_family_use:x #1 { \use:c { \_xeCJK_family_nfss_csname:n {#1} } }
2637 \cs_new_protected_nopar:Npn \_xeCJK_gset_family_nfss_cs:xx #1#2
2638 {
2639   \prop_gput:Nxx \g__xeCJK_family_name_prop {#1} {#2}
2640   \cs_gset_protected_nopar:cpx { \_xeCJK_family_nfss_csname:n {#1} }
2641   {
2642     \exp_not:N \fontencoding { \c__xeCJK_encoding_tl }
2643     \tl_set:Nx \exp_not:N \f@family {#2}
2644     \exp_not:N \selectfont
2645   }
2646 }
2647 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
(End definition for \_xeCJK_family_csname:n and others.)

```

```

\_xeCJK_family_if_exist:xTF
2648 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
2649 {
2650   \cs_if_exist:cTF { \_xeCJK_family_nfss_csname:n {#1} }
2651   { \use_i:nn }
2652   { \cs_if_exist_use:cTF { \_xeCJK_family_csname:n {#1} } }
2653   { \prg_return_true: } { \prg_return_false: }
2654 }
(End definition for \xeCJK_family_if_exist:xTF)

```

\CJKfamily 用于切换 CJK 字体族。

```

2655 \NewDocumentCommand \CJKfamily { t+ t- m }
2656 {
2657   \xeCJK_if_blank_x:nTF {#3}
2658   {
2659     \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
2660     \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
2661   }
2662   {
2663     \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
2664     {
2665       \xeCJK_family_if_exist:xTF {#3}
2666       {
2667         \tl_set:Nx \l_xeCJK_family_tl {#3}
2668         \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2669         \IfBooleanT {#1} { \_xeCJK_family_use:x {#3} }
2670       }
2671       { \_xeCJK_family_unknown_warning:x {#3} }
2672     }
2673   }
2674   \tex_ignorespaces:D
2675 }
2676 \cs_new_protected_nopar:Npn \xeCJK_switch_family:n #1
2677 {
2678   \xeCJK_family_if_exist:xTF {#1}
2679   {
2680     \tl_set:Nx \l_xeCJK_family_tl {#1}
2681     \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2682   }
2683   { \_xeCJK_family_unknown_warning:x {#1} }
2684 }

```

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

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

```
2685 \tl_new:N \l_xeCJK_family_tl
```

(End definition for \l_xeCJK_family_tl.)

__xeCJK_gobble_CJKfamily:

```
2686 \cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
```

```
2687 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
```

```
2688 \DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }
```

(End definition for __xeCJK_gobble_CJKfamily:.)

\xeCJK_family_if_exist_use:x

```
2689 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
```

```
2690 {
```

```
2691   \xeCJK_family_if_exist:xTF {#1}
```

```
2692   { \__xeCJK_family_use:x {#1} }
```

```
2693   { \__xeCJK_family_unknown_warning:x {#1} }
```

```
2694 }
```

(End definition for \xeCJK_family_if_exist_use:x.)

_xeCJK_family_unknown_warning:n

```
2695 \cs_new_protected_nopar:Npn \_xeCJK_family_unknown_warning:n #1
```

```
2696 {
```

```
2697   \prop_if_empty:NF \g__xeCJK_family_font_name_prop
```

```
2698   {
```

```
2699     \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
```

```
2700     {
```

```
2701       \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
```

```
2702       \_xeCJK_warning:nx { CJKfamily-Unknown } {#1}
```

```
2703     }
```

```
2704   }
```

```
2705 }
```

```
2706 \cs_generate_variant:Nn \_xeCJK_family_unknown_warning:n { x }
```

```
2707 \seq_new:N \g__xeCJK_unknown_family_seq
```

```
2708 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
```

```
2709 {
```

```
2710   Unknown~CJK~family~\_xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\\
```

```
2711   Try~to~use~\_xeCJK_msg_def_family_map:n {#1}'~to~define~it.
```

```
2712 }
```

```
2713 \cs_new_nopar:Npn \__xeCJK_msg_def_family_map:n #1
```

```
2714 {
```

```
2715   \str_case:x:nnn {#1}
```

```
2716   {
```

```
2717     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
```

```
2718     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
```

```
2719     \CJKttdefault { \token_to_str:N \setCJKmonofont }
```

```
2720   }
```

```
2721   { \token_to_str:N \setCJKfamilyfont {#1} }
```

```
2722   [...]{}...
```

```
2723 }
```

```
2724 \cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1
```

```
2725 {
```

```
2726   \str_case:x:nnn {#1}
```

```
2727   {
```

```
2728     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
```

```
2729     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
```

```
2730     \CJKttdefault { \token_to_str:N \CJKttdefault }
```

```
2731   }
```

```
2732   {#1}
```

```
2733 }
```

(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。

```
2734 \NewDocumentCommand \setCJKmainfont { 0 { } m }
```

```
2735 {
```

```
2736   \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2}
```

```

2737 \normalfont
2738 }
2739 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
2740 \NewDocumentCommand \setCJKsansfont { 0 { } m }
2741 {
2742   \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2}
2743   \normalfont
2744 }
2745 \NewDocumentCommand \setCJKmonofont { 0 { } m }
2746 {
2747   \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2}
2748   \normalfont
2749 }

```

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

```

2750 \@onlypreamble \setCJKmainfont
2751 \@onlypreamble \setCJKmathfont
2752 \@onlypreamble \setCJKsansfont
2753 \@onlypreamble \setCJKmonofont
2754 \@onlypreamble \setCJKromanfont

```

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

```

\newCJKfontfamily
\CJKfontspec
2755 \NewDocumentCommand \setCJKfamilyfont { m 0 { } m }
2756 { \xeCJK_set_family:xxx {#1} {#2} {#3} }
2757 \NewDocumentCommand \newCJKfontfamily { o m 0 { } m }
2758 {
2759   \tl_set:Nx \l__xeCJK_tmp_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
2760   \cs_new_protected_nopar:Npx #2 { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
2761   \xeCJK_set_family:xxx { \l__xeCJK_tmp_tl } {#3} {#4}
2762 }
2763 \NewDocumentCommand \CJKfontspec { 0 { } m }
2764 {
2765   \use:x { \xeCJK_fontspec:nn {#1} {#2} }
2766   \tex_ignorespaces:D
2767 }

```

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

`\xeCJK_fontspec:nn`

```

2768 \cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
2769 {
2770   \prop_get:NnNTF \g__xeCJK_fontspec_prop
2771   { CJKfontspec/#1/#2/id } \l__xeCJK_family_tl
2772   { \tl_set_eq:NN \xeCJK@family \l__xeCJK_family_tl }
2773   {
2774     \__xeCJK_fontspec:xnn
2775     { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
2776     {#1} {#2}
2777   }
2778 }
2779 \cs_new_protected_nopar:Npn \__xeCJK_fontspec:nnn #1#2#3
2780 {
2781   \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
2782   \xeCJK_set_family:nnn {#1} {#2} {#3}
2783   \xeCJK_switch_family:n {#1}
2784 }
2785 \cs_generate_variant:Nn \xeCJK_fontspec:nn { VV }
2786 \cs_generate_variant:Nn \__xeCJK_fontspec:nnn { x }
2787 \prop_new:N \g__xeCJK_fontspec_prop

```

(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
2788 \clist_new:N \g__xeCJK_default_features_clist
2789 \NewDocumentCommand \defaultCJKfontfeatures { m }
2790 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
2791 \@onlypreamble \defaultCJKfontfeatures
2792 \NewDocumentCommand \addCJKfontfeatures { s 0 { } m }
2793 {

```

```

2794 \xeCJK_add_font_features:Nxx #1 {#2} {#3}
2795 \tex_ignorespaces:D
2796 }
2797 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
(End definition for \defaultCJKfontfeatures and \addCJKfontfeatures. These functions are documented on page 6.)

```

\xeCJK_add_font_features:Nnn

```

2798 \cs_new_protected_nopar:Npn \xeCJK_add_font_features:Nnn #1#2#3
2799 {
2800   \prop_get:NVNTF \g__xeCJK_family_font_name_prop
2801   \l__xeCJK_family_tl \l__xeCJK_font_name_tl
2802   {
2803     \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
2804     \seq_map_inline:Nn \g__xeCJK_sub_key_seq
2805     { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
2806     \seq_clear:N \l__xeCJK_sub_key_seq
2807     \clist_clear:N \l__xeCJK_add_block_features_clist
2808     \clist_map_inline:nn {#2}
2809     {
2810       \seq_if_in:NnTF \g__xeCJK_sub_key_seq {##1}
2811       {
2812         \seq_put_right:Nn \l__xeCJK_sub_key_seq {##1}
2813         \__xeCJK_add_sub_class_features:n {##1}
2814       }
2815       { \__xeCJK_warning:nx { SubBlock-undefined } {##1} }
2816     }
2817     \bool_if:nT { #1 && \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
2818     {
2819       \seq_map_function:NN
2820       \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
2821     }
2822     \prop_get:NVNTF \g__xeCJK_family_font_options_prop
2823     \l__xeCJK_family_tl \l__xeCJK_font_options_clist
2824     {
2825       \bool_if:nT
2826       { \seq_if_empty_p:N \l__xeCJK_sub_key_seq || #1 }
2827       {
2828         \clist_concat:NNN \l__xeCJK_font_options_clist
2829         \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
2830       }
2831       \clist_concat:NNN \l__xeCJK_font_options_clist
2832       \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
2833     }
2834     \xeCJK_fontspec:VV \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
2835   }
2836   { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
2837 }
2838 \clist_new:N \l__xeCJK_add_font_features_clist
2839 \clist_new:N \l__xeCJK_add_block_features_clist
2840 \cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nxx , Nnx }
2841 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
2842 {
2843   \token_to_str:N \addCJKfontfeature (s)~ignored.\\
2844   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
2845 }

```

(End definition for \xeCJK_add_font_features:Nnn.)

__xeCJK_add_sub_class_features:n

```

2846 \cs_new_protected_nopar:Npn \__xeCJK_add_sub_class_features:n #1
2847 {
2848   \prop_get:NoNTF \g__xeCJK_family_font_name_prop
2849   { \l__xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
2850   {
2851     \prop_get:NoN \g__xeCJK_family_font_options_prop
2852     { \l__xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
2853   }
2854   {
2855     \prop_get:NxNTF \g__xeCJK_family_font_name_prop

```

```

2856     { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
2857     {
2858         \prop_get:NxN \g__xeCJK_family_font_options_prop
2859         { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
2860     }
2861     {
2862         \prop_get:NVN \g__xeCJK_family_font_options_prop
2863         \l__xeCJK_family_tl \l__xeCJK_sub_font_options_clist
2864         \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl
2865     }
2866 }
2867 \clist_concat:NNN \l__xeCJK_sub_font_options_clist
2868 \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
2869 \clist_put_right:Nx \l__xeCJK_add_block_features_clist
2870 {
2871     #1 =
2872     {
2873         [ \exp_not:V \l__xeCJK_sub_font_options_clist ]
2874         { \exp_not:V \l__xeCJK_sub_font_name_tl }
2875     }
2876 }
2877 }
2878 \cs_generate_variant:Nn \prop_get:NnN { Nx }
2879 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }
(End definition for \__xeCJK_add_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。
2880 \__xeCJK_at_end_preamble:n
2881 {
2882     \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
2883     \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
2884     {
2885         \tl_gset:Nx \CJKfamilydefault
2886         {
2887             \str_case_x:nnn { \familydefault }
2888             {
2889                 { \rmdefault } { \exp_not:N \CJKrmdefault }
2890                 { \sfdefault } { \exp_not:N \CJKsfdefault }
2891                 { \ttdefault } { \exp_not:N \CJKttdefault }
2892             }
2893             { \CJKfamilydefault }
2894         }
2895     }
2896     \cs_undefine:N \__xeCJK_family_default_wrap:n
2897     \prop_if_empty:NNTF \g__xeCJK_family_font_name_prop
2898     { \__xeCJK_warning:nx { no-CJKfamily } { \CJKfamilydefault } }
2899     {
2900         \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2901         {
2902             \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
2903             \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
2904             { \use:n }
2905             {
2906                 \xeCJK_family_if_exist:xTF { \CJKrmdefault }
2907                 { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
2908             }
2909             {
2910                 \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
2911                 {
2912                     \prop_map_break:n
2913                     { \tl_gset_rescan:Nnn \CJKfamilydefault { } {#1} }
2914                 }
2915             }
2916             \__xeCJK_warning:nxx { CJKfamilydefault-undefined }
2917             { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
2918         }

```



```

2919         \xeCJK_switch_family:n { \CJKfamilydefault }
2920     \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
2921 }
2922 }
2923 \__xeCJK_msg_new:nn { no-CJKfamily }
2924 {
2925     It~seems~that~you~have~not~declare~a~CJKfamily.\\
2926     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
2927     `\_xeCJK_msg_def_family_map:n {#1}'\\\\
2928     in~the~preamble~to~declare~the~default~CJKfamily.\\
2929 }
2930 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
2931 {
2932     Undefined~CJK~default~family~`\_xeCJK_msg_family_map:n {#1}'~
2933     has~been~replaced~by~`\_xeCJK_msg_family_map:n {#2}'\\\\
2934     Try~to~use~`\_xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2935 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```

2936 \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。

```

2937 \NewDocumentCommand \setCJKmathfont { 0 { } m }
2938 { \xeCJK_set_family:xxx { \c__xeCJK_math_tl } {#1} {#2} }
2939 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }

```

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

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

```

2940 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
2941 {
2942     \xeCJK_family_if_exist:xF { \c__xeCJK_math_tl }
2943     { \tl_set:Nx \l__xeCJK_tmp_tl { \c__xeCJK_math_tl } }
2944     {
2945         \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2946         { \tl_set:Nx \l__xeCJK_tmp_tl { \CJKfamilydefault } }
2947         { \use_none:nnnnn }
2948     }
2949     \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmp_tl \l__xeCJK_tmp_tl
2950     {
2951         \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmp_tl }
2952         \DeclareSymbolFont { \c__xeCJK_math_tl } { \c__xeCJK_encoding_tl }
2953         { \c__xeCJK_math_family_tl } { \mddefault } { \shapedefault }
2954         \cs_if_free:cF
2955         { \c__xeCJK_encoding_tl/\c__xeCJK_math_family_tl/\bfdefault/\shapedefault }
2956         {
2957             \SetSymbolFont { \c__xeCJK_math_tl } { bold } { \c__xeCJK_encoding_tl }
2958             { \c__xeCJK_math_family_tl } { \bfdefault } { \shapedefault }
2959         }
2960         \int_const:Nn \c__xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }
2961         \clist_concat:NNN \l__xeCJK_tmp_clist
2962         \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
2963         \clist_concat:NNN \l__xeCJK_tmp_clist
2964         \l__xeCJK_tmp_clist \c__xeCJK_FullRight_chars_clist
2965         \clist_map_inline:Nn \l__xeCJK_tmp_clist
2966         {
2967             \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
2968             { \c_zero } { \c__xeCJK_math_fam_int }
2969         }
2970     }
2971 }

```

(End definition for *\xeCJK_set_mathfont:*.)

`\xeCJK_gset_mathcode:nnnn`

```
2972 \cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
2973 {
2974   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
2975   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
2976   {
2977     \tex_global:D \xeCJK_xetex_mathcode:w
2978     \l__xeCJK_begin_int = #3 ~ #4 \l__xeCJK_begin_int
2979     \int_incr:N \l__xeCJK_begin_int
2980   }
2981 }
(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。

```
2982 \int_new:N \l__xeCJK_verb_case_int
2983 \keys_define:nn { xecjk / options }
2984 {
2985   Verb .choices:nn =
2986     { true , env , env+ , false }
2987     { \int_set_eq:NN \l__xeCJK_verb_case_int \l_keys_choice_int } ,
2988   Verb .default:n = { env }
2989 }
2990 \cs_new_protected_nopar:Npn \__xeCJK_verb_font_hook:
2991 {
2992   \if_case:w \l__xeCJK_verb_case_int
2993   \or:
2994     \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
2995     \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
2996   \or:
2997     \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
2998       { \xeCJKVerbAddon }
2999       {
3000         \cs_set_eq:NN \CJKglue \__xeCJK_zero_glue:
3001         \cs_set_eq:NN \CJKecglue \__xeCJK_zero_glue:
3002       }
3003   \or:
3004     \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
3005       { \xeCJKVerbAddon }
3006   \fi:
3007 }
3008 \__xeCJK_after_preamble:n
3009 {
3010   \cs_set_protected_nopar:Npx \verbatim@font
3011   { \exp_not:o { \verbatim@font } \__xeCJK_verb_font_hook: }
3012 }
```

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

\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。

```
3013 \NewDocumentCommand \xeCJKVerbAddon { }
3014 {
3015   \int_compare:nNnF \etex_currentgrouplevel:D = \c_zero
3016   {
3017     \bool_if:NF \l__xeCJK_listings_env_bool
3018     {
3019       \__xeCJK_set_verb_exspace:
3020       \__xeCJK_verb_addon:
3021     }
3022   }
3023 }
3024 \bool_new:N \l__xeCJK_listings_env_bool
3025 \NewDocumentCommand \xeCJKOffVerbAddon { } { }
```

```

3026 \cs_new_protected_nopar:Npn \__xeCJK_verb_addon:
3027 {
3028   \bool_if:NF \l__xeCJK_verb_addon_bool
3029   {
3030     \bool_set_true:N \l__xeCJK_verb_addon_bool
3031     \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
3032     \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
3033     \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
3034     \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
3035     \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
3036     \cs_set_eq:NN \__xeCJK_verb_CJKglue: \CJKglue
3037     \cs_set_eq:NN \__xeCJK_verb_CJKecglue: \CJKecglue
3038     \cs_set_eq:NN \__xeCJK_verb_check_for_glue: \xeCJK_check_for_glue:
3039     \cs_set_eq:NN \__xeCJK_verb_boundary:w \xeCJK_CJK_and_Boundary:w
3040     \cs_set_protected_nopar:Npx \xeCJKOffVerbAddon
3041     {
3042       \__xeCJK_reset_char_class:n { FullLeft }
3043       \__xeCJK_reset_char_class:n { FullRight }
3044       \__xeCJK_reset_char_class:n { HalfLeft }
3045       \__xeCJK_reset_char_class:n { HalfLeft }
3046       \__xeCJK_reset_char_class:n { NormalSpace }
3047       \exp_not:c
3048       { \bool_set_ \bool_if:NTF \l__xeCJK_xecglue_bool { true } { false } :N }
3049       \l__xeCJK_xecglue_bool
3050       \exp_not:n
3051       {
3052         \cs_set_eq:NN \CJKglue \__xeCJK_verb_CJKglue:
3053         \cs_set_eq:NN \CJKecglue \__xeCJK_verb_CJKecglue:
3054         \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_verb_check_for_glue:
3055         \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_boundary:w
3056       }
3057     }
3058     \__xeCJK_add_to_shipout:n { \xeCJKOffVerbAddon }
3059     \xeCJKsetup { xCJKecglue = false }
3060   }
3061   \skip_if_eq:nnTF { \l__xeCJK_verb_exspace_skip } { \c_zero_skip }
3062   {
3063     \xeCJK_cs_clear:N \CJKglue
3064     \xeCJK_cs_clear:N \CJKecglue
3065   }
3066   {
3067     \cs_set_eq:NN \CJKglue \__xeCJK_verb_ccglue:
3068     \cs_set_eq:NN \CJKecglue \__xeCJK_verb_ecglue:
3069   }
3070   \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
3071   \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_CJK_and_Boundary:w
3072 }
3073 \cs_new_protected_nopar:Npn \__xeCJK_verb_CJK_and_Boundary:w
3074 { \xeCJK_class_group_end: \CJKecglue }
3075 \cs_new_protected_nopar:Npn \__xeCJK_verb_ccglue:
3076 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_verb_exspace_skip }
3077 \cs_new_protected_nopar:Npn \__xeCJK_verb_ecglue:
3078 { \xeCJK_no_break: \skip_horizontal:n { 0.5 \l__xeCJK_verb_exspace_skip } }
3079 \cs_new_protected_nopar:Npn \__xeCJK_reset_char_class:n #1
3080 {
3081   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
3082   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
3083   { \XeTeXcharclass ##1 = \l__xeCJK_tmp_int }
3084 }
3085 \bool_new:N \l__xeCJK_verb_addon_bool
3086 \cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon

```

(End definition for `\xeCJKOffVerbAddon` and `\xeCJKVerbAddon`. These functions are documented on page 12.)

`__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。

```

3087 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
3088 {
3089   \tl_if_exist:cTF { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }

```

```

3090 {
3091   \skip_set:Nn \l__xeCJK_verb_exspace_skip
3092   { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
3093 }
3094 {
3095   \tl_set:Nx \l__xeCJK_current_coor_tl { \l_xeCJK_family_tl/\curr@fontshape }
3096   \prop_get:NVNTF \g__xeCJK_scale_family_prop
3097   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
3098   {
3099     \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
3100     \skip_zero:N \l__xeCJK_verb_exspace_skip
3101   }
3102   {
3103     \group_begin: \xeCJK_select_font: \exp_after:wN \group_end:
3104     \exp_after:wN \__xeCJK_set_verb_exspace:n
3105     \exp_after:wN { \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00 }
3106   }
3107 }
3108 }
3109 \skip_new:N \l__xeCJK_verb_exspace_skip
(End definition for \__xeCJK_set_verb_exspace:.)

```

`__xeCJK_set_verb_exspace:n` 当两个西文空格的宽度小于一个 CJK 文字的宽度时,对目前使用的 CJK 字体进行适当缩小。

```

3110 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:n #1
3111 {
3112   \skip_set:Nn \l__xeCJK_verb_exspace_skip
3113   { \c_two \tex_fontdimen:D \c_two \tex_font:D - #1 }
3114   \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
3115   {
3116     \skip_zero:N \l__xeCJK_verb_exspace_skip
3117     \use:x
3118     {
3119       \__xeCJK_set_verb_scale:nn
3120       { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } }
3121       { \dim_to_fp:n {#1} }
3122     }
3123   }
3124   {
3125     \tl_const:cx { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
3126     { \skip_use:N \l__xeCJK_verb_exspace_skip }
3127   }
3128 }
(End definition for \__xeCJK_set_verb_exspace:n.)

```

`__xeCJK_set_verb_scale:nn` 缩小 CJK 字体,并保存相关信息。

```

3129 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_scale:nn #1#2
3130 {
3131   \fp_set:Nn \l__xeCJK_scale_factor_fp { #1 / #2 }
3132   \__xeCJK_warning:nxx { scale-factor }
3133   { \fp_eval:n { round - ( \l__xeCJK_scale_factor_fp , 4 ) } }
3134   { \fp_eval:n { round + ( #2 / #1 , 4 ) } }
3135   \xeCJK_add_font_features:Nnx \c_true_bool
3136   { } { Scale = { \fp_use:N \l__xeCJK_scale_factor_fp } }
3137   \prop_gput:NVV \g__xeCJK_scale_family_prop
3138   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
3139 }
3140 \__xeCJK_msg_new:nn { scale-factor }
3141 {
3142   \token_to_str:N \xeCJKVerbAddon'~may~not~work~properly.\\
3143   You~may~set~`Scale=#1'~to~CJKfamily~
3144   \__xeCJK_msg_family_map:n { \l_xeCJK_family_tl }',\\
3145   or~set~`Scale=#2'~to~family~
3146   \str_if_eq:x:nnTF \f@family \ttdefault
3147   { \token_to_str:N \ttdefault } { \f@family }'.
3148 }
3149 \fp_new:N \l__xeCJK_scale_factor_fp
3150 \prop_new:N \g__xeCJK_scale_family_prop
(End definition for \__xeCJK_set_verb_scale:nn.)

```

`\xeCJK_visible_space:` 如果文档不使用 EU1 作为默认字体编码,那么默认的打字机字体族很可能是传统的 T_EX 字体,这时可视空格按照 OT1 编码传统一般就是字体中的 `\char32`。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,保证在当前西文字体而不是在 CJK 字体中检查有没有 U+2423。

```

3151 \cs_new_protected_nopar:Npn \xeCJK_visible_space:
3152 {
3153   \bool_if:NT \l__xeCJK_CJK_group_bool { \scan_stop: }
3154   \xeCJK_glyph_if_exist:NTF { ~~~~2423 }
3155   { ~~~~2423 }
3156   {
3157     \int_compare:nNnTF { \XeTeXfonttype \tex_font:D } = \c_zero
3158     {
3159       \str_if_eq_x:nnTF { \f@family } { \ttdefault }
3160       { \c_catcode_other_space_tl }
3161       { \textvisiblespace }
3162     }
3163     { \xeCJK_visible_space_fallback: }
3164   }
3165 }
3166 \AtEndOfPackage
3167 { \cs_gset_eq:NN \fontspec_visible_space: \xeCJK_visible_space: }
(End definition for \xeCJK_visible_space:.)

```

`\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。

```

3168 \cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
3169 { {
3170   \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
3171   { \xeCJK_set_visible_space_font: }
3172   ~~~~2423
3173 } }
(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。

```

3174 \cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
3175 {
3176   \tl_set:Nx \l__xeCJK_current_coord_tl { xeCJK/space/\curr@fontshape/\f@size }
3177   \exp_after:wN \__xeCJK_set_visible_space_size:n
3178   \exp_after:wN { \dim_use:N \tex_fontdimen:D \c_two \tex_font:D }
3179   \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coord_tl }
3180 }
3181 \cs_new_protected_nopar:Npn \__xeCJK_set_visible_space_size:n #1
3182 {
3183   \fontencoding { \g_fontspec_encoding_tl }
3184   \tl_set:Nx \f@family { lmtt }
3185   \selectfont
3186   \dim_compare:nNnF {#1} = { \tex_fontdimen:D \c_two \tex_font:D }
3187   {
3188     \fontsize
3189     {
3190       \dim_eval:n
3191       {
3192         \f@size pt *
3193         \dim_ratio:nn {#1} { \tex_fontdimen:D \c_two \tex_font:D }
3194       }
3195     }
3196     { \f@baselineskip }
3197     \selectfont
3198   }
3199 }
(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。

```
3200 \keys_define:nn { xeCJK / options }
3201 {
3202   LocalConfig .choice: ,
3203   LocalConfig / false .code:n =
3204     { \bool_gset_false:N \g__xeCJK_config_bool } ,
3205   LocalConfig / true .code:n =
3206     {
3207       \bool_gset_true:N \g__xeCJK_config_bool
3208       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
3209     } ,
3210   LocalConfig / unknown .code:n =
3211     {
3212       \bool_gset_true:N \g__xeCJK_config_bool
3213       \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
3214     } ,
3215   LocalConfig .default:n = { true }
3216 }
3217 \tl_new:N \g__xeCJK_config_name_tl
3218 \bool_new:N \g__xeCJK_config_bool
```

(End definition for *LocalConfig*. This function is documented on page 2.)

indentfirst 首行是否缩进。

```
3219 \keys_define:nn { xeCJK / options }
3220 {
3221   CJKnumber .bool_gset:N = \g__xeCJK_number_bool ,
3222   indentfirst .bool_gset:N = \g__xeCJK_indent_bool ,
3223   normalindentfirst .meta:n = { indentfirst = false }
3224 }
```

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

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

```
3225 \keys_define:nn { xeCJK / options }
3226 {
3227   quiet .code:n =
3228     {
3229       \msg_redirect_module:nnn { xeCJK } { warning } { info }
3230       \msg_redirect_module:nnn { xeCJK } { info } { none }
3231       \xeCJK_if_package_loaded:nF { fontspec }
3232       { \PassOptionsToPackage { quiet } { fontspec } }
3233     } ,
3234   silent .code:n =
3235     {
3236       \msg_redirect_module:nnn { xeCJK } { warning } { none }
3237       \msg_redirect_module:nnn { xeCJK } { info } { none }
3238       \xeCJK_if_package_loaded:nF { fontspec }
3239       { \PassOptionsToPackage { silent } { fontspec } }
3240     } ,
3241   unknown .code:n =
3242     {
3243       \xeCJK_if_package_loaded:nTF { fontspec }
3244       { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
3245       { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
3246     }
3247 }
3248 \__xeCJK_msg_new:nn { key-unknown }
3249 {
3250   Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~`#1'.\\\\
3251   The~key~`#1'~is~being~ignored.
3252 }
```

(End definition for *quiet* and *silent*.)

5.17 xeCJK 初始化设置

```
\CJKsymbol
\CJKpunctsymbol 3253 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
3254 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}
(End definition for \CJKsymbol and \CJKpunctsymbol.)
xeCJK 宏包的初始化设置。

3255 \keys_set:nn { xeCJK / options }
3256 {
3257   CJKglue      = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
3258   CJKecglue    = { ~ } ,
3259   xCJKecglue   = false ,
3260   CheckSingle  = false ,
3261   PlainEquation = false ,
3262   CheckFullRight = false ,
3263   CJKspace     = false ,
3264   CJKmath      = false ,
3265   CJKnumber    = false ,
3266   xeCJKactive  = true ,
3267   LocalConfig  = true ,
3268   indentfirst  = true ,
3269   Verb        = env ,
3270   EmboldenFactor = 4 ,
3271   SlantFactor   = 0.167 ,
3272   PunctStyle    = quanjiao ,
3273   NewLineCS     = { \par \[ } ,
3274   EnvCS        = { \begin \end } ,
3275   NoBreakCS     = { \footnote \footnotemark \nobreak } ,
3276   KaiMingPunct  = { ~~~~3002 ~~~~ff0e ~~~~ff1f ~~~~ff01 } ,
3277   LongPunct     = { ~~~~2014 ~~~~2015 ~~~~2500 ~~~~2025 ~~~~2026 } ,
3278   MiddlePunct   = { ~~~~2014 ~~~~2015 ~~~~2027 ~~~~2500 ~~~~00b7 ~~~~30fb ~~~~ff65 } ,
3279   AllowBreakBetweenPuncts = false
3280 }
3281 \defaultCJKfontfeatures { Script = CJK }

执行宏包选项,并载入 fontspec 宏包和 xunicode-addon。

3282 \ProcessKeysOptions { xeCJK / options }
3283 \RequirePackage { fontspec } [ 2012/05/01 ]
3284 \RequirePackage { xunicode-addon }

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

3285 \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。

3286 \bool_if:NT \g__xeCJK_indent_bool { \RequirePackage { indentfirst } }

对不能通过 \xeCJKsetup 设置的选项给出警告。

3287 \keys_define:nn { xeCJK / options }
3288 {
3289   LocalConfig .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
3290   CJKnumber .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
3291   indentfirst .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } }
3292 }
3293 \__xeCJK_msg_new:nn { option-invalid }
3294 {
3295   The~#1'~option~only~can~be~set~in~the~optional~argument~to~the\\
3296   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\\\
3297   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
3298 }

\CJKrmdefault
\CJKsfdefault 3299 \tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
\CJKttdefault 3300 \tl_if_exist:NF \CJKsfdefault { \tl_gset:Nn \CJKsfdefault { sf } }
\CJKfamilydefault 3301 \tl_if_exist:NF \CJKttdefault { \tl_gset:Nn \CJKttdefault { tt } }
3302 \tl_new:N \l__xeCJK_family_default_init_tl
3303 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
3304 \tl_set:Nx \l__xeCJK_family_default_init_tl
```

```

3305 {
3306   \exp_not:N \__xeCJK_family_default_wrap:n
3307   {
3308     \tl_if_exist:NTF \CJKfamilydefault
3309     { \exp_not:N \CJKfamilydefault }
3310     { \exp_not:N \CJKrmdefault }
3311   }
3312 }
3313 \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。

```

3314 \NewDocumentCommand \xeCJKsetup { +m }
3315 {
3316   \keys_set:nn { xeCJK / options } {#1}
3317   \tex_ignorespaces:D
3318 }

```

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

\xeCJKsetemboldenfactor

```

\xeCJKsetslantfactor
3319 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
3320 { \xeCJKsetup { EmboldenFactor = {#1} } }
3321 \NewDocumentCommand \xeCJKsetslantfactor { m }
3322 { \xeCJKsetup { SlantFactor = {#1} } }

```

(End definition for `\xeCJKsetemboldenfactor` and `\xeCJKsetslantfactor`.)

\punctstyle

```

\xeCJKplainchr
3323 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
3324 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }

```

(End definition for `\punctstyle` and `\xeCJKplainchr`.)

\CJKsetecglue

```

3325 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
3326 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue

```

(End definition for `\CJKsetecglue`.)

\CJKspace

```

\CJKnospace
3327 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
3328 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }

```

(End definition for `\CJKspace` and `\CJKnospace`.)

\xeCJKallowbreakbetweenpuncts

```

\xeCJKnobreakbetweenpuncts
3329 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
3330 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
3331 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
3332 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }

```

(End definition for `\xeCJKallowbreakbetweenpuncts` and `\xeCJKnobreakbetweenpuncts`.)

\xeCJKenablefallback

```

\xeCJKdisablefallback
3333 \NewDocumentCommand \xeCJKenablefallback { }
3334 { \xeCJKsetup { AutoFallBack = true } }
3335 \NewDocumentCommand \xeCJKdisablefallback { }
3336 { \xeCJKsetup { AutoFallBack = false } }

```

(End definition for `\xeCJKenablefallback` and `\xeCJKdisablefallback`.)

\xeCJKsetcharclass

```

3337 \NewDocumentCommand \xeCJKsetcharclass { m m m }
3338 {
3339   \xeCJK_set_char_class:nnn {#1} {#2} {#3}
3340   \xeCJKResetPunctClass
3341 }

```

(End definition for `\xeCJKsetcharclass`.)

5.18 兼容性修补

`_xeCJK_update_url_font:` 使通过 `\urlstyle` 或者 `\UrlFont` 设置的路径中使用的 CJK 字体生效。

```
\Url@MathSetup 3342 \cs_new_protected_nopar:Npn \_xeCJK_update_url_font:
3343 {
3344   \group_begin: \xeCJK_select_font: \exp_after:wN \group_end:
3345   \exp_after:wN \tex_textfont:D \exp_after:wN \c_xeCJK_math_fam_int
3346   \tex_the:D \tex_font:D
3347 }
3348 \_xeCJK_after_end_preamble:n
3349 {
3350   \bool_if:nT { \g_xeCJK_math_bool && \cs_if_exist_p:N \Url@MathSetup }
3351   { \tl_put_right:Nn \Url@MathSetup { \_xeCJK_update_url_font: } }
3352 }
(End definition for \_xeCJK_update_url_font: and \Url@MathSetup.)
```

`\fontspec_setup_maths:` 如果没有设置 `\setboldmathrm`, 即 `\g_fontspec_bfmathrm_tl` 为空, 那么 `\mathrm` 的字体实际与 `operators` 字体族完全一致。这时候应该通过 `\DeclareSymbolFontAlphabet` 来定义 `\mathrm`, 避免使用它的时候再声明一个重复的数学字体族。

```
3353 \cs_if_free:NF \fontspec_setup_maths:
3354 {
3355   \cs_gset_protected_nopar:Npx \fontspec_setup_maths:
3356   {
3357     \exp_not:o
3358     {
3359       \fontspec_setup_maths:
3360       \tl_if_empty:NT \g_fontspec_bfmathrm_tl
3361       { \DeclareSymbolFontAlphabet \mathrm { operators } }
3362     }
3363   }
3364 }
(End definition for \fontspec_setup_maths: and \mathrm.)
```

`\(` 的在 $\text{\LaTeX}2_{\epsilon}$ 中的定义是

```
\)
\math \def\(\{\relax\ifmmode\@badmath\else$\fi}
```

`\endmath` 这个定义最开始的 `\relax` 是为了防止 `\(` 出现在表格单元格的开始位置时, 模式判断不正确 (因为 \TeX 会先看单元格中第一个不可展的非空格记号是否是 `\omit` 或 `\noalign`)。但是它会

`\ensuremath` 造成一个边界, 使 \xeCJK 不能看到 `\relax` 后面出现的 `$`, 从而不能加入间距¹⁰。使用 $\epsilon\text{-}\text{\TeX}$ 的 `\protected` 来定义它, 可以不需要 `\relax`, 或者将 `\relax` 改成 `\scan_align_safe_stop:`, 都可以避免这些情况。同时 `fixltx2e` 中还使用了 `\MakeRobust\()`, 我们需要小心处理。另外 `ulem` 也定义了一个 `\MakeRobust`, 如果它被放在 `fixltx2e` 之前载入, 那么 `fixltx2e` 的定义就会失效 (因为 `fixltx2e` 使用 `\providecommand*` 来定义 `\MakeRobust`)。但是 `ulem` 的定义并不完全正确, 没有考虑 \TeX 不会略去控制符号后面的空格的情况。

`_xeCJK_math_robust:N`

```
3365 \cs_new_protected_nopar:Npn \_xeCJK_math_robust:N #1
3366 { \exp_args:Nnc \_xeCJK_math_robust_aux:NN #1 { \cs_to_str:N #1 ~ } }
3367 \cs_new_protected_nopar:Npn \_xeCJK_math_robust_aux:NN #1#2
3368 {
3369   \exp_args:Nx \str_case:nnTF { \token_get_replacement_spec:N #1 }
3370   {
3371     { \x@protect #1 \protect #2 } { }
3372     { \protect #2 } { }
3373   }
3374   { \_xeCJK_math_robust:NN #1#2 }
3375   { \_xeCJK_math_robust:NN #1#1 }
3376 }
3377 \cs_new_protected_nopar:Npn \_xeCJK_math_robust:NN #1#2
3378 {
3379   \str_if_eq:x:nnTF { \token_get_arg_spec:N #2 } { } {
3380     {
3381       \exp_args:No \tl_if_head_eq_meaning:nNTF {#2} \scan_stop:
3382       {
3383         \cs_gset_protected_nopar:Npx #1
```

¹⁰<http://tex.stackexchange.com/q/124773>

```

3384         { \scan_align_safe_stop: \tl_tail:N #2 }
3385     }
3386     {
3387         \cs_if_eq:NNTF #1 \ensuremath
3388         {
3389             \cs_gset_protected_nopar:Npx #1
3390             { \scan_align_safe_stop: \exp_not:o {#2} }
3391         }
3392         {
3393             \__xeCJK_warning:nxx { robust-failure }
3394             { \token_to_str:N #1 } { \token_to_meaning:N #2 }
3395         }
3396     }
3397 }
3398 {
3399     \__xeCJK_warning:nxx { robust-failure }
3400     { \token_to_str:N #1 } { \token_to_meaning:N #2 }
3401 }
3402 }
3403 \__xeCJK_msg_new:nnn { robust-failure }
3404 { xeCJK~can~not~make~`#1'~robust. }
3405 {
3406     The~current~meaning~of~`#1'~is:\\
3407     \iow_indent:n {#2}
3408 }
3409 \__xeCJK_math_robust:N \ (
3410 \__xeCJK_math_robust:N \ )
3411 \__xeCJK_math_robust:N \math
3412 \__xeCJK_math_robust:N \endmath
3413 \__xeCJK_math_robust:N \ensuremath
(End definition for \ ( and others.)

```

\[当 amsmath 没有在 amsthm 之前被调用时, amsthm 会展开 \[, 并用 \$ 作为参数定界记号, 相关代码为

```

\def\@tempa#1$#2#3\@nil{%
\def\[{#1$#2\def\@currenvir{displaymath}#3}%
}%
\expandafter\@tempa\[\@nil

```

而 fixltx2e 中使用了 \MakeRobust\[, 使得将 \[展开一次的内容中并不直接含有 \$, 从而造成了 Runaway argument? 的错误。可以在 amsthm 之前引入 amsmath, 避免出现这个错误。我们下面用 ϵ -TeX 的 \protected 来定义它。当然, 如果之后只使用 amsthm, 那么 \[会被修改, 将不再是“健壮”的了。这也是上面 __xeCJK_math_robust:NN 中还使用 \scan_align_safe_stop: 的原因。

```

3414 \bool_if:nF
3415 {
3416     \xeCJK_if_package_loaded_p:n { amsmath } ||
3417     \xeCJK_if_package_loaded_p:n { amsthm }
3418 }
3419 {
3420     \__xeCJK_math_robust:N \[
3421     \__xeCJK_math_robust:N \]
3422 }
(End definition for \[ and \].)

```

\nobreakspace 空格在 TeX 中是特殊的记号, 似乎不应该把它定义为字体中的符号(U+00A0)。

```

3423 \UndeclareTextCommand \nobreakspace { \UTFencname }
3424 \RenewDocumentCommand \nobreakspace {} { \leavevmode \nobreak \ }
(End definition for \nobreakspace.)
    当符号命令紧跟在 CJK 字符类后面时, 强制发生状态转移, 使字体回到西文状态。
3425 \AtBeginUTFCommand { \bool_if:NT \l__xeCJK_CJK_group_bool { \scan_stop: } }

```

比较老版本的 `realscripts` 定义了 `\dim_max:nn` 和 `\dim_min:nn`, 这与新版本的 `expl3` 冲突。

```

3426 \__xeCJK_msg_new:nn { conflict-package }
3427 {
3428   The~`#1'~package~is~too~old.  \
3429   Please~update~an~up~to~date~version~of~it\
3430   using~your~TeX~package~manager~or~from~CTAN.
3431 }
3432 \xeCJK_if_package_loaded:nTF { realscripts }
3433 {
3434   \ifpackagelater { realscripts } { 2010/10/10 } { }
3435   {
3436     \__xeCJK_error:nx { conflict-package }
3437     {
3438       \xeCJK_if_package_loaded:nTF { xltextra }
3439       { xltextra } { realscripts }
3440     }
3441   }
3442 }
3443 {
3444   \cs_new_eq:NN \__xeCJK_dim_max:nn \dim_max:nn
3445   \cs_new_eq:NN \__xeCJK_dim_min:nn \dim_min:nn
3446   \__xeCJK_at_end_preamble:n
3447   {
3448     \xeCJK_if_package_loaded:nT { realscripts }
3449     {
3450       \ifpackagelater { realscripts } { 2010/10/10 } { }
3451       {
3452         \cs_gset_eq:NN \dim_max:nn \__xeCJK_dim_max:nn
3453         \cs_gset_eq:NN \dim_min:nn \__xeCJK_dim_min:nn
3454       }
3455     }
3456     \cs_undefine:N \__xeCJK_dim_max:nn
3457     \cs_undefine:N \__xeCJK_dim_min:nn
3458   }
3459 }

```

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

```

3460 \RenewDocumentCommand \fontfamily { m }
3461 {
3462   \tl_set:Nx \f@family {#1}
3463   \__xeCJK_update_family:nn {#1}
3464   {
3465     { \rmdefault } { \xeCJK_switch_family:n { \CJKrmdefault } }
3466     { \sfdefault } { \xeCJK_switch_family:n { \CJKsfdefault } }
3467     { \ttdefault } { \xeCJK_switch_family:n { \CJKttdefault } }
3468     { \familydefault } { \xeCJK_switch_family:n { \CJKfamilydefault } }
3469   }
3470 }
3471 \cs_new_eq:NN \__xeCJK_update_family:nn \str_case:nn
(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₁`\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。

```

3472 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
3473 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
3474 \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。

```

3475 \cs_new_protected_nopar:Npn \xeCJK@italiccorr
3476 {
3477   \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero

```

```

3478 {
3479   \xeCJK_if_last_node:nTF { default }
3480   {
3481     \xeCJK_remove_node: \@@italiccorr
3482     { \xeCJK_make_node:n { default } }
3483   }
3484   {
3485     \xeCJK_if_last_node:nTF { CJK }
3486     {
3487       \xeCJK_remove_node: \@@italiccorr
3488       { \xeCJK_make_node:n { CJK } } \use:n
3489     }
3490     {
3491       \xeCJK_if_last_node:nTF { CJK-space }
3492       {
3493         \xeCJK_remove_node: \@@italiccorr
3494         { \xeCJK_make_node:n { CJK-space } } \use:n
3495       }
3496       { \@@italiccorr \use_none:n }
3497     }
3498   }

```

\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:)。

```

3498 {
3499   \exp_after:wN \exp_after:wN \exp_after:wN
3500   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3501   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3502   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
3503   \xeCJK_ignore_spaces:w
3504 }
3505 }
3506 { \@@italiccorr }
3507 }
3508 }

```

(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。

```

3509 \_xeCJK_after_end_preamble:n
3510 {
3511   \int_compare:nNnF
3512   { \c_three + \seq_count:N \g__xeCJK_new_class_seq } = \xe@alloc@intercharclass
3513   {
3514     \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
3515     {
3516       \seq_if_in:NnF \g__xeCJK_new_class_seq {##1}
3517       { \_xeCJK_set_others_toks:n {##1} }
3518     }
3519   }
3520 }
3521 \cs_new_protected_nopar:Npn \_xeCJK_set_others_toks:n #1
3522 {
3523   \int_set:cn { \_xeCJK_class_csname:n { Others } } {##1}
3524   \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
3525   {
3526     \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
3527     \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
3528     \xeCJK_app_inter_class_toks:nnx {##1} { Others }
3529     { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
3530     \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
3531     { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
3532     \xeCJK_if_blank_x:nT
3533     { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
3534     {
3535       \xeCJK_copy_inter_class_toks:nnnn
3536       { Others } { Boundary } { Default } { Boundary }

```

```

3537     }
3538     \xeCJK_if_blank_x:nT
3539     { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
3540     {
3541         \xeCJK_copy_inter_class_toks:nnnn
3542         { Boundary } { Others } { Boundary } { Default }
3543     }
3544 }
3545 }
(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: 3546 \cs_new_eq:NN \_xeCJK_group_begin: \group_begin:
3547 \cs_new_eq:NN \_xeCJK_group_end: \group_end:
(End definition for \_xeCJK_group_begin: and \_xeCJK_group_end:.)

```

\textellipsis 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。

```

3548 \tl_map_inline:nn
3549 {
3550     \textellipsis \textemdash \textperiodcentered \textcentereddot
3551     \textquoteleft \textquoteright \textquotedblleft \textquotedblright
3552 }
3553 {
3554     \AtBeginUTFCommand [#1] { \_xeCJK_group_begin: \makeCJKinactive }
3555     \AtEndUTFCommand [#1] { \_xeCJK_group_end: }
3556 }
(End definition for \textellipsis.)

```

常被用作中文间隔号的 U+00B7 与 T1 等旧字体编码下定义的符号命令冲突。

```

3557 \_xeCJK_after_preamble:n
3558 {
3559     \tl_put_left:Nn \tipaencoding { \makeCJKinactive }
3560     \cs_new_eq:NN \_xeCJK_aux_r:n \r
3561     \cs_set_nopar:Npn \r #1
3562     {
3563         \bool_if:nTF
3564         {
3565             \str_if_eq_x_p:nn { \f@encoding } { T1 } &&
3566             \str_if_eq_x_p:nn {#1} { u }
3567         }
3568         { { \makeCJKinactive \_xeCJK_aux_r:n {#1} } }
3569         { \_xeCJK_aux_r:n {#1} }
3570     }
3571     \xeCJK_if_package_loaded:nT { pifont }
3572     {
3573         \RenewDocumentCommand \Pifont { m }
3574         { \makeCJKinactive \usefont { U } {#1} { m } { n } }
3575     }
3576 }

```

简单处理与 hyperref 宏包的兼容问题。

```

3577 \_xeCJK_after_end_preamble:n
3578 {
3579     \xeCJK_if_package_loaded:nT { hyperref }
3580     {
3581         \pdfstringdefDisableCommands
3582         {
3583             \_xeCJK_gobble_CJKfamily:
3584             \xeCJK_cs_clear:N \makeCJKinactive
3585             \xeCJK_cs_clear:N \_xeCJK_group_begin:
3586             \xeCJK_cs_clear:N \_xeCJK_group_end:
3587         }
3588     }
3589 }

```

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

```

3590 \__xeCJK_after_end_preamble:n
3591 {
3592   \bool_if:nT
3593     { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
3594     { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
3595 }

```

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

```

3596 \cs_if_exist:NF \CJK@ifundefined
3597   { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
3598 \NewDocumentCommand \xeCJKcaption { o m }
3599 {
3600   \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
3601   \use:x
3602   {
3603     \char_set_catcode_letter:n { 64 }
3604     \file_input:n { #2.cpx }
3605     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
3606   }
3607   \XeTeXdefaultencoding "UTF-8"
3608 }

```

(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。

```

3609 \cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
3610   { \AtEndOfPackage { \RequirePackage { xeCJKfntef } } }
3611   {
3612     \__xeCJK_at_end_preamble:n
3613     {
3614       \xeCJK_if_package_loaded:nF { xeCJKfntef }
3615       {
3616         \xeCJK_if_package_loaded:nTF { CJKfntef }
3617           { \RequirePackage { xeCJKfntef } }
3618           {
3619             \xeCJK_if_package_loaded:nT { ulem }
3620             { \RequirePackage { xeCJKfntef } }
3621           }
3622       }
3623     }
3624   }

```

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

```

3625 \__xeCJK_at_end_preamble:n
3626 {
3627   \xeCJK_if_package_loaded:nT { listings }
3628   { \RequirePackage { xeCJK-listings } }
3629 }

```

`\CJKaddEncHook` 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。

```

3630 \cs_new_protected:Npn \CJKaddEncHook #1#2
3631 {
3632   \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
3633   {
3634     \group_begin:
3635     \cs_set_nopar:Npn \Unicode ##1##2
3636       { (##1) * \c_two_hundred_fifty_six + (##2) }
3637     \cs_set_eq:NN \def \xeCJK_char_from_charcode:Nn
3638       #2
3639     \group_end:
3640     \tl_gset:Nn \CJK@tenthousand { ~~~~4e07 }
3641     \tl_gset:Nn \CJK@hundredmillion { ~~~~4ebf }
3642   }
3643 }
3644 \cs_new_protected_nopar:Npn \xeCJK_char_from_charcode:Nn #1#2

```

```

3645 {
3646   \group_begin:
3647   \char_set_lccode:nn { "4E00 } {#2}
3648   \tl_to_lowercase:n
3649   {
3650     \group_end:
3651     \tl_const:Nn #1 { ~~~~4e00 }
3652   }
3653 }
3654 \bool_if:NT \g__xeCJK_number_bool { \RequirePackage { CJKnumb } }
(End definition for \CJKaddEncHook.)
    最后引入本地配置文件。
3655 \bool_if:NT \g__xeCJK_config_bool
3656 {
3657   \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
3658   \@onefilewithoptions
3659   { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
3660 }
3661 \</package>

```

5.19 xeCJKfntef

3662 `<*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。

```

3663 \PassOptionsToPackage { normalem } { ulem }
3664 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
3665 \ProcessOptions \scan_stop:
3666 \RequirePackage { xeCJK }
3667 \RequirePackage { ulem }
3668 \RequirePackage { CJKfntef }
3669 \RequirePackage { environ }
3670 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

```

\xeCJK_hook_for_ulem:

```

3671 \cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
3672 {
3673   \bool_if:NF \l__xeCJK_ulem_hook_used_bool
3674   {
3675     \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
3676     \xeCJKsetup { CheckFullRight = false , xCJKecglue = false }
3677     \bool_if:NTF \l__xeCJK_ulem_skip_punct_bool
3678     { \cs_set_eq:NN \__xeCJK_ulem_leader_type: \UL@leadtype }
3679     {
3680       \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
3681       \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
3682     }
3683     \__xeCJK_ulem_initial:
3684     \xeCJK_glue_to_skip:nN
3685     {
3686       \cs_set_eq:NN \ \tex_space:D
3687       \cs_set_eq:NN \penalty \tex_penalty:D
3688       \cs_set_eq:NN \hskip \skip_horizontal:N
3689       \CJKglue
3690     } \l__xeCJK_ccglue_skip
3691     \xeCJK_glue_to_skip:nN
3692     {
3693       \cs_set_eq:NN \ \tex_space:D
3694       \cs_set_eq:NN \penalty \tex_penalty:D
3695       \cs_set_eq:NN \hskip \skip_horizontal:N
3696       \CJKecglue
3697     } \l__xeCJK_ecglue_skip
3698     \cs_set_protected_nopar:Npn \CJKglue
3699     { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
3700     \cs_set_protected_nopar:Npn \CJKecglue
3701     { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
3702   }

```

```

3703 }
3704 \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

```

3705 \cs_set_eq:NN \CJK@UL \CJK@@UL
3706 \tl_replace_once:Nnn \CJK@UL { \ULon }
3707 { \bool_set_true:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3708 \tl_replace_once:Nnn \CJK@@UL { \ULon }
3709 { \bool_set_false:N \l__xeCJK_ulem_skip_punct_bool \ULon }
3710 \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:

```

3711 \cs_new_protected_nopar:Npn \_xeCJK_ulem_skip_punct_begin:
3712 { \xeCJK_cs_clear:N \UL@leadtype }
3713 \cs_new_protected_nopar:Npn \_xeCJK_ulem_skip_punct_end:
3714 { \cs_set_eq:NN \UL@leadtype \_xeCJK_ulem_leader_type: }
3715 \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。

```

3716 \cs_new_protected_nopar:Npn \_xeCJK_ulem_initial:
3717 {
3718   \_xeCJK_ulem_swap_cs:NN
3719   \xeCJK_FullLeft_and_Default: \_xeCJK_ulem_FullLeft_and_Default:
3720   \xeCJK_FullLeft_and_CJK: \_xeCJK_ulem_FullLeft_and_CJK:
3721   \xeCJK_FullRight_and_Default: \_xeCJK_ulem_FullRight_and_Default:
3722   \xeCJK_FullRight_and_CJK: \_xeCJK_ulem_FullRight_and_CJK:
3723   \xeCJK_CJK_and_CJK:N \_xeCJK_ulem_CJK_and_CJK:N
3724   \xeCJK_Boundary_and_Default: \_xeCJK_ulem_Boundary_and_Default:
3725   \xeCJK_Boundary_and_NormalSp: \_xeCJK_ulem_Boundary_and_NormalSp:
3726   \xeCJK@fix@penalty \_xeCJK_ulem_fix_penalty:
3727   \_xeCJK_punct_kern:n \_xeCJK_ulem_punct_kern:n
3728   \_xeCJK_punct_hskip:n \_xeCJK_ulem_punct_hskip:n
3729   \_xeCJK_punct_breakable_kern:n \_xeCJK_ulem_punct_breakable_kern:n
3730   \_xeCJK_CJK_and_Boundary_aux: \_xeCJK_ulem_CJK_and_Boundary_aux:
3731   \_xeCJK_Default_and_FullLeft_glue:N \_xeCJK_ulem_Default_and_FullLeft_glue:N
3732   \_xeCJK_Default_and_FullRight_glue:N \_xeCJK_ulem_Default_and_FullRight_glue:N
3733   \_xeCJK_CJK_and_FullLeft_glue:N \_xeCJK_ulem_CJK_and_FullLeft_glue:N
3734   \_xeCJK_CJK_and_FullRight_glue:N \_xeCJK_ulem_CJK_and_FullRight_glue:N
3735   \_xeCJK_Boundary_and_FullLeft_glue:N \_xeCJK_ulem_Boundary_and_FullLeft_glue:N
3736   \q_recursion_tail \q_nil \q_recursion_stop
3737   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3738   {
3739     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3740     {
3741       \str_if_eq:nnTF {##1} {####1}
3742       {
3743         \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
3744         { \_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3745         \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
3746         { \_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3747       }
3748       {
3749         \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
3750         { \_xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
3751       }
3752     }
3753   }
3754 }
3755 \cs_new_protected_nopar:Npn \_xeCJK_ulem_swap_cs:NN #1#2
3756 {
3757   \quark_if_recursion_tail_stop:N #1
3758   \xeCJK_swap_cs:NN #1#2
3759   \_xeCJK_ulem_swap_cs:NN
3760 }

```


(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。

```
3761 \cs_new_nopar:Npn \xeCJK_if_ulem_patch:TF
3762 {
3763   \if_meaning:w \_ \LA@space
3764   \exp_after:wN \use_ii:nn
3765   \else:
3766     \exp_after:wN \use_i:nn
3767   \fi:
3768 }
```

(End definition for `\xeCJK_if_ulem_patch:TF`.)

`_xeCJK_ulem_Boundary_and_Default:`

```
3769 \cs_new_protected_nopar:Npn \_xeCJK_ulem_Boundary_and_Default:
3770 {
3771   \xeCJK_if_ulem_patch:TF
3772   {
3773     \xeCJK_if_last_node:nTF { CJK }
3774     { \xeCJK_remove_node: \skip_horizontal:N \l_xeCJK_ecglue_skip }
3775     { \xeCJK_if_last_node:nT { CJK-space } { \xeCJK_remove_node: \c_space_tl } }
3776   }
3777   { \_xeCJK_ulem_Boundary_and_Default: }
3778 }
```

(End definition for `_xeCJK_ulem_Boundary_and_Default:`.)

`_xeCJK_ulem_Boundary_and_NormalSp:`

```
3779 \cs_new_protected_nopar:Npn \_xeCJK_ulem_Boundary_and_NormalSp:
3780 {
3781   \xeCJK_if_ulem_patch:TF
3782   { \xeCJK_if_last_node:nT { CJK-space } { \xeCJK_remove_node: \c_space_tl } }
3783   { \_xeCJK_ulem_Boundary_and_NormalSp: }
3784 }
```

(End definition for `_xeCJK_ulem_Boundary_and_NormalSp:`.)

`_xeCJK_ulem_CJK_and_Boundary_aux:`

```
3785 \cs_new_protected_nopar:Npn \_xeCJK_ulem_CJK_and_Boundary_aux:
3786 {
3787   \xeCJK_if_ulem_patch:TF
3788   {
3789     \xeCJK_class_group_end:
3790     \UL@stop \_xeCJK_ulem_hskip:n { \c_zero_skip } \UL@start
3791     { \xeCJK_make_node:n { CJK } }
3792   }
3793   { \_xeCJK_ulem_CJK_and_Boundary_aux: }
3794 }
```

(End definition for `_xeCJK_ulem_CJK_and_Boundary_aux:`.)

`_xeCJK_ulem_fix_penalty:`

```
3795 \cs_new_protected_nopar:Npn \_xeCJK_ulem_fix_penalty:
3796 {
3797   \xeCJK_if_ulem_patch:TF
3798   { \fix@penalty }
3799   { \_xeCJK_ulem_fix_penalty: }
3800 }
```

(End definition for `_xeCJK_ulem_fix_penalty:`.)

`_xeCJK_ulem_CJK_and_CJK:N`

```
3801 \cs_new_protected_nopar:Npn \_xeCJK_ulem_CJK_and_CJK:N
3802 {
3803   \xeCJK_if_ulem_patch:TF
3804   {
3805     \xeCJK_class_group_end:
3806     \UL@stop \_xeCJK_ulem_ccglue: \UL@start
3807     \_xeCJK_ulem_class_group_begin:
```

```

3808         \CJKsymbol
3809     }
3810     { \_xeCJK_ulem_CJK_and_CJK:N }
3811 }

```

(End definition for _xeCJK_ulem_CJK_and_CJK:N.)

_xeCJK_ulem_class_group_begin:

```

3812 \cs_new_protected_nopar:Npn \_xeCJK_ulem_class_group_begin:
3813 {
3814     \xeCJK_class_group_begin:
3815     \xeCJK_clear_Boundary_and_CJK_toks:
3816     \xeCJK_select_font:
3817 }

```

(End definition for _xeCJK_ulem_class_group_begin:.)

_xeCJK_ulem_between_CJK_blocks:nnN

```

3818 \cs_new_protected_nopar:Npn \_xeCJK_ulem_between_CJK_blocks:nnN #1#2
3819 {
3820     \xeCJK_if_ulem_patch:TF
3821     {
3822         \xeCJK_class_group_end:
3823         \UL@stop \_xeCJK_ulem_ccglue: \UL@start
3824         \xeCJK_class_group_begin:
3825         \xeCJK_clear_Boundary_and_CJK_toks:
3826         \_xeCJK_switch_font:nn {#1} {#2}
3827         \CJKsymbol
3828     }
3829     {
3830         \skip_horizontal:N \l_xeCJK_ccglue_skip
3831         \_xeCJK_switch_font:nn {#1} {#2}
3832         \CJKsymbol
3833     }
3834 }

```

(End definition for _xeCJK_ulem_between_CJK_blocks:nnN.)

_xeCJK_ulem_Default_and_FullLeft_glue:N

```

3835 \cs_new_protected_nopar:Npn \_xeCJK_ulem_Default_and_FullLeft_glue:N #1
3836 {
3837     \xeCJK_if_ulem_patch:TF
3838     {
3839         \UL@stop
3840         \_xeCJK_ulem_skip_punct_begin:
3841         \_xeCJK_punct_glue:NN \c_xeCJK_left_tl {#1}
3842         \_xeCJK_punct_offset:NN \c_xeCJK_left_tl {#1}
3843         \UL@start
3844     }
3845     { \_xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
3846 }

```

(End definition for _xeCJK_ulem_Default_and_FullLeft_glue:N.)

_xeCJK_ulem_Boundary_and_FullLeft_glue:N

```

3847 \cs_new_protected_nopar:Npn \_xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
3848 {
3849     \xeCJK_if_ulem_patch:TF
3850     {
3851         \UL@stop
3852         \_xeCJK_ulem_skip_punct_begin:
3853         \int_compare:nNnTF \etex_lastnodetype:D = \c_one
3854         { \_xeCJK_zero_glue: }
3855         { \_xeCJK_punct_glue:NN \c_xeCJK_left_tl {#1} }
3856         \UL@start
3857     }
3858     { \_xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
3859 }

```

(End definition for _xeCJK_ulem_Boundary_and_FullLeft_glue:N.)

_xeCJK_ulem_CJK_and_FullLeft_glue:N

```
3860 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
3861 {
3862   \xeCJK_if_ulem_patch:TF
3863   {
3864     \xeCJK_class_group_end:
3865     \UL@stop
3866     \__xeCJK_ulem_skip_punct_begin:
3867     \__xeCJK_ulem_ccglue:
3868     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3869     \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
3870     \UL@start
3871     \__xeCJK_ulem_class_group_begin:
3872   }
3873   { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
3874 }
(End definition for \__xeCJK_ulem_CJK_and_FullLeft_glue:N.)
```

_xeCJK_ulem_Default_and_FullRight_glue:N

```
3875 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
3876 {
3877   \xeCJK_if_ulem_patch:TF
3878   {
3879     \UL@stop
3880     \__xeCJK_ulem_skip_punct_begin:
3881     \__xeCJK_punct_if_long:NTF {#1}
3882     { \__xeCJK_ulem_ccglue: }
3883     {
3884       \__xeCJK_punct_if_middle:NTF {#1}
3885       {
3886         \xeCJK_no_break:
3887         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
3888         \__xeCJK_punct_bound_rule:NN \c__xeCJK_left_tl {#1}
3889       }
3890       { \xeCJK_no_break: }
3891     }
3892     \UL@start
3893   }
3894   { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
3895 }
(End definition for \__xeCJK_ulem_Default_and_FullRight_glue:N.)
```

_xeCJK_ulem_CJK_and_FullRight_glue:N

```
3896 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
3897 {
3898   \xeCJK_if_ulem_patch:TF
3899   {
3900     \xeCJK_class_group_end:
3901     \__xeCJK_Default_and_FullRight_glue:N {#1}
3902     \__xeCJK_ulem_class_group_begin:
3903   }
3904   { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
3905 }
(End definition for \__xeCJK_ulem_CJK_and_FullRight_glue:N.)
```

_xeCJK_ulem_FullLeft_and_Default:

```
3906 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_Default:
3907 {
3908   \xeCJK_if_ulem_patch:TF
3909   {
3910     \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3911     {
3912       \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3913       \__xeCJK_punct_bound_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3914       \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
3915       \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3916     }
3917   }
3918 }
```

```

3917         { \xeCJK_class_group_end: \UL@stop }
3918         \__xeCJK_ulem_skip_punct_end:
3919         \xeCJK_no_break:
3920         \UL@start
3921     }
3922     { \__xeCJK_ulem_FullLeft_and_Default: }
3923 }

```

(End definition for __xeCJK_ulem_FullLeft_and_Default:.)

_xeCJK_ulem_FullLeft_and_CJK:

```

3924 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_CJK:
3925 {
3926     \xeCJK_if_ulem_patch:TF
3927     {
3928         \xeCJK_FullLeft_and_Default:
3929         \__xeCJK_ulem_class_group_begin:
3930     }
3931     { \__xeCJK_ulem_FullLeft_and_CJK: }
3932 }

```

(End definition for __xeCJK_ulem_FullLeft_and_CJK:.)

_xeCJK_ulem_FullRight_and_Default:

```

3933 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_Default:
3934 {
3935     \xeCJK_if_ulem_patch:TF
3936     {
3937         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3938         \xeCJK_class_group_end:
3939         \UL@stop
3940         \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3941         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3942         \__xeCJK_ulem_skip_punct_end:
3943         \UL@start
3944     }
3945     { \__xeCJK_ulem_FullRight_and_Default: }
3946 }

```

(End definition for __xeCJK_ulem_FullRight_and_Default:.)

_xeCJK_ulem_FullRight_and_CJK:

```

3947 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_CJK:
3948 {
3949     \xeCJK_if_ulem_patch:TF
3950     {
3951         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3952         \xeCJK_class_group_end:
3953         \UL@stop
3954         \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3955         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3956         \__xeCJK_ulem_ccglue:
3957         \__xeCJK_ulem_skip_punct_end:
3958         \UL@start
3959         \__xeCJK_ulem_class_group_begin:
3960     }
3961     { \__xeCJK_ulem_FullRight_and_CJK: }
3962 }

```

(End definition for __xeCJK_ulem_FullRight_and_CJK:.)

_xeCJK_ulem_punct_hskip:n

```

3963 \cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_hskip:n
3964 {
3965     \xeCJK_if_ulem_patch:TF
3966     { \__xeCJK_ulem_hskip:n }
3967     { \__xeCJK_ulem_punct_hskip:n }
3968 }

```

(End definition for __xeCJK_ulem_punct_hskip:n.)

_xeCJK_ulem_punct_kern:n

```
3969 \cs_new_protected_nopar:Npn \_xeCJK_ulem_punct_kern:n
3970 {
3971   \xeCJK_if_ulem_patch:TF
3972   { \_xeCJK_ulem_hskip:n }
3973   { \_xeCJK_ulem_punct_kern:n }
3974 }
(End definition for \_xeCJK_ulem_punct_kern:n.)
```

_xeCJK_ulem_punct_breakable_kern:n

```
3975 \cs_new_protected_nopar:Npn \_xeCJK_ulem_punct_breakable_kern:n #1
3976 {
3977   \xeCJK_if_ulem_patch:TF
3978   {
3979     \xeCJK_class_group_end:
3980     \UL@stop \_xeCJK_ulem_hskip:n {#1} \UL@start
3981     \_xeCJK_ulem_class_group_begin:
3982   }
3983   { \_xeCJK_ulem_punct_breakable_kern:n {#1} }
3984 }
(End definition for \_xeCJK_ulem_punct_breakable_kern:n.)
```

_xeCJK_ulem_glue:n

在下划线状态下的分别代替 \CJKglue 等。

_xeCJK_ulem_ccglue:

```
3985 \cs_new_protected_nopar:Npn \_xeCJK_ulem_glue:n #1
```

_xeCJK_ulem_hskip:n

```
3986 {
3987   \xeCJK_if_ulem_patch:TF
3988   { \UL@stop \_xeCJK_ulem_hskip:n {#1} \UL@start }
3989   { \skip_horizontal:n {#1} }
3990 }
3991 \cs_new_protected_nopar:Npn \_xeCJK_ulem_ccglue:
3992 { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders }
3993 \cs_new_protected_nopar:Npn \_xeCJK_ulem_hskip:n #1
3994 {
3995   \int_compare:nNnTF \tex_lastkern:D = \c_three
3996   { \skip_horizontal:n {#1} }
3997   { \skip_set:Nn \UL@skip {#1} \UL@leaders }
3998 }
(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。因此需要我们在这一处修改它们。

```
3999 \RenewDocumentCommand \CJKunderdot { m }
4000 {
4001   \bool_if:NT \l__xeCJK_ulem_hook_used_bool
4002   { \UL@stop \_xeCJK_ulem_restore_CJK_and_Boundary: }
4003   \CJK@preUnderdot
4004   \_xeCJK_make_under_symbol:n { \CJK@underdotSkip }
4005   \cs_gset_eq:NN \_xeCJK_save_under_CJKsymbol:N \CJKsymbol
4006   \cs_set_eq:NN \CJKsymbol \_xeCJK_under_CJKsymbol:N
4007   \_xeCJK_restore_shipout_CJKsymbol:
4008   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
4009   #1
4010   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
4011   \cs_set_eq:NN \CJKsymbol \_xeCJK_save_under_CJKsymbol:N
4012   \tl_clear:N \l__xeCJK_fntef_shipout_tl
4013   \CJK@postUnderdot
4014   \bool_if:NT \l__xeCJK_ulem_hook_used_bool
4015   { \_xeCJK_ulem_restore_CJK_and_Boundary: \UL@start }
4016   \tex_ignorespaces:D
4017 }
4018 \box_new:N \g__xeCJK_under_symbol_box
4019 \cs_new_protected_nopar:Npn \_xeCJK_ulem_restore_CJK_and_Boundary:
4020 {
4021   \xeCJK_if_ulem_patch:TF
4022   {
4023     \xeCJK_swap_cs:NN
4024     \_xeCJK_CJK_and_Boundary_aux: \_xeCJK_ulem_CJK_and_Boundary_aux:
```

```

4025     }
4026     { }
4027 }
(End definition for \CJKunderdot.)

```

\CJKunderanysymbol

```

4028 \RenewDocumentCommand \CJKunderanysymbol { m m m }
4029 {
4030   \group_begin:
4031   \hbox_set:Nn \CJK@underdotBox {#2}
4032   \__xeCJK_make_under_symbol:n {#1}
4033   \cs_set_eq:NN \__xeCJK_save_under_CJKsymbol:N \CJKsymbol
4034   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
4035   \__xeCJK_restore_shipout_CJKsymbol:
4036   \tl_clear:N \l__xeCJK_underdot_shipout_tl
4037   #3
4038   \group_end:
4039   \tex_ignorespaces:D
4040 }
(End definition for \CJKunderanysymbol.)

```

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

```

4041 \cs_new_protected:Npn \__xeCJK_restore_shipout_CJKsymbol:
4042 {
4043   \tl_set:Nn \l__xeCJK_fntef_shipout_tl
4044   {
4045     \l__xeCJK_underdot_shipout_tl
4046     \cs_set_eq:NN \CJKsymbol \__xeCJK_save_under_CJKsymbol:N
4047   }
4048 }
4049 \tl_new:N \l__xeCJK_fntef_shipout_tl
4050 \tl_new:N \l__xeCJK_underdot_shipout_tl
4051 \tl_set:Nn \l__xeCJK_underdot_shipout_tl
4052 {
4053   \cs_set_eq:NN \CJKf@global \scan_stop:
4054   \int_zero:N \CJK@nest \CJK@postUnderdot
4055 }
4056 \__xeCJK_add_to_shipout:n { \l__xeCJK_fntef_shipout_tl }
(End definition for \__xeCJK_restore_shipout_CJKsymbol:.)

```

__xeCJK_make_under_symbol:n

```

4057 \cs_new_protected:Npn \__xeCJK_make_under_symbol:n #1
4058 {
4059   \hbox_set:Nn \l__xeCJK_tmp_box { ^^^~4e00 }
4060   \vbox_gset_to_ht:Nnn \g__xeCJK_under_symbol_box \c_zero_dim
4061   {
4062     \skip_vertical:n {#1}
4063     \hbox_to_zero:n
4064     {
4065       \tex_kern:D \box_wd:N \l__xeCJK_tmp_box
4066       \tex_hss:D \box_use:N \CJK@underdotBox \tex_hss:D
4067     }
4068     \tex_vss:D
4069   }
4070 }
(End definition for \__xeCJK_make_under_symbol:n.)

```

__xeCJK_under_CJKsymbol:N

```

4071 \cs_new_protected_nopar:Npn \__xeCJK_under_CJKsymbol:N #1
4072 {
4073   \hbox_overlap_right:n { \box_use:N \g__xeCJK_under_symbol_box }
4074   \xeCJK_no_break: \__xeCJK_save_under_CJKsymbol:N {#1}
4075 }
(End definition for \__xeCJK_under_CJKsymbol:N.)

```

CJKfilltwosides 使用 minipage 和 L^AT_EX 表格(tabular)来定义 CJKfilltwosides 环境。可选参数 #1 表示环境的垂直对齐位置,默认居中;参数 #2 表示环境的宽度。带星号的环境,如果 #2 不大于零或者不大于环境最长文本行的宽度,则取环境的自然宽度。

```

4076 \RenewDocumentEnvironment { CJKfilltwosides } { 0 { c } m }
4077 {
4078   \use:x { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
4079   \cs_set_eq:NN \CJKglue \tex_hfill:D
4080 }
4081 {
4082   \endminipage
4083   \ignorespacesafterend
4084 }
4085 \NewEnviron { CJKfilltwosides* } [ 2 ] [ c ]
4086 {
4087   \cs_set_eq:NN \CJKglue \tex_hfill:D
4088   \tl_set:Nn \arraystretch { 1 }
4089   \token_if_dim_register:NT \extrarowheight
4090   { \dim_set_eq:NN \extrarowheight \c_zero_dim }
4091   \dim_compare:nNnTF {#2} > \c_zero_dim
4092   {
4093     \hbox_set:Nn \l__xeCJK_tmp_box
4094     {
4095       \tabular [#1] { @ { } c @ { } }
4096       \BODY
4097     \endtabular
4098   }
4099   \dim_compare:nNnTF {#2} > { \box_wd:N \l__xeCJK_tmp_box }
4100   {
4101     \tabular [#1] { @ { } p { \dim_eval:n {#2} } @ { } }
4102     \BODY
4103   \endtabular
4104   }
4105   { \hbox_unpack:N \l__xeCJK_tmp_box }
4106 }
4107 {
4108   \tabular [#1] { @ { } c @ { } }
4109   \BODY
4110 \endtabular
4111 }
4112 }
4113 [ \ignorespacesafterend ]
(End definition for CJKfilltwosides.)
4114 \</fntef>

```

5.20 xeCJK-listings

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

```

4115 (*listings)
4116 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
4117 \ProcessOptions \scan_stop:
4118 \RequirePackage { xeCJK }
4119 \RequirePackage { listings }
4120 \lst@AddToHook { Init } { \__xeCJK_listings_initial_hook: }
4121 \lst@AddToHook { SelectCharTable } { \__xeCJK_listings_toks_hook: }
4122 \lst@AddToHook { OutputBox }
4123 {
4124   \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
4125   \l__xeCJK_restore_listings_toks_tl
4126   \__xeCJK_listings_output_IVS:
4127 }
4128 \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 `\shipout` 盒子中

恢复成正常定义。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。

```

4129 \cs_new_protected_nopar:Npn \__xeCJK_listings_initial_hook:
4130 {
4131   \tex_noindent:D
4132   \bool_gset_false:N \g__xeCJK_listings_IVS_bool
4133   \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
4134   \__xeCJK_add_to_shipout:n { \l__xeCJK_restore_listings_toks_tl }
4135   \lst@ifbreaklines
4136     \cs_set_eq:NN \__xeCJK_listings_CJK_toks: \__xeCJK_listings_breaklines_toks:
4137   \fi:
4138 }

```

(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。

```

4139 \cs_new_protected_nopar:Npn \__xeCJK_listings_toks_hook:
4140 {
4141   \tl_set:Nx \l__xeCJK_restore_listings_toks_tl
4142   {
4143     \__xeCJK_backup_inter_class_toks:nn { Boundary } { Default }
4144     \__xeCJK_backup_inter_class_toks:nn { Boundary } { CJK }
4145     \__xeCJK_backup_inter_class_toks:nn { Boundary } { IVS }
4146     \__xeCJK_backup_inter_class_toks:nn { Boundary } { FullLeft }
4147     \__xeCJK_backup_inter_class_toks:nn { Boundary } { FullRight }
4148   }
4149   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
4150   {
4151     \tl_put_right:Nx \l__xeCJK_restore_listings_toks_tl
4152     { \__xeCJK_backup_inter_class_toks:nn { Boundary } { CJK/##1 } }
4153   }
4154   \xeCJK_inter_class_toks:nnn { Boundary } { Default }
4155   { \__xeCJK_listings_process_Default:N }
4156   \xeCJK_inter_class_toks:nnn { Boundary } { IVS }
4157   { \__xeCJK_listings_process_IVS:nN { \c_zero } }
4158   \__xeCJK_listings_CJK_toks_hook:
4159 }
4160 \tl_new:N \l__xeCJK_restore_listings_toks_tl
4161 \cs_new_nopar:Npn \__xeCJK_backup_inter_class_toks:nn #1#2
4162 {
4163   \xeCJK_inter_class_toks:nnn {#1} {#2}
4164   { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
4165 }

```

(End definition for `__xeCJK_listings_toks_hook:.`)

`__xeCJK_listings_CJK_toks_hook:` 根据 `breaklines` 选项的使用与否, 选择不同的处理方式。

```

\__xeCJK_listings_breaklines_toks: 4166 \cs_new_protected_nopar:Npn \__xeCJK_listings_CJK_toks_hook:
4167 {
4168   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
4169   { \__xeCJK_listings_process_CJK:nN { \c_two } }
4170   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
4171   { \__xeCJK_listings_process_CJK:nN { \c_two } }
4172   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
4173   { \__xeCJK_listings_process_CJK:nN { \c_two } }
4174   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
4175   {
4176     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
4177     { \__xeCJK_listings_process_CJK:nN { \c_two } }
4178   }
4179 }
4180 \cs_new_protected_nopar:Npn \__xeCJK_listings_breaklines_toks:
4181 {
4182   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
4183   { \__xeCJK_listings_process_breaklines_CJK:nN { \c_two } }
4184   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
4185   { \__xeCJK_listings_process_FullLeft:nN { \c_two } }
4186   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }

```



```

4187     { \_xeCJK_listings_process_FullRight:nN { \c_two } }
4188 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
4189 {
4190     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
4191     { \_xeCJK_listings_process_breaklines_CJK:nN { \c_two } }
4192 }
4193 }
(End definition for \_xeCJK_listings_CJK_toks_hook: and \_xeCJK_listings_breaklines_toks:.)

```

_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 4194 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_Default:N #1
4195 {
4196     \token_if_letter:NTF #1
4197     { \lst@ProcessLetter #1 }
4198     { \lst@ProcessOther #1 }
4199 }
4200 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_CJK:nN #1#2
4201 {
4202     \token_if_letter:NTF #2
4203     { \_xeCJK_listings_process_letter:nN {#1} #2 }
4204     { \_xeCJK_listings_process_other:nN {#1} #2 }
4205 }
(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。

```

4206 \cs_new_protected_nopar:Npn \_xeCJK_listings_append:nN #1#2
4207 {
4208     \int_add:Nn \lst@length { #1 - \c_one }
4209     \lst@Append #2
4210 }
(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

```

4211 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_letter:nN
4212 {
4213     \lst@whitespacefalse
4214     \bool_if:NTF \l__xeCJK_listings_letter_bool
4215     { \lst@lettertrue }
4216     {
4217         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
4218         \bool_set_true:N \l__xeCJK_listings_letter_bool
4219     }
4220     \_xeCJK_listings_append:nN
4221 }
4222 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_other:nN #1#2
4223 {
4224     \lst@whitespacefalse
4225     \bool_if:NTF \l__xeCJK_listings_letter_bool
4226     {
4227         \lst@Output \lst@letterfalse
4228         \bool_set_false:N \l__xeCJK_listings_letter_bool
4229     }
4230     { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
4231     \cs_set_eq:NN \lst@lastother #2
4232     \_xeCJK_listings_append:nN {#1} #2
4233 }
(End definition for \_xeCJK_listings_process_letter:nN and \_xeCJK_listings_process_other:nN.)

```

_xeCJK_listings_process_breaklines_CJK:nN 当使用 breaklines 选项时,立即输出之前的单个文字,以便于断行。并将标点与它前/后的 CJK 文字放在同一个盒子中,以保持禁则。但是不能区分 letter 和 other。

_xeCJK_listings_process_FullLeft:nN

_xeCJK_listings_process_FullRight:nN

```

4234 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_breaklines_CJK:nN
4235 {
4236     \lst@whitespacefalse
4237     \bool_if:NTF \l__xeCJK_listings_letter_bool

```

```

4238     {
4239         \int_compare:nNf \l__xeCJK_listings_flag_int = \c_two { \lst@Output }
4240         \lst@lettertrue
4241     }
4242     {
4243         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
4244         \bool_set_true:N \l__xeCJK_listings_letter_bool
4245     }
4246     \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one
4247     \__xeCJK_listings_append:nN
4248 }
4249 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_FullLeft:nN #1#2
4250 {
4251     \lst@whitespacefalse
4252     \bool_if:NTF \l__xeCJK_listings_letter_bool
4253     {
4254         \bool_if:nF
4255         {
4256             \int_compare_p:nNn \l__xeCJK_listings_flag_int = \c_two ||
4257             ( \int_compare_p:nNn \l__xeCJK_listings_flag_int = \c_three &&
4258               ! \l__xeCJK_punct_breakable_bool )
4259         }
4260         { \lst@Output }
4261         \lst@lettertrue
4262     }
4263     {
4264         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
4265         \bool_set_true:N \l__xeCJK_listings_letter_bool
4266     }
4267     \int_set_eq:NN \l__xeCJK_listings_flag_int \c_two
4268     \__xeCJK_listings_append:nN {#1} #2
4269 }
4270 \cs_new_protected_nopar:Npn \__xeCJK_listings_process_FullRight:nN #1#2
4271 {
4272     \lst@whitespacefalse
4273     \bool_if:NTF \l__xeCJK_listings_letter_bool
4274     {
4275         \bool_if:nT
4276         {
4277             \int_compare_p:nNn \l__xeCJK_listings_flag_int < \c_two &&
4278             \__xeCJK_punct_if_long_p:N #2
4279         }
4280         { \lst@Output }
4281         \lst@lettertrue
4282     }
4283     {
4284         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
4285         \bool_set_true:N \l__xeCJK_listings_letter_bool
4286     }
4287     \int_set_eq:NN \l__xeCJK_listings_flag_int \c_three
4288     \__xeCJK_listings_append:nN {#1} #2
4289 }
4290 \int_new:N \l__xeCJK_listings_flag_int
(End definition for \__xeCJK_listings_process_breaklines_CJK:nN, \__xeCJK_listings_process_FullLeft:nN, and
\__xeCJK_listings_process_FullRight:nN.)

```

\lst@AppendLetter

\lst@AppendOther

```

4291 \cs_set_protected_nopar:Npn \lst@AppendLetter
4292 {
4293     \bool_if:NTF \l__xeCJK_listings_letter_bool
4294     {
4295         \lst@Output \lst@lettertrue
4296         \bool_set_false:N \l__xeCJK_listings_letter_bool
4297     }
4298     { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
4299     \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
4300     \lst@Append
4301 }

```

```

4302 \cs_set_protected_nopar:Npn \lst@AppendOther
4303 {
4304   \bool_if:NTF \l__xeCJK_listings_letter_bool
4305   {
4306     \lst@Output \lst@letterfalse
4307     \bool_set_false:N \l__xeCJK_listings_letter_bool
4308   }
4309   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
4310   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
4311   \tex_futurelet:D \lst@lastother \lst@Append
4312 }
(End definition for \lst@AppendLetter and \lst@AppendOther.)

```

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

```

4313 \cs_new_protected_nopar:Npn \_xeCJK_listings_process_IVS:nN
4314 {
4315   \reverse_if:N \lst@ifflexible
4316   \bool_gset_true:N \g__xeCJK_listings_IVS_bool
4317   \fi:
4318   \_xeCJK_listings_process_letter:nN
4319 }
(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 \LaTeX 将基本字和 IVS 正确的组合起来。

```

4320 \cs_new_protected_nopar:Npn \_xeCJK_listings_output_IVS:
4321 {
4322   \reverse_if:N \lst@ifflexible
4323   \bool_if:NT \g__xeCJK_listings_IVS_bool
4324   {
4325     \bool_gset_false:N \g__xeCJK_listings_IVS_bool
4326     \xeCJK_cs_clear:N \lst@FillOutputBox
4327     \cs_set_eq:NN \CJKglue \tex_hss:D
4328   }
4329   \fi:
4330 }
4331 \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。

```

4332 \cs_new_protected:Npn \_xeCJK_listings_peek_active_loop:TF #1#2#3
4333 {
4334   \token_if_active:NTF #3
4335   { #1#3 }
4336   {
4337     \token_if_cs:NTF #3
4338     { #2#3 }
4339     {
4340       \int_compare:nNnTF { `#3 } > { \lst@ifec 255 \else: 127 \fi: }
4341       { \_xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
4342       { #2#3 }
4343     }
4344   }
4345 }
4346 \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。

```

4347 \cs_new_protected:Npn \_xeCJK_listings_inside_convert:nw #1 ~ \@empty
4348 {
4349   \tl_set_rescan:Nnn \l__xeCJK_tmp_tl { } {#1}
4350   \_xeCJK_set_listings_escape:

```

```

4351 \tl_put_right:NV \lst@arg \l__xeCJK_tmp_tl
4352 }
4353 \cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
4354 \cs_new_protected_nopar:Npn \__xeCJK_listings_inline_group:w
4355 {
4356 \exp_after:wN \__xeCJK_listings_inline_group:n
4357 \exp_after:wN { \if_false: } \fi:
4358 }
4359 \cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
4360 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
4361 {
4362 \tl_set_rescan:Nnn \lst@arg { } {#1}
4363 \__xeCJK_set_listings_escape:
4364 \lst@InlineGJEnd
4365 }
(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 TeX 特殊字符, 与原来宏包一致。

```

4366 \group_begin:
4367 \cs_set:Npn \__xeCJK_tmp:w #1
4368 {
4369 \group_end:
4370 \cs_new_protected:Npn \__xeCJK_set_listings_escape:
4371 { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
4372 \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
4373 { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
4374 }
4375 \use:n
4376 {
4377 \char_set_catcode_active:N \
4378 \__xeCJK_tmp:w
4379 }
4380 { \ }
(End definition for \__xeCJK_set_listings_escape:.)
4381 </listings>

```

5.21 xunicode-addon

```

4382 <*xunicode>

```

xunicode 对编码相关的符号命令的定义中用的是诸如 \char"0022\relax 的形式。例如 \textbar 被展开为 \char"007C\relax。并且诸如下述的定义是无效的：

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

我们在这里做的修改是把符号命令定义为实际的字符并且使上述定义生效。另外在使用这些符号命令的时候, 先判断当前字体中是否存在对应的字符, 如果不存在, 则使用这些符号命令的默认设置。

```

4383 \pdfTeX_if_engine:T
4384 {
4385 \msg_new:nnnn { xunicode-addon } { cannot-use-pdfTeX }
4386 { This~package~requires~either~XeTeX~or~LuaTeX~to~function.}
4387 {
4388 You~must~change~your~typesetting~engine~to,~e.g.,\
4389 "xelatex"~or~"lualatex"~instead~of~plain~"latex"~or~"pdflatex".
4390 }
4391 \msg_critical:nn { xunicode-addon } { cannot-use-pdfTeX }
4392 }
4393 \RequirePackage { xparse }

宏包选项是编码的名字。

4394 \clist_new:N \g__xunadd_encname_clist
4395 \DeclareOption*
4396 { \clist_gput_left:NV \g__xunadd_encname_clist \CurrentOption }
4397 \ProcessOptions \scan_stop:
4398 \tl_if_exist:NT \UTFencname
4399 { \clist_gput_left:Nx \g__xunadd_encname_clist { \UTFencname } }

```

若 xunicode 已经被调用,则在宏包结束的时候,重新设置 \UTFencname 对应的编码命令。否则设置 \UTFencname,如果使用的是 Lua¹TeX,则需要作一些设置,使得 xunicode 可用。

```

4400 \ifpackageloaded { xunicode } { }
4401 {
4402   \clist_get:NNF \g__xunadd_encname_clist \UTFencname
4403   {
4404     \xetex_if_engine:TF
4405     { \tl_set:Nn \UTFencname { EU1 } }
4406     { \tl_set:Nn \UTFencname { EU2 } }
4407     \clist_set_eq:NN \g__xunadd_encname_clist \UTFencname
4408   }
4409   \xetex_if_engine:TF
4410   { \RequirePackage { xunicode } }
4411   {
4412     \cs_set_eq:NN \__xunadd_tmp:w \XeTeXpicfile
4413     \cs_set_eq:NN \XeTeXpicfile \prg_do_nothing:
4414     \RequirePackage { xunicode }
4415     \cs_set_eq:NN \XeTeXpicfile \__xunadd_tmp:w
4416   }
4417 }
4418 \AtEndOfPackage { \ReloadXunicode { \g__xunadd_encname_clist } }

```

\ReloadXunicode 参数可以是多个编码,设置这些编码对应的命令。如果编码没有预先声明,则给出一个错误警告。

```

4419 \RenewDocumentCommand \ReloadXunicode { m }
4420 {
4421   \clist_set:Nx \l__xunadd_encname_clist {#1}
4422   \clist_remove_duplicates:N \l__xunadd_encname_clist
4423   \use:x
4424   {
4425     \bool_if:NT \l__kernel_expl_bool { \ExplSyntaxOff }
4426     \char_set_catcode_letter:n { 64 }
4427     \__xunadd_reload:N \exp_not:N \l__xunadd_encname_clist
4428     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
4429     \bool_if:NT \l__kernel_expl_bool { \ExplSyntaxOn }
4430   }
4431 }
4432 \cs_new_protected:Npn \__xunadd_reload:N #1
4433 {
4434   \cs_set_eq:NN \__xunadd_tmp:w \iftipaonetoken
4435   \cs_set_eq:NN \iftipaonetoken \scan_stop:
4436   \clist_map_inline:Nn #1
4437   {
4438     \cs_if_exist:cTF { T@ ##1 }
4439     {
4440       \tl_set:Nx \UTFencname {##1}
4441       \clist_gput_right:Nx \g__xunadd_encname_clist {##1}
4442       \file_input:n { xunicode.sty }
4443     }
4444     { \msg_error:nnn { xunicode-addon } { encoding-unknown } {##1} }
4445   }
4446   \cs_set_eq:NN \iftipaonetoken \__xunadd_tmp:w
4447   \clist_gremove_duplicates:N \g__xunadd_encname_clist
4448 }
4449 \clist_new:N \l__xunadd_encname_clist
4450 \msg_new:nnnn { xunicode-addon } { encoding-unknown }
4451 {Encoding~scheme~"##1"~unknown.}
4452 {
4453   You~may~use \\\
4454   \token_to_str:N \usepackage [ #1 , \encodingdefault ] {fontenc} \\\
4455   before~xunicode-addon~or~xunicode.
4456 }

```

(End definition for \ReloadXunicode.)

__xunadd_glyph_if_exist_p:n
__xunadd_glyph_if_exist:nTF

判断字符在当前字体中是否存在。

```

4457 \prg_new_conditional:Npnn \__xunadd_glyph_if_exist:n #1 { p , T , F , TF }
4458 {
4459   \etex_iffontchar:D \tex_font:D \etex_numexpr:D #1 \scan_stop:

```

```

4460     \prg_return_true: \else: \prg_return_false: \fi:
4461   }
(End definition for \__xunadd_glyph_if_exist:n)

```

\UndeclareUTFcharacter 取消编码 #1 下的符号命令 #3。

```

4462 \RenewDocumentCommand \UndeclareUTFcharacter { 0 { \UTFencname } m m }
4463 {
4464   \__xunadd_if_csname:nTF {#3}
4465   { \UndeclareTextCommand {#3} }
4466   { \exp_args:Nc \UndeclareTextCommand { \tl_to_str:n {#3} } }
4467   {#1}
4468 }
(End definition for \UndeclareUTFcharacter.)

```

\UndeclareUTFcomposite 取消编码 #1 下的复合符号命令 #3{#4}。

```

4469 \RenewDocumentCommand \UndeclareUTFcomposite { 0 { \UTFencname } m m m }
4470 {
4471   \__xunadd_if_csname:nTF {#3}
4472   { \__xunadd_undeclare_composite:Nnnn #3 }
4473   { \exp_args:Nc \__xunadd_undeclare_composite:Nnnn { \tl_to_str:n {#3} } }
4474   {#1} {#4} {#2}
4475 }
4476 \cs_new_protected:Npn \__xunadd_undeclare_composite:Nnnn #1#2#3#4
4477 { \cs_undefine:c { \token_to_str:c {#2} \token_to_str:N #1 - \tl_to_str:n {#3} } }
(End definition for \UndeclareUTFcomposite.)

```

__xunadd_if_csname:nTF 判断 #1 是否可以作为控制序列的名字。这是因为 xunicode 使用了下面的定义。

```

\DeclareUTFcharacter[\UTFencname]{x0149}{'n}

4478 \prg_new_conditional:Npnn \__xunadd_if_csname:n #1 { TF }
4479 {
4480   \tl_if_single_token:nTF {#1}
4481   {
4482     \if_predicate:w
4483     \bool_if_p:n { \token_if_cs_p:N #1 || \token_if_active_p:N #1 }
4484     \prg_return_true: \else: \prg_return_false: \fi:
4485   }
4486   { \prg_return_false: }
4487 }
(End definition for \__xunadd_if_csname:nTF.)

```

\DeclareUTFcharacter 定义编码 #1 下的符号命令 #3, 其对应符号的 Unicode 是 #2。

```

4488 \RenewDocumentCommand \DeclareUTFcharacter { 0 { \UTFencname } m m }
4489 {
4490   \__xunadd_if_csname:nTF {#3}
4491   { \__xunadd_declare_character:Nnn #3 }
4492   { \__xunadd_declare_character:cnn { \tl_to_str:n {#3} } }
4493   {#1} {#2}
4494 }
(End definition for \DeclareUTFcharacter.)

```

__xunadd_declare_character:Nnn 通过 lowercase 技巧, 直接由 Unicode #3 得到编码 #2 下的符号命令 #1 对应的实际字符。

```

4495 \cs_new_protected:Npn \__xunadd_declare_character:Nnn #1#2#3
4496 {
4497   \__xunadd_provide_text_command_default:N #1
4498   \group_begin:
4499   \char_set_lccode:nn { `0 } { \__xunadd_check_slot:n {#3} }
4500   \tl_to_lowercase:n
4501   {
4502     \group_end:
4503     \__xunadd_declare_character:NNxn 0
4504   }
4505   #1 { \token_to_str:N #1 } {#2}
4506 }
4507 \cs_generate_variant:Nn \__xunadd_declare_character:Nnn { c }
(End definition for \__xunadd_declare_character:Nnn.)

```

`_xunadd_provide_text_command_default:N` 如果控制序列 #1 已经存在,但不是符号命令,xunicode 会将它定义为 `\UTFencname` 编码下的符号命令。但是编码被转换之后,再使用这些控制序列,NFSS 就会报错。为此需要给出这些符号命令的默认定义,与原来的意义相同。这些命令包括

```
\nobreakspace macro:->\protect \nobreakspace
\copyright      macro:->\protect \copyright
\AA             macro:->\r A
\aa             macro:->\r a
\texttrhookopeno \long macro:->\textrethookbelow {\textopeno }
\hbar           macro:->{\mathchar '26\mkern -9muh}
\texttaolig     macro:->{a\kern -.25em o}
```

影响比较大的是 `\nobreakspace`、`\copyright` 和 `\hbar`。

```
4508 \cs_new_protected:Npn \_xunadd_provide_text_command_default:N #1
4509 {
4510   \bool_if:nF
4511   {
4512     \cs_if_exist_p:c { ? \token_to_str:N #1 } ||
4513     \cs_if_free_p:c { ? - \token_to_str:N #1 }
4514   }
4515   { \exp_args:NNv \ProvideTextCommandDefault #1 { ? - \token_to_str:N #1 } }
4516 }
```

(End definition for `_xunadd_provide_text_command_default:N`.)

`_xunadd_declare_character:NNnn` 使用编码 #4 下的符号命令 #2 的时候先判断它对应的实际字符 #1 在当前字体中是否存在。如果不存在则转换到 `\DeclareTextSymbolDefault` 中设置的编码或者使用 `\DeclareTextCommandDefault` 中设置的命令。

```
4517 \cs_new_protected:Npn \_xunadd_declare_character:NNnn #1#2#3#4
4518 { \DeclareTextCommand #2 {#4} { \_xunadd_text_character:nN {#3} {#1} } }
4519 \cs_new_protected:Npn \_xunadd_text_character:nN #1#2
4520 {
4521   \_xunadd_begin_hook:nn {#1} {#2}
4522   \_xunadd_glyph_if_exist:nTF { `#2 }
4523   {#2} { \cs_if_exist_use:cF { ? #1 } {#2} }
4524   \_xunadd_end_hook:nn {#1} {#2}
4525 }
4526 \cs_generate_variant:Nn \_xunadd_declare_character:NNnn { NNx }
```

(End definition for `_xunadd_declare_character:NNnn`.)

`_xunadd_check_slot:n` xunicode 中使用的 Unicode 格式是诸如 x0022 的形式,这就需要一些转换。

```
4527 \cs_new_nopar:Npn \_xunadd_check_slot:n #1
4528 {
4529   \int_eval:n
4530   {
4531     \tl_if_head_eq_charcode:nNTF {#1} x
4532     { " \use_none:n #1 } {#1}
4533   }
4534 }
```

(End definition for `_xunadd_check_slot:n`.)

`\DeclareUTFcomposite` 设置编码 #1 下的符号命令 #3 与它的参数 #4 的复合对应的符号的 Unicode 是 #2。

```
4535 \RenewDocumentCommand \DeclareUTFcomposite { O { \UTFencname } m m m }
4536 {
4537   \_xunadd_if_csname:nTF {#3}
4538   { \_xunadd_declare_composite:Nnnn #3 }
4539   { \_xunadd_declare_composite:cnnn { \tl_to_str:n {#3} } }
4540   {#1} {#4} {#2}
4541 }
```

(End definition for `\DeclareUTFcomposite`.)

`_xunadd_declare_composite:Nnnn` 这里使用 `\tex_afterassignment:D` 是因为 xunicode 有如下的定义。

```
\DeclareUTFcomposite[\UTFencname]{x02E8\char"02E5}{\tonebar}{25}
\DeclareUTFcomposite[\UTFencname]{x02E5\char"02E8}{\tonebar}{52}
```

对复合符号命令的定义用的是 `\chardef`, 这有利于下面字符是否存在的判断。

```
4542 \cs_new_protected:Npn \__xunadd_declare_composite:Nnnn #1#2#3#4
4543 {
4544   \tex_afterassignment:D \use_none_delimit_by_q_stop:w
4545   \__xunadd_chardef:cn
4546   { \token_to_str:c {#2} \token_to_str:N #1 - \tl_to_str:n {#3} }
4547   { \__xunadd_check_slot:n {#4} }
4548   \q_stop
4549 }
4550 \cs_new_protected:Npn \__xunadd_chardef:Nn #1#2
4551 { \tex_chardef:D #1 = \etex_numexpr:D #2 \scan_stop: }
4552 \cs_generate_variant:Nn \__xunadd_chardef:Nn { c }
4553 \cs_generate_variant:Nn \__xunadd_declare_composite:Nnnn { c }
(End definition for \__xunadd_declare_composite:Nnnn.)
```

`\tonebar` 使得 `\tonebar{25}` 和 `\tonebar{52}` 生效。

```
4554 \AtEndOfPackage
4555 {
4556   \DeclareUTFCompositeCommand \tonebar { 25 } { \tonebar { 2 } \tonebar { 5 } }
4557   \DeclareUTFCompositeCommand \tonebar { 52 } { \tonebar { 5 } \tonebar { 2 } }
4558 }
(End definition for \tonebar.)
```

`\DeclareUTFCompositeCommand` 设置编码 #1 下的符号命令 #2 与它的参数 #3 的复合对应结果是 #4。不能直接用 `\DeclareTextCompositeCommand` 来定义, 它与我们的机制冲突。

```
4559 \NewDocumentCommand \DeclareUTFCompositeCommand { 0 { \UTFencname } m m m }
4560 {
4561   \__xunadd_if_csname:nTF {#2}
4562   { \__xunadd_declare_composite_command:Nnnn #2 }
4563   { \__xunadd_declare_composite_command:cnnn { \tl_to_str:n {#2} } }
4564   {#1} {#3} {#4}
4565 }
(End definition for \DeclareUTFCompositeCommand.)
```

`__xunadd_declare_composite_command:Nnnn`

```
4566 \cs_new_protected:Npn \__xunadd_declare_composite_command:Nnnn #1#2#3#4
4567 {
4568   \cs_set_protected:cpn
4569   { \token_to_str:c {#2} \token_to_str:N #1 - \tl_to_str:n {#3} }
4570   {#4}
4571 }
4572 \cs_generate_variant:Nn \__xunadd_declare_composite_command:Nnnn { c }
(End definition for \__xunadd_declare_composite_command:Nnnn.)
```

`\DeclareEncodedCompositeCharacter` #1 是编码, #2 是重音命令, #3 是组合重音符号的 Unicode, #4 是基本重音符号的 Unicode。当 #2 的参数为空时, 输出 #4, 否则是 #2 与 #3 的组合。

```
4573 \RenewDocumentCommand \DeclareEncodedCompositeCharacter { m m m m }
4574 {
4575   \__xunadd_if_csname:nTF {#2}
4576   { \__xunadd_declare_accent:Nnnn #2 }
4577   { \__xunadd_declare_accent:cnnn { \tl_to_str:n {#2} } }
4578   {#1} {#3} {#4}
4579 }
(End definition for \DeclareEncodedCompositeCharacter.)
```

`\DeclareEncodedCompositeAccents` #1 是编码, #2 是重音命令, #3 和 #4 都是组合重音的 Unicode。当 #2 的参数为空时, 输出 #3, 否则是 #2 与 #4 的组合。

```
4580 \RenewDocumentCommand \DeclareEncodedCompositeAccents { m m m m }
4581 {
4582   \__xunadd_if_csname:nTF {#2}
4583   { \__xunadd_declare_accent:Nnnn #2 }
4584   { \__xunadd_declare_accent:cnnn { \tl_to_str:n {#2} } }
4585   {#1} {#4} {#3}
4586 }
(End definition for \DeclareEncodedCompositeAccents.)
```


`_xunadd_declare_accent:Nnnn` 通过 `lowercase` 技巧,直接由重音符号的 Unicode 得到实际字符。

```
4587 \cs_new_protected:Npn \_xunadd_declare_accent:Nnnn #1#2#3#4
4588 {
4589   \group_begin:
4590   \char_set_lccode:nn { `3 } { "#3 }
4591   \char_set_lccode:nn { `4 } { \tl_if_blank:nTF {#4} { "#3 } { "#4 } }
4592   \tl_to_lowercase:n
4593   {
4594     \group_end:
4595     \_xunadd_declare_accent:NNNxxn 3 4
4596   }
4597   #1 { \token_to_str:c {#2} } { \token_to_str:N #1 } {#2}
4598 }
4599 \cs_new_protected:Npn \_xunadd_declare_accent:NNNnnn #1#2#3#4#5#6
4600 {
4601   \DeclareTextCommand #3 {#6}
4602   { \_xunadd_text_composite:nnNNn {#4} {#5} {#1} {#2} }
4603 }
4604 \cs_generate_variant:Nn \_xunadd_declare_accent:Nnnn { c }
4605 \cs_generate_variant:Nn \_xunadd_declare_accent:NNNnnn { NNNxx }
(End definition for \_xunadd_declare_accent:Nnnn.)
```

`_xunadd_text_composite:nnNNn` 若重音命令 #2 与它的参数 #5 的复合已经由 `\DeclareUTFcomposite` 设置,并且在当前字体中存在该字符,则直接使用。否则使用组合命令。

```
4606 \cs_new_protected:Npn \_xunadd_text_composite:nnNNn #1#2#3#4#5
4607 {
4608   \_xunadd_begin_hook:nn {#2} {#5}
4609   \cs_if_exist:cTF { #1#2 - \tl_to_str:n {#5} }
4610   {
4611     \_xunadd_text_composite_aux:cnNNn
4612     { #1#2 - \tl_to_str:n {#5} } {#2} {#3} {#4} {#5}
4613   }
4614   { \_xunadd_add_accent:nnNN {#5} {#2} {#3} {#4} }
4615   \_xunadd_end_hook:nn {#2} {#5}
4616 }
4617 \cs_new_protected:Npn \_xunadd_text_composite_aux:NnNNn #1#2#3#4#5
4618 {
4619   \token_if_chardef:NTF #1
4620   {
4621     \_xunadd_glyph_if_exist:nTF {#1}
4622     {#1} { \_xunadd_add_accent:nnNN {#5} {#2} {#3} {#4} }
4623   }
4624   {#1}
4625 }
4626 \cs_generate_variant:Nn \_xunadd_text_composite_aux:NnNNn { c }
(End definition for \_xunadd_text_composite:nnNNn.)
```

`_xunadd_add_accent:nnNN` 若组合重音符号的 #3 和基本重音符号 #4 在当前字体中都不存在,则转换到 `\DeclareTextAccentDefault` 设置的编码或者使用 `\DeclareTextCommandDefault` 中设置的命令。0.9999 版以前的 \XeTeX 需要设置 `\XeTeXinputnormalization` 为 1,才能使用字体中由基础字符和组合符号对应的实际字符;而 0.9999 版以后的 \XeTeX 默认就启用这个功能,`\XeTeXinputnormalization` 似乎是无效的,怀疑是使用 HarfBuzz 库替代 ICU 进行字体排版的缘故¹¹。

```
4627 \cs_new_protected:Npn \_xunadd_add_accent:nnNN #1#2#3#4
4628 {
4629   \tl_if_blank:nTF {#1}
4630   {
4631     \_xunadd_glyph_if_exist:nTF { `#4 }
4632     {#4}
4633     {
4634       \cs_if_exist_use:cTF { ? #2 }
4635       { {#1} } {#4}
4636     }
4637   }
4638   {
```

¹¹<http://tug.org/pipermail/xetex/2013-July/024579.html>

```

4639     \__xunadd_glyph_if_exist:nTF { `#3 }
4640     { #1#3 }
4641     {
4642         \__xunadd_glyph_if_exist:nTF { `#4 }
4643         { \add@accent { `#4 } {#1} }
4644         {
4645             \cs_if_exist_use:cTF { ? #2 }
4646             { {#1} } {#1}
4647         }
4648     }
4649 }
4650 }

```

(End definition for __xunadd_add_accent:nnNN.)

\AtBeginUTFCommand
\AtEndUTFCommand

设置在符号命令前后使用的钩子,可选参数用于指定单个符号命名。可以用 #1 引用带参数的组合符号命令的参数或者符号命令对应的符号。

```

4651 \NewDocumentCommand \AtBeginUTFCommand { s O { } +m }
4652 {
4653     \tl_if_blank:nTF {#2}
4654     {
4655         \IfBooleanTF {#1}
4656         { \tl_set:Nn \l__xunadd_begin_hook_tl {#3} }
4657         { \tl_put_right:Nn \l__xunadd_begin_hook_tl {#3} }
4658     }
4659     { \__xunadd_set_cmd_hook:nnn { begin } {#2} {#3} }
4660 }
4661 \NewDocumentCommand \AtEndUTFCommand { s O { } +m }
4662 {
4663     \tl_if_blank:nTF {#2}
4664     {
4665         \IfBooleanTF {#1}
4666         { \tl_set:Nn \l__xunadd_end_hook_tl {#3} }
4667         { \tl_put_right:Nn \l__xunadd_end_hook_tl {#3} }
4668     }
4669     { \__xunadd_set_cmd_hook:nnn { end } {#2} {#3} }
4670 }
4671 \tl_new:N \l__xunadd_begin_hook_tl
4672 \tl_new:N \l__xunadd_end_hook_tl

```

(End definition for \AtBeginUTFCommand and \AtEndUTFCommand.)

__xunadd_set_cmd_hook:nnn

```

4673 \cs_new_protected:Npn \__xunadd_set_cmd_hook:nnn #1#2#3
4674 {
4675     \cs_set_protected:cpn
4676     {
4677         \tl_if_single:nTF {#2}
4678         { \use:c { __xunadd_#1_csname:n } { \token_to_str:N #2 } }
4679         { \__xunadd_set_cmd_hook_aux:Nnwn #2 \q_stop {#1} }
4680     } ##1
4681     {#3}
4682 }
4683 \cs_new:Npn \__xunadd_set_cmd_hook_aux:Nnwn #1#2 \q_stop #3
4684 { \use:c { __xunadd_#3_csname:n } { \token_to_str:N #1 - \tl_to_str:n {#2} } }
4685 \cs_new_nopar:Npn \__xunadd_begin_csname:n #1 { __xunadd_begin_#1_hook:n }
4686 \cs_new_nopar:Npn \__xunadd_end_csname:n #1 { __xunadd_end_#1_hook:n }

```

(End definition for __xunadd_set_cmd_hook:nnn.)

__xunadd_begin_hook:nn
__xunadd_end_hook:nn

```

4687 \cs_new_protected:Npn \__xunadd_begin_hook:nn #1#2
4688 {
4689     \tl_use:N \l__xunadd_begin_hook_tl
4690     \cs_if_exist_use:cF { \__xunadd_begin_csname:n { #1 - \tl_to_str:n {#2} } }
4691     { \cs_if_exist_use:cF { \__xunadd_begin_csname:n {#1} } { \use_none:n } }
4692     {#2}
4693 }
4694 \cs_new_protected:Npn \__xunadd_end_hook:nn #1#2
4695 {

```

```

4696     \cs_if_exist_use:cF { \__xunadd_end_csname:n { #1 - \tl_to_str:n {#2} } }
4697     { \cs_if_exist_use:cF { \__xunadd_end_csname:n {#1} } { \use_none:n } }
4698     {#2}
4699     \tl_use:N \l__xunadd_end_hook_tl
4700 }
(End definition for \__xunadd_begin_hook:nn and \__xunadd_end_hook:nn.)
4701 </xunicode>

```

5.22 xeCJK.cfg

```

4702 <*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。

```

4703
4704 </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。.....	47
删除多余的 <code>default-itcorr</code> 结点。.....	30
取消 <code>\cprotect</code> 的外部宏限制。.....	82
改用 <code>indentfirst</code> 宏包处理缩进的问题。.....	75
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。..	70
放弃对 <code>\outer</code> 宏的特殊处理。.....	1
<code>_xeCJK_switch_font:n</code> : 改进定义, 加快切换速度。..	63
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑 <code>\spaceskip</code> 不为零的情况。.....	16
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。.....	74
<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。.....	79
<code>\xeCJK_fallback_loop:N</code> : 调整备用字体的循环方式。..	56
<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> 。.....	15
<code>\xeCJK_hook_for_ulem::</code> : 简化对 <code>ulem</code> 宏包的兼容补丁。..	83
<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。.....	73
<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。.....	70

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。.....	42
增加小宏包 <code>xeCJKfntef</code> , 用于处理下划线的问题。....	83
对于与 <code>xltxtra</code> 的冲突给出错误警告。.....	79
<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。.....	41
<code>_xeCJK_set_char_class_eq:n</code> : 交换参数的顺序。....	26
<code>_xeCJK_set_verb_exspace::</code> : 调整间距的计算方法。..	71
<code>\fontfamily</code> : 修改主要 CJK 字体族的自动更新方式。....	79
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。....	38
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。.....	43
<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。.....	43
<code>\xeCJK_check_single:NNw</code> : 改进定义, 减少使用 <code>peek</code> 函数的次数。.....	41
<code>\xeCJK_hook_for_ulem::</code> : 完全处理下划线里的标点符号的有关问题。.....	83
<code>\xeCJK_peek_catcode_ignore_spaces:N</code> : 新增有省略空格标识的 <code>peek</code> 函数。.....	17
<code>\xeCJK_save_class:n</code> : 使用 <code>\xeCJK_save_class:n</code> 保存 <code>X_gTeX</code> 预定义的字符类别。.....	19
<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> 。.....	25
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。.....	40

v3.1.2

General: 修正重定义 <code>\CJKfamilydefault</code> 无效的问题, 恢复容错能力。.....	68
解决在下划线状态下使用 <code>\makebox</code> 时的错误。.....	85
<code>_xeCJK_check_single_space:NN</code> : 使用 <code>\xeCJK_if_CJK_class:N</code> 来代替 <code>\int_case:nnn</code> 判断是否是 CJK 字符类。.....	41
<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。..	65
<code>\fontfamily</code> : 不将参数完全展开。.....	79

<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。.....	78
---	----

v3.2.0

General: 增加 <code>IVS</code> 字符类用于处理异体字选择符。.....	20
增加 <code>Verb</code> 选项。.....	70
<code>_xeCJK_Boundary_and_FullLeft_glue:N</code> : 当全角左标点前面是 <code>hlist</code> 、 <code>none</code> 、 <code>glue</code> 和 <code>penalty</code> 等节点时, 压缩其左空白。.....	35
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑到 <code>\spacefactor</code> 和 <code>\xspaceskip</code> 的情况。.....	16
<code>\l_xeCJK_family_tl</code> : 不将其初始化为 <code>\CJKfamilydefault</code> 。.....	65
<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。.....	65
<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。.....	34

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> 。.....	70
--	----

v3.2.2

General: 修正某些重音不能正确显示的问题。.....	1
增加小宏包 <code>xeCJK-listings</code> , 用于支持 <code>listings</code> 宏包。....	91
<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。.....	87

v3.2.3

General: 不再改变 CJK 字符类的 <code>\catcode</code> 。.....	25
完善对 <code>listings</code> 宏包的支持。.....	91
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 <code>X_gTeX</code> 的脚本重新整理全角标点符号。.....	20
解决 <code>CheckSingle</code> 选项与 <code>tablists</code> 宏包的冲突。.....	42
<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。.....	91
<code>_xeCJK_listings_process_CJK:n</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> 。.....	93
<code>_xeCJK_restore_shipout_CJKsymbol::</code> : 解决 <code>\CJKunderdot</code> 跨页使用时影响到页眉页脚的问题。....	90
<code>_xeCJK_ulem_FullLeft_and_CJK::</code> : 修正全角左标点后下划线与 <code>\CJKunderdot</code> 连用时结果不正常的问题。....	88
<code>\xeCJKVerbAddon</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。.....	70

v3.2.4

General: 不再使用 <code>CJKnumber</code> 选项, 可以在 <code>xeCJK</code> 之后直接使用 <code>CJKnumb</code> 宏包得到中文数字。.....	82
使 <code>listings</code> 的 <code>breaklines</code> 选项对 CJK 字符类可用, 并保持标点符号的禁则。.....	93
使用 <code>AllowBreakBetweenPuncts</code> 时, 相应标点符号仍能与边界对齐。.....	38
修正 <code>xeCJKfntef</code> 与 <code>natbib</code> 等的冲突。.....	83
内部调整分区字体的设置方法。.....	58
尽量移除用作判断标志的 <code>\kern</code> 。.....	30
改进获取分区字体属性的办法。.....	58
解决使用 <code>CheckSingle</code> 时, 某些 <code>\CJKglue</code> 不能被正确加入的问题。.....	42
遵循 <code>L_AT_EX3</code> 变量需要预先声明的原则。.....	1
<code>_xeCJK_Boundary_and_FullLeft_glue:N</code> : 细化边界与全角左标点之间是否压缩空白的判断。.....	35
<code>_xeCJK_set_verb_exspace::</code> : 当计算得出的间距为负时, 缩小 CJK 字体。.....	71

\addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。	66	v3.2.6	General: AutoFakeBold 和 AutoFakeSlant 选项直接使用 fontspec 的设置,修正不能调用相应实际字体的问题。 . .	59
CJKfilltwosides: 改用 minipage 和 L ^A T _E X 表格(tabular)来实现。	91		case 类函数的用法与 L ^A T _E X3 同步。	1
\xeCJK_fallback_loop:Nn: 使 \CJKfamilydefault 的 FallBack 设置全局可用。	56		_xeCJK_math_robust:N: 考虑 \math 和 \ensuremath。	77
\xeCJK_tl_remove_outer_braces:n: 去掉外层分组括号时,移除空格,避免死循环。	15		考虑 ulem 对 \MakeRobust 的不当定义。	77
\xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。	18		\AtEndUTFCommand: 可以指定特定符号命令使用的钩子。	102
v3.2.5			\mathrm: 为 \mathrm 减少一个可能的数学字体族。	77
General: 修正 CJK 和 NormalSpace 字符类之间因为边界造成的间距不正确的问题。	29		\xeCJK_CJK_and_Boundary:w: 更好的处理边界是 \relax 的情况。	32
增加小宏包 xunicode-addon, 为 xunicode 提供判断字符是否存在的功能。	96		\xeCJK_set_mathfont:: 设置粗体时先检查对应字体是否存在。	69
\]: 解决 fixltx2e 和 amsthm 的冲突。	78	v3.2.7	General: 使用 everypage 往 \shipout 盒子里加钩子。	14
_xeCJK_Boundary_and_FullLeft_glue:N: 增加对 enu-mitem 宏包修改的 \item 的判断。	35		修正 unicode-letters.tex 中谚文符号 \catcode 不准的问题。	25
细化全角左标点是否位于段首的判断。	35		处理 AllowBreakBetweenPuncts 与 xeCJKfntef 的兼容问题。	38
_xeCJK_math_robust:N: 解决汉字后紧跟 \(\dots\) 形式的行内数学公式时,不能加入间距的问题。	77		实现自定义行首/尾标点符号宽度功能。	45
Verb: 微调定义。	70		标点宽度设置禁用比例选项的值改为 nan。	51
\nobreakspace: 恢复 \nobreakspace 的原始定义。	78		_xeCJK_punct_glue:NN: 标点符号左/右空白的伸展值不超过原始边界,收缩值不小于另一侧边界。	34
\xeCJK_visible_space:: 可视空格考虑传统 T _E X 字体的情况。	73		\Url@MathSetup: 使通过 \UrlFont 等命令设置的 CJK 字体生效。	77
\xeCJKVerbAddon: 禁止自动换行,与西文一致。	70		\xeCJK_check_single:NNw: 与 \CJKspace 兼容。	41
			\xeCJK_set_mathfont:: 将 CJK 字符的数学归类由 7 改为 0,解决汉字路径的问题。	69

代码索引

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

Symbols	
\(.	3365
\)	3365
\@italiccorr	3473, 3481, 3487, 3493, 3496, 3507
\@empty	4347
\@ifpackagelater	46, 49, 3434, 3450
\@ifpackageloaded	4400
\@onefilewithoptions	3658
\@onlypreamble	1447, 1565, 1569, 2155, 2162, 2750, 2751, 2752, 2753, 2754, 2791
\@pgkextension	16
\[.	3273, 3414
\]	5, 6, 7, 19, 24, 40, 41, 42, 43, 344, 345, 1483, 1537, 1538, 2138, 2139, 2140, 2152, 2153, 2245, 2710, 2843, 2925, 2926, 2927, 2928, 2933, 3142, 3144, 3250, 3295, 3296, 3406, 3428, 3429, 4377, 4388, 4453, 4454
\]	3414
_xeCJK_Boundary_and_FullLeft_glue:N	1014, 1023, 1023, 3735
_xeCJK_CJK_and_Boundary_aux:	851, 855, 859, 3730, 4024
_xeCJK_CJK_and_Boundary_relax:N	850, 857
_xeCJK_CJK_and_FullLeft_glue:N	1000, 1005, 3733
_xeCJK_CJK_and_FullRight_glue:N	1101, 1105, 1105, 1119, 3734
_xeCJK_CJK_class_tl:c	461
_xeCJK_CJK_class_tl:n	295, 298, 300
_xeCJK_Default_and_FullLeft_glue:N	983, 992, 3731
_xeCJK_Default_and_FullRight_glue:N	1080, 1091, 1105, 1119, 3732, 3901
_xeCJK_add_special_punct:nn	1587, 1590, 1593, 1618
_xeCJK_add_sub_class_features:n	2813, 2820, 2846, 2846
_xeCJK_add_to_shipout:n	108, 3058, 4056, 4134
_xeCJK_after_end_preamble:n	71, 78, 3348, 3509, 3577, 3590
_xeCJK_after_preamble:n	71, 76, 3008, 3557
_xeCJK_at_end_preamble:n	71, 74, 2880, 3446, 3612, 3625
_xeCJK_aux_r:n	3560, 3568, 3569
_xeCJK_backup_inter_class_toks:nn	4143, 4144, 4145, 4146, 4147, 4152, 4161
_xeCJK_block_select_font:n	2611, 2615, 2615
_xeCJK_calc_kerning_margin:NN	1969, 2004, 2004
_xeCJK_calc_kerning_margin:nNN	2017, 2026, 2070, 2070
_xeCJK_check_family:V	2420, 2532
_xeCJK_check_family:n	2468, 2468, 2477
_xeCJK_check_num_range:nnNN	489, 489, 510, 2974
_xeCJK_check_single_save:N	1265, 1272, 1295, 1306, 1309, 1318, 1342, 1343, 1354, 1355, 1357, 1363, 1366, 1376, 1379
_xeCJK_check_single_space:NN	1307, 1317, 1349, 1349
_xeCJK_class_csname:n	321, 324, 327, 332, 335, 340, 340, 341, 461, 464, 1471, 1489, 3523
_xeCJK_clear_Boundary_and_CJK_toks:n	579, 580
_xeCJK_copy_family:nn	2561, 2561, 2580
_xeCJK_copy_family:xx	2623
_xeCJK_copy_sub_family:n	2523, 2530
_xeCJK_dim_max:nn	3444, 3452, 3456
_xeCJK_dim_min:nn	3445, 3453, 3457
_xeCJK_document_left_hook:	88, 94
_xeCJK_document_right_hook:	90, 96
_xeCJK_error:n	60, 62, 1541
_xeCJK_error:nx	60, 63, 322, 333, 1477, 1547, 2131, 2160, 3244, 3436

__xeCJK_fallback_save_CJKsymbol:N	2194, 2201, 2209, 2224, 2233	__xeCJK_peek_catcode_false:w	229, 248, 253
__xeCJK_family_csname:n	2456, 2547, 2626, 2634, 2634, 2652	__xeCJK_peek_catcode_ignore_spaces_branches:w	232, 234, 239
__xeCJK_family_default_wrap:n	2882, 2896, 3303, 3306	__xeCJK_peek_catcode_true:w	228, 245, 252
__xeCJK_family_nfss_csname:n	2473, 2576, 2577, 2634, 2635, 2636, 2640, 2650	__xeCJK_peek_ignore_spaces_branches:w	259, 261, 265
__xeCJK_family_unknown_warning:n	2695, 2695, 2706	__xeCJK_prop_put_aux:n	2435, 2450
__xeCJK_family_unknown_warning:x	2671, 2683, 2693	__xeCJK_prop_put_aux:nn	2435, 2452
__xeCJK_family_use:x	2589, 2598, 2630, 2634, 2636, 2669, 2692	__xeCJK_punct_bound_rule:NN	913, 913, 948, 958, 1114, 3888, 3913
__xeCJK_font_csname:n	2582, 2583, 2595, 2597, 2617, 2620	__xeCJK_punct_breakable_kern:NN	1153, 1153, 1578
__xeCJK_fontspec:nnn	2779, 2786	__xeCJK_punct_breakable_kern:n	1156, 1160, 3729
__xeCJK_fontspec:xnn	2774	__xeCJK_punct_dim_csname:nn	1651, 1656, 1660
__xeCJK_get_sub_features:Vn	2259	__xeCJK_punct_dim_csname:nnn	1653, 1658, 1662, 1739, 1771
__xeCJK_get_sub_features:nn	2313, 2319, 2319, 2349	__xeCJK_punct_glue:NN	931, 931, 950, 960, 970, 978, 994, 1008, 1074, 1075, 1113, 1204, 3841, 3855, 3868, 3887, 3915, 3941, 3955
__xeCJK_get_sub_features:w	2319, 2323, 2334	__xeCJK_punct_hskip:n	933, 940, 3728
__xeCJK_gobble_CJKfamily:	2686, 2686, 3583	__xeCJK_punct_if_long:NT	2100
__xeCJK_gobble_CJKfamily:wn	2687, 2688	__xeCJK_punct_if_long:NTF	1107, 3881
__xeCJK_group_begin:	3546, 3546, 3554, 3585	__xeCJK_punct_if_long_p:N	4278
__xeCJK_group_end:	3546, 3547, 3555, 3586	__xeCJK_punct_if_middle:NTF	946, 956, 1110, 1801, 1817, 1841, 1866, 1879, 1910, 3884, 3910
__xeCJK_gset_family_cs:x	2421, 2454, 2454	__xeCJK_punct_if_mixed_width:NTF	1804, 1853
__xeCJK_gset_family_nfss_cs:xx	2463, 2554, 2634, 2637	__xeCJK_punct_if_right:N	1637
__xeCJK_info:nxx	60, 68, 2608	__xeCJK_punct_if_right:NF	1979
__xeCJK_int_until_do:wn	219, 220, 221, 222	__xeCJK_punct_if_right:NT	1981
__xeCJK_kerning_width_or_ratio:nNN	2022, 2023, 2041, 2041	__xeCJK_punct_if_right:NTF	1997, 2000, 2076, 2079
__xeCJK_listings_CJK_toks:	4136	__xeCJK_punct_kern:NN	941, 941, 1125, 1134, 1149, 1165
__xeCJK_listings_CJK_toks_hook:	4158, 4166, 4166	__xeCJK_punct_kern:n	928, 929, 3727
__xeCJK_listings_append:nN	4206, 4206, 4220, 4232, 4247, 4268, 4288	__xeCJK_punct_min_bound:NN	2010, 2013, 2056, 2056
__xeCJK_listings_breaklines_toks:	4136, 4166, 4180	__xeCJK_punct_nobreak_kern:NN	1147, 1147, 1152, 1583
__xeCJK_listings_escape:N	4371, 4372, 4373	__xeCJK_punct_offset:NN	927, 927, 969, 977, 995, 1009, 1015, 1198, 3842, 3869, 3940, 3954
__xeCJK_listings_initial_hook:	4120, 4129, 4129	__xeCJK_punct_rule:NN	920, 920, 968, 975, 989, 1002, 1020, 1155, 1158, 1197, 3937, 3951
__xeCJK_listings_inline_group:n	4356, 4360	__xeCJK_punct_width_or_ratio:nN	1802, 1805, 1806, 1900, 1933, 1933
__xeCJK_listings_inline_group:w	4347, 4354, 4359	__xeCJK_remove_duplicate_keys:N	2416, 2432, 2432
__xeCJK_listings_inside_convert:nw	4347, 4347, 4353	__xeCJK_reset_char_class:n	3042, 3043, 3044, 3045, 3046, 3079
__xeCJK_listings_output_IVS:	4126, 4320, 4320	__xeCJK_restore_shipout_CJKsymbol:	4007, 4035, 4041, 4041
__xeCJK_listings_peek_active_loop:TF	4332, 4332, 4341, 4346	__xeCJK_save_CJK_class:n	458, 463, 1526
__xeCJK_listings_process_CJK:nN	4169, 4171, 4173, 4177, 4194, 4200	__xeCJK_save_FullRight_check:	1177, 1187
__xeCJK_listings_process_Default:N	4155, 4194, 4194	__xeCJK_save_FullRight_symbol:N	1178, 1188, 1209
__xeCJK_listings_process_FullLeft:nN	4185, 4234, 4249	__xeCJK_save_family_info:	2422, 2510, 2510
__xeCJK_listings_process_FullRight:nN	4187, 4234, 4270	__xeCJK_save_punct_dim:nnn	1659, 2096, 2098
__xeCJK_listings_process_IVS:nN	4157, 4313, 4313	__xeCJK_save_punct_dim:nnnn	1661, 1743, 1744, 1745, 1746, 1747, 1748, 1749, 1775, 1776, 1861, 1862, 1875, 1929, 1930, 1973, 1983, 2087, 2089, 2102
__xeCJK_listings_process_breaklines_CJK:nN	4183, 4191, 4234, 4234	__xeCJK_save_under_CJKsymbol:N	4005, 4011, 4033, 4046, 4074
__xeCJK_listings_process_letter:nN	4203, 4211, 4211, 4318	__xeCJK_set_char_class_aux:Nnw	470, 479, 485, 2967
__xeCJK_listings_process_other:nN	4204, 4211, 4222	__xeCJK_set_char_class_eq:nn	518, 518, 3031, 3032, 3033, 3034, 3035
__xeCJK_listings_toks_hook:	4121, 4139, 4139	__xeCJK_set_family_initial:	2388, 2388, 2410
__xeCJK_make_under_symbol:n	4004, 4032, 4057, 4057	__xeCJK_set_listings_escape:	4350, 4363, 4366, 4370
__xeCJK_margin_width_or_ratio:n	1854, 1855, 1925, 1946, 1946	__xeCJK_set_others_toks:n	3509, 3517, 3521
__xeCJK_math_robust:N	3365, 3365, 3409, 3410, 3411, 3412, 3413, 3420, 3421	__xeCJK_set_special_punct:nn	1586, 1589, 1592, 1607
__xeCJK_math_robust:NN	3374, 3375, 3377	__xeCJK_set_sub_block_family:	2423, 2517, 2517
__xeCJK_math_robust_aux:NN	3366, 3367	__xeCJK_set_sub_class_toks:nn	1492, 1498, 1498
__xeCJK_msg_def_family_map:n	2711, 2713, 2927, 2934	__xeCJK_set_verb_exspace:	3019, 3087, 3087
__xeCJK_msg_family_map:n	2243, 2479, 2710, 2724, 2932, 2933, 3144	__xeCJK_set_verb_exspace:n	3104, 3110, 3110
__xeCJK_msg_new:nn	60, 60, 342, 1481, 1535, 2136, 2150, 2241, 2478, 2614, 2708, 2841, 2923, 2930, 3140, 3248, 3293, 3426	__xeCJK_set_verb_scale:nn	3119, 3129, 3129
__xeCJK_msg_new:nnn	61, 3403	__xeCJK_set_visible_space_size:n	3177, 3181
__xeCJK_original_kerning_margin:NN	1970, 1992, 1992, 2053	__xeCJK_space_skip_scale:nnn	164, 168, 181, 185, 195
__xeCJK_parse_font_shape:	2419, 2480, 2480	__xeCJK_special_punct_seq:n	1603, 1606, 1609, 1611, 1615, 1622, 1625, 1634
__xeCJK_peek_after_do:w	257, 268		

_xeCJK_special_punct_tl:nN	3037, 3053
..... 1604, 1610, 1614, 1624, 1633, 1647	_xeCJK_verb_CJKglue:
_xeCJK_sub_restore_or_cancel:n	3036, 3052
..... 1467, 1467, 1480	_xeCJK_verb_addon:
_xeCJK_sub_restore_or_cancel:x	3020, 3026
..... 1454, 1463	_xeCJK_verb_boundary:w
_xeCJK_sub_special_punct:nn	3039, 3055
..... 1588, 1591, 1594, 1629	_xeCJK_verb_cgglue:
_xeCJK_swap_cs_aux:w	3067, 3075
..... 133, 135, 136	_xeCJK_verb_check_for_glue:
_xeCJK_switch_font:nn	3038, 3054
..... 1507, 1512, 1521, 1523, 1530, 2604, 2604, 3826, 3831	_xeCJK_verb_ecglue:
_xeCJK_tl_remove_outer_braces:w	3068, 3077
..... 119, 121	_xeCJK_verb_font_hook:
_xeCJK_tmp:w	2990, 3011
..... 4367, 4378	_xeCJK_warning:n
_xeCJK_token_value_charcode:w	64, 2836
..... 274, 277, 289	_xeCJK_warning:nx
_xeCJK_ulem_Boundary_and_Default: 60, 65, 2145, 2702, 2815, 2898, 3289, 3290, 3291
..... 3724, 3769, 3769, 3777	_xeCJK_warning:nxx
_xeCJK_ulem_Boundary_and_FullLeft_glue:N	66, 2474, 2916, 3132, 3393, 3399
..... 3735, 3847, 3847, 3858	_xeCJK_warning:nxxx
_xeCJK_ulem_Boundary_and_NormalSp:	67, 2230
..... 3725, 3779, 3779, 3783	_xeCJK_zero_glue:
_xeCJK_ulem_CJK_and_Boundary_aux: 962, 964, 1038, 1046, 2994, 2995, 3000, 3001, 3854
..... 3730, 3785, 3785, 3793, 4024	_xunadd_add_accent:nnNN
_xeCJK_ulem_CJK_and_CJK:N	4614, 4622, 4627, 4627
..... 3723, 3801, 3801, 3810	_xunadd_begin_csname:n
_xeCJK_ulem_CJK_and_FullLeft_glue:N	4685, 4690, 4691
..... 3733, 3860, 3860, 3873	_xunadd_begin_hook:nn
_xeCJK_ulem_CJK_and_FullRight_glue:N	4521, 4608, 4687, 4687
..... 3734, 3896, 3896, 3904	_xunadd_chardef:Nn
_xeCJK_ulem_Default_and_FullLeft_glue:N	4550, 4552
..... 3731, 3835, 3835, 3845	_xunadd_chardef:cn
_xeCJK_ulem_Default_and_FullRight_glue:N	4545
..... 3732, 3875, 3875, 3894	_xunadd_check_slot:n
_xeCJK_ulem_FullLeft_and_CJK: ..	4499, 4527, 4527, 4547
..... 3720, 3924, 3924, 3931	_xunadd_declare_accent:NNNnnn
_xeCJK_ulem_FullLeft_and_Default:	4599, 4605
..... 3719, 3906, 3906, 3922	_xunadd_declare_accent:NNNxxn
_xeCJK_ulem_FullRight_and_CJK: ..	4595
..... 3722, 3947, 3947, 3961	_xunadd_declare_accent:Nnnn 4576, 4583, 4587, 4587, 4604
_xeCJK_ulem_FullRight_and_Default:	_xunadd_declare_accent:cnnn
..... 3721, 3933, 3933, 3945	4577, 4584
_xeCJK_ulem_between_CJK_blocks:nnN	_xunadd_declare_character:NNnn
..... 3744, 3746, 3750, 3818, 3818	4517, 4517, 4526
_xeCJK_ulem_cgglue:	_xunadd_declare_character:NNxn
..... 3806, 3823, 3867, 3882, 3956, 3985, 3991	4503
_xeCJK_ulem_class_group_begin:	_xunadd_declare_character:Nnn ..
..... 3807, 3812, 3812, 3871, 3902, 3929, 3959, 3981	4491, 4495, 4495, 4507
_xeCJK_ulem_fix_penalty:	_xunadd_declare_character:cnn
..... 3726, 3795, 3795, 3799	4492
_xeCJK_ulem_glue:n	_xunadd_declare_composite:Nnnn ..
..... 3699, 3701, 3985, 3985	4538, 4542, 4542, 4553
_xeCJK_ulem_hskip:n	_xunadd_declare_composite:cnnn
..... 3790, 3966, 3972, 3980, 3985, 3988, 3993	4539
_xeCJK_ulem_initial:	_xunadd_declare_composite_command:Nnnn
..... 3683, 3716, 3716 4562, 4566, 4566, 4572
_xeCJK_ulem_leader_type:	_xunadd_declare_composite_command:cnnn
..... 3678, 3714, 3715	4563
_xeCJK_ulem_punct_breakable_kern:n	_xunadd_end_csname:n
..... 3729, 3975, 3975, 3983	4686, 4696, 4697
_xeCJK_ulem_punct_hskip:n	_xunadd_end_hook:nn
..... 3728, 3963, 3963, 3967	4524, 4615, 4687, 4694
_xeCJK_ulem_punct_kern:n	_xunadd_glyph_if_exist:n
..... 3727, 3969, 3969, 3973	4457
_xeCJK_ulem_restore_CJK_and_Boundary: 4002, 4015, 4019	_xunadd_glyph_if_exist:nTF
_xeCJK_ulem_skip_punct_begin: 4457, 4522, 4621, 4631, 4639, 4642
..... 3680, 3711, 3711, 3840, 3852, 3866, 3880	_xunadd_glyph_if_exist:p:n
_xeCJK_ulem_skip_punct_end:	4457
..... 3681, 3711, 3713, 3918, 3942, 3957	_xunadd_if_csname:n
_xeCJK_ulem_swap_cs:NN 4464, 4471, 4478, 4490, 4537, 4561, 4575, 4582
..... 3718, 3755, 3759	_xunadd_provide_text_command_default:N
_xeCJK_under_CJKsymbol:N 4497, 4508, 4508
..... 4006, 4034, 4071, 4071	_xunadd_reload:N
_xeCJK_update_cs_case_tl:NNnn ..	4427, 4432
..... 1219, 1225, 1230, 1234	_xunadd_set_cmd_hook:nnn
_xeCJK_update_family:nn	4659, 4669, 4673, 4673
..... 2459, 3463, 3471	_xunadd_set_cmd_hook_aux:Nnn
_xeCJK_update_inline_env_case_tl:	4679, 4683
..... 1412, 1421, 1427, 1431, 1431	_xunadd_text_character:nN
_xeCJK_update_url_font:	4518, 4519
..... 3342, 3342, 3351	_xunadd_text_composite:nnNNn
_xeCJK_use_punct_dim:nn	4602, 4606, 4606
..... 1655, 1820, 1832,	_xunadd_text_composite_aux:NnNNn
1868, 1869, 1914, 1918, 1942, 2030, 2031, 2081, 2082, 2099	4617, 4626
_xeCJK_use_punct_dim:nnn	_xunadd_text_composite_aux:cnNNn
..... 916, 923, 928, 935, 936, 937, 942,	4611
1157, 1657, 1754, 1757, 1913, 1976, 1977, 1986, 1987,	_xunadd_tmp:w
1996, 1999, 2061, 2062, 2066, 2067, 2075, 2078, 2093, 2094	4412, 4415, 4434, 4446
_xeCJK_verb_CJK_and_Boundary:w	_xunadd_undeclare_composite:Nnnn ...
..... 3071, 3073	4472, 4473, 4476

A

\\add@accent	4643
\\addCJKfontfeature	2797, 2843
\\addCJKfontfeatures	6, 2788, 2792, 2797
\\AddEveryPageHook	99
\\addto@hook	3670
\\AfterEndPreamble	84
\\AfterPreamble	83
KaiMingPunct	4
KaiMingPunct+	4
KaiMingPunct-	4

FallBack	7	\c__xeCJK_NS_chars_clist	383, 383, 404
\AllowBreakBetweenPuncts	1572	\c__xeCJK_OP_chars_clist	361, 361, 372
Mapping	5	\c__xeCJK_package_ext_tl	13, 16, 36
\arraystretch	4088	\c__xeCJK_PO_chars_clist	400, 400, 408
\AssignTemplateKeys	1736	\c__xeCJK_PR_chars_clist	368, 368, 373
\AtBeginDocument	87	\c__xeCJK_punct_style_plain_tl	1741, 1773, 2126, 2135, 4124
\AtBeginUTFCommand	3425, 3554, 4651, 4651	\c__xeCJK_right_tl	948,
\AtEndOfPackage	589, 1542, 3166, 3610, 4418, 4554		958, 968, 969, 970, 975, 977, 978, 1079, 1090, 1100,
\AtEndPreamble	82		1113, 1132, 1155, 1163, 1196, 1197, 1198, 1204, 1570,
\AtEndUTFCommand	3555, 4651, 4661		1571, 1749, 1758, 1976, 1986, 1998, 2062, 2067, 2080,
\AutoFakeBold	2271, 2357		2089, 2094, 3887, 3913, 3937, 3940, 3941, 3951, 3954, 3955
\AutoFakeSlant	2271, 2357	\c_catcode_letter_token	1279, 1301
\AutoFallBack	2187	\c_catcode_other_space_tl	3160
B			
\begin	3274	\c_eleven	624, 671, 726, 1039, 1245
\bfdefault	2955, 2958	\c_four	156, 177, 3514
\BODY	4096, 4102, 4109	\c_fourteen	2997, 3004
\BoldFont	2352	\c_group_begin_token	584
\bool_gset_false:N	3204, 4132, 4325	\c_group_end_token	588
\bool_gset_true:N	2277, 2285, 3207, 3212, 4316	\c_math_toggle_token	828, 865, 1361
\bool_if:NF	102, 1256, 1451, 1474, 3017, 3028, 3673	\c_max_dim	1667, 1673, 1679, 1681, 1689, 1692, 1694,
\bool_if:nF	3414, 4254, 4510		1696, 1815, 1843, 1908, 1935, 1939, 1948, 2016, 2043, 2048
\bool_if:NT	871, 1460, 1954, 2484, 2496, 2920, 3153, 3286,	\c_minus_one	1032, 1038
	3425, 3654, 3655, 4001, 4008, 4010, 4014, 4323, 4425, 4429	\c_nan_fp	1680, 1693, 1695
\bool_if:nT	723, 875, 2817, 2825, 3350, 3592, 4275	\c_one	314, 348, 1027, 1067, 2088, 2775, 3514, 3853, 4208, 4246
\bool_if:NTF		\c_one_fp	1668, 1674, 1697
	747, 830, 835, 867, 1247, 1303, 1305, 1316, 1324, 1336,	\c_one_thousand	150, 204
	1341, 1789, 1824, 1837, 1889, 1904, 1922, 1968, 2009,	\c_seven	171, 186
	2012, 3048, 3677, 4214, 4225, 4237, 4252, 4273, 4293, 4304	\c_space_tl	806, 3775, 3782
\bool_if:nTF	123, 491, 621, 629, 637, 668, 676, 709,	\c_space_token	236, 263
	716, 732, 740, 837, 1030, 1041, 1051, 1287, 1330, 1961, 3563	\c_ten	719, 735
\bool_if:p:n	4483	\c_ten_thousand	70
\bool_new:N	57, 111, 254, 587, 818, 1448, 1600,	\c_thirteen	1053, 1062
	2267, 2268, 2403, 2404, 3024, 3085, 3218, 3704, 3710, 4331	\c_three	155, 176, 350, 2090, 3512, 3995, 4257, 4287
\bool_set_eq:NN	2395, 2396	\c_true_bool	3135
\bool_set_false:N	230,	\c_two	154, 164, 170, 349, 1821,
	805, 1249, 1462, 1582, 2366, 2380, 3709, 4228, 4296, 4307		1848, 1869, 3113, 3120, 3178, 3186, 3193, 4169, 4171,
\bool_set_true:N			4173, 4177, 4183, 4185, 4187, 4191, 4239, 4256, 4267, 4277
	104, 237, 585, 800, 810, 1244, 1453, 1577, 2362, 2369,	\c_two_hundred_fifty_five	351
	2376, 2383, 3030, 3675, 3707, 4128, 4218, 4244, 4265, 4285	\c_two_hundred_fifty_six	3636
\Boundary	347	\c__xeCJK_math_fam_int	2960, 2968, 3345
\box_new:N	55, 4018	\c__xeCJK_space_skip_tl	148, 148, 625, 672, 727
\box_set_to_last:N	1029	\c_zero	69, 315, 347, 1054, 1064,
\box_use:N	4066, 4073		1068, 1069, 1663, 1664, 1700, 2968, 3015, 3157, 3477, 4157
\box_use_clear:N	1035, 1036	\c_zero_dim	917, 918, 924, 925,
\box_wd:N	1033, 4065, 4099		1686, 1698, 1743, 1744, 1745, 1746, 1747, 1748, 1749,
C			
\c__xeCJK_CJ_chars_clist	391, 391, 407		1775, 1776, 1864, 1877, 2106, 3114, 3257, 4060, 4090, 4091
\c__xeCJK_CJK_chars_clist	410, 410, 538, 2962	\c_zero_fp	1682, 1938, 2019, 2046
\c__xeCJK_CL_chars_clist	375, 375, 403	\c_zero_skip	152, 161, 166, 183, 965, 3061, 3790
\c__xeCJK_config_ext_tl	3657, 3659	\char_set_catcode:nn	3605, 4428
\c__xeCJK_encoding_tl ..	2642, 2952, 2955, 2957, 3285, 3285	\char_set_catcode_active:N	4377
\c__xeCJK_EX_chars_clist	388, 388, 405	\char_set_catcode_ignore:n	316
\c__xeCJK_FullLeft_chars_clist	370, 370, 533, 2962	\char_set_catcode_letter:n	505, 3603, 4426
\c__xeCJK_FullRight_chars_clist	401, 401, 534, 2964	\char_set_lccode:nn	3647, 4499, 4590, 4591
\c__xeCJK_HalfLeft_chars_clist	356, 356, 531	\char_value_catcode:n	3605, 4428
\c__xeCJK_HalfRight_chars_clist	356, 358, 532	\CheckFullRight	1170
\c__xeCJK_IS_chars_clist	390, 390, 406	\CheckSingle	1258
\c__xeCJK_IVS_chars_clist	446, 446, 540	\CJK	347
\c__xeCJK_left_tl	950,	\CJK@@UL	3705, 3705, 3708
	960, 982, 989, 994, 995, 999, 1002, 1008, 1009, 1013,	\CJK@hundredmillion	3641
	1015, 1020, 1074, 1075, 1114, 1123, 1141, 1158, 1570,	\CJK@ifundefined	3596, 3597
	1570, 1748, 1758, 1977, 1987, 2001, 2061, 2066, 2077,	\CJK@nest	4054
	2087, 2093, 3841, 3842, 3855, 3868, 3869, 3888, 3912, 3915	\CJK@postUnderdot	4013, 4054
\c__xeCJK_math_family_tl	2951, 2953, 2955, 2958	\CJK@preUnderdot	4003
\c__xeCJK_math_tl ..	2938, 2939, 2942, 2943, 2952, 2957, 2960	\CJK@tenthousand	3640
\c__xeCJK_NormalSpace_chars_clist	356, 360, 539	\CJK@UL	3705, 3705, 3706
		\CJK@underdotBox	4031, 4066
		\CJK@underdotSkip	4004
		\CJK@UnicodeEnc	3632
		\CJKaddEncHook	3630, 3630

\CJkecglue	617, 634, 642, 662, 681, 721, 738, 790, 794, 801, 811, 813, 832, 868, 2995, 3001, 3037, 3053, 3064, 3068, 3070, 3074, 3696, 3700
xCJkecglue	3
\CJKf@global	4053
\CJKfamily	6, 2655, 2655, 2687
\CJKfamilydefault	6, 2228, 2236, 2626, 2627, 2856, 2859, 2883, 2885, 2893, 2898, 2900, 2902, 2903, 2907, 2913, 2917, 2919, 2945, 2946, 3299, 3308, 3309, 3313, 3468
\CJKfilltwosides	4076
\CJKfixedspacing	3086
\CJKfontspec	6, 2755, 2763
\CJKglue	711, 714, 748, 781, 785, 885, 971, 1007, 1108, 2994, 3000, 3036, 3052, 3063, 3067, 3689, 3698, 4079, 4087, 4327
\CJKmath	2936
\CJKnospace	3327, 3328
\CJKpunctsymbol	990, 1003, 1021, 1127, 1145, 1193, 3253, 3254
\CJKrmdefault	6, 2717, 2728, 2736, 2889, 2903, 2906, 2907, 3299, 3299, 3310, 3465
\CJKsetecglue	3325, 3325, 3326
\CJKsfdefault	6, 2718, 2729, 2742, 2890, 3299, 3300, 3466
\CJKspace	819, 3327, 3327
\CJKsymbol	610, 705, 885, 887, 889, 1511, 1512, 2192, 2194, 2195, 2200, 2201, 3253, 3253, 3808, 3827, 3832, 4005, 4006, 4011, 4033, 4034, 4046
\CJKttdefault	6, 2719, 2730, 2747, 2891, 3299, 3301, 3467
\CJKunderanysymbol	4028, 4028
\CJKunderdot	3999, 3999
\clist_clear:N	2322, 2394, 2436, 2807
\clist_concat:NNN	2414, 2828, 2831, 2867, 2961, 2963
\clist_const:Nn	356, 358, 360, 361, 368, 375, 383, 388, 390, 391, 400, 410, 446
\clist_const:Nx	370, 401
\clist_gconcat:ccN	473
\clist_gconcat:NNN	487
\clist_get:NNF	4402
\clist_gput_left:Nv	4396
\clist_gput_left:Nx	4399
\clist_gput_right:Nx	4441
\clist_gremove_duplicates:N	4447
\clist_gset:Nn	1602, 2790
\clist_map_function:nN	463
\clist_map_inline:cn	521, 3082
\clist_map_inline:Nn	475, 1605, 1642, 2965, 4436
\clist_map_inline:nn	27, 32, 602, 614, 649, 892, 908, 910, 1416, 1425, 1469, 1527, 2256, 2808
\clist_new:c	325, 336
\clist_new:N	59, 1601, 2348, 2402, 2428, 2788, 2838, 2839, 4394, 4449
\clist_put_left:Nn	2260
\clist_put_right:Nn	2441
\clist_put_right:Nx	2443, 2486, 2491, 2498, 2503, 2869
\clist_remove_all:Nn	2308, 2542, 2805
\clist_remove_duplicates:N	4422
\clist_set:Nn	472, 2343, 2412, 2803
\clist_set:Nx	4421
\clist_set_eq:NN	4407
\cs:w	141, 3594
\cs_end:	141, 295, 1647, 3594
\cs_generate_variant:Nn	300, 487, 488, 546, 556, 562, 1207, 1382, 1392, 1480, 1497, 1566, 1784, 2111, 2349, 2350, 2429, 2449, 2477, 2580, 2647, 2706, 2785, 2786, 2840, 2878, 2879, 4507, 4526, 4552, 4553, 4572, 4604, 4605, 4626
\cs_gset_eq:cc	2575
\cs_gset_eq:NN	130, 140, 3167, 3452, 3453, 4005
\cs_gset_nopar:Npx	92
\cs_gset_protected_nopar:cpx	2456, 2546, 2640
\cs_gset_protected_nopar:Npx	3355, 3383, 3389
\cs_if_eq:NNF	1175, 1263, 2192
\cs_if_eq:NNT	1185, 1271, 2200
\cs_if_eq:NNTF	3387, 3609, 4373
\cs_if_exist:cTF	2625, 2650, 4438, 4609
\cs_if_exist:NF	1533, 3596
\cs_if_exist_p:c	4512
\cs_if_exist_p:N	3350, 3593
\cs_if_exist_use:cF	2586, 2595, 2617, 3170, 4523, 4690, 4691, 4696, 4697
\cs_if_exist_use:cTF	1791, 1891, 2652, 4634, 4645
\cs_if_free:cF	2954
\cs_if_free:NF	3353
\cs_if_free:NFT	3597
\cs_if_free_p:c	4513
\cs_new:Npn	116, 121, 4683
\cs_new_eq:cN	341
\cs_new_eq:NN	286, 291, 588, 940, 1119, 1152, 1160, 2603, 2739, 2797, 3086, 3303, 3326, 3444, 3445, 3471, 3472, 3546, 3547, 3560
\cs_new_nopar:Npn	112, 195, 271, 273, 277, 289, 298, 340, 464, 547, 1193, 1603, 1604, 1651, 1653, 1655, 1657, 1933, 1946, 1992, 2004, 2041, 2056, 2070, 2112, 2583, 2634, 2635, 2636, 2713, 2724, 3253, 3254, 3761, 4161, 4527, 4685, 4686
\cs_new_protected:Npn	74, 76, 78, 108, 127, 129, 131, 219, 221, 225, 255, 857, 1210, 1234, 1431, 3630, 4041, 4057, 4332, 4347, 4360, 4370, 4372, 4432, 4476, 4495, 4508, 4517, 4519, 4542, 4550, 4566, 4587, 4599, 4606, 4617, 4627, 4673, 4687, 4694
\cs_new_protected_nopar:Npn	60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 88, 90, 114, 138, 205, 234, 261, 319, 330, 458, 470, 489, 508, 518, 544, 549, 551, 557, 563, 569, 578, 580, 582, 619, 666, 707, 762, 768, 773, 826, 855, 863, 885, 906, 913, 920, 927, 929, 931, 941, 944, 954, 964, 966, 973, 980, 992, 997, 1005, 1011, 1023, 1077, 1088, 1098, 1105, 1120, 1129, 1139, 1147, 1153, 1161, 1194, 1208, 1277, 1299, 1349, 1359, 1368, 1383, 1467, 1487, 1498, 1607, 1618, 1629, 1659, 1661, 1737, 1769, 1785, 1885, 1957, 2085, 2206, 2217, 2249, 2297, 2319, 2334, 2388, 2407, 2430, 2432, 2450, 2452, 2454, 2468, 2480, 2510, 2517, 2530, 2561, 2584, 2593, 2604, 2615, 2637, 2676, 2686, 2689, 2695, 2768, 2779, 2798, 2846, 2940, 2972, 2990, 3026, 3073, 3075, 3077, 3079, 3087, 3110, 3129, 3151, 3168, 3174, 3181, 3342, 3365, 3367, 3377, 3475, 3521, 3644, 3671, 3711, 3713, 3716, 3755, 3769, 3779, 3785, 3795, 3801, 3812, 3818, 3835, 3847, 3860, 3875, 3896, 3906, 3924, 3933, 3947, 3963, 3969, 3975, 3985, 3991, 3993, 4019, 4071, 4129, 4139, 4166, 4180, 4194, 4200, 4206, 4211, 4222, 4234, 4249, 4270, 4313, 4320, 4354
\cs_new_protected_nopar:Npx	2760
\cs_set:Npn	4367
\cs_set_eq:NN	128, 133, 134, 135, 227, 801, 806, 813, 861, 1177, 1178, 1179, 1180, 1187, 1188, 1265, 1266, 1272, 1578, 1583, 2194, 2195, 2201, 2459, 2687, 2882, 2994, 2995, 3000, 3001, 3036, 3037, 3038, 3039, 3052, 3053, 3054, 3055, 3067, 3068, 3070, 3071, 3597, 3637, 3678, 3686, 3687, 3688, 3693, 3694, 3695, 3705, 3714, 4006, 4011, 4033, 4034, 4046, 4053, 4079, 4087, 4136, 4231, 4327, 4346, 4353, 4359, 4412, 4413, 4415, 4434, 4435, 4446
\cs_set_nopar:Npn	3561, 3635
\cs_set_protected:cpn	4568, 4675
\cs_set_protected_nopar:Npn	785, 794, 811, 3698, 3700, 4291, 4302
\cs_set_protected_nopar:Npx	3010, 3040
\cs_to_str:N	2759, 3366
\cs_undefine:c	1610, 1633, 2473, 4477
\cs_undefine:N	136, 2896, 3456, 3457
\curr@fontshape	3089, 3092, 3095, 3125, 3170, 3176

\CurrentOption	3664, 4116, 4396	1292, 1294, 2323, 3103, 3104, 3105, 3177, 3178, 3344, 3345, 3499, 3500, 3501, 3502, 3594, 3764, 3766, 4356, 4357
D		
\DeclareEncodedCompositeAccents	4580, 4580	\exp_args:Nc 324, 1644, 4466, 4473
\DeclareEncodedCompositeCharacter	4573, 4573	\exp_args:NNc 3366
\DeclareExpandableDocumentCommand	2688	\exp_args:NNv 4515
\DeclareInstance	2147	\exp_args:Nnx 2147, 2159
\DeclareObjectType	1663	\exp_args:No 1290, 3381
\DeclareOption	3664, 4116, 4395	\exp_args:NV 115
\DeclareSymbolFont	2952	\exp_args:Nx 3369
\DeclareSymbolFontAlphabet	3361	\exp_last_unbraced:Nf 118
\DeclareTemplateCode	1700	\exp_not:c 895, 3047
\DeclareTemplateInterface	1664	\exp_not:N 243, 2463, 2551, 2552, 2556, 2642, 2643, 2644, 2889, 2890, 2891, 3306, 3310, 4078, 4427
\DeclareTextCommand	4518, 4601	\exp_not:n 554, 560, 2459, 2460, 2882, 3050
\DeclareUTFcharacter	4488, 4488	\exp_not:o 95, 2444, 3011, 3357, 3390
\DeclareUTFccomposite	4535, 4535	\exp_not:V 567, 575, 2330, 2331, 2444, 2461, 2462, 2492, 2504, 2873, 2874, 3309
\DeclareUTFCompositeCommand	4556, 4557, 4559, 4559	\exp_stop_f: 145, 212, 215, 304, 930, 942, 2115
\def	3637	\ExplSyntaxOff 4425
\Default	347	\ExplSyntaxOn 4429
\defaultCJKfontfeatures	6, 2788, 2789, 2791, 3281	\extrarowheight 4089, 4090
\dim_compare:nNnF	3186	
\dim_compare:nNnTF	1815, 1843, 1908, 1935, 1948, 2016, 2043, 2048, 3114, 4091, 4099	F
\dim_compare_p:nNn	1033	\f@baselineskip 3196
\dim_const:cn	765	\f@encoding 3565
\dim_eval:n	197, 1660, 1662, 1994, 2072, 3190, 4078, 4101	\f@family 2643, 3146, 3147, 3159, 3184, 3462
\dim_gset:cn	765	\f@series 2583
\dim_if_exist:cTF	764	\f@shape 2583
\dim_max:nn	1812, 1826, 1864, 1877, 1906, 2006, 2013, 2058, 3444, 3452	\f@size 2583, 3089, 3092, 3125, 3170, 3176, 3192
\dim_min:nn	1827, 1838, 1923, 1955, 2060, 2065, 3445, 3453	\Fallback 2351
\dim_new:N	56, 1767, 1768	\familydefault 2887, 3468
\dim_ratio:nn	3193	\fi: 146, 216, 217, 222, 249, 250, 269, 296, 305, 760, 1640, 1648, 3006, 3767, 4137, 4217, 4230, 4243, 4264, 4284, 4298, 4299, 4309, 4310, 4317, 4329, 4340, 4357, 4460, 4484
\dim_set:Nn	1753, 1755, 1787, 1810, 1887, 1902, 1959, 2091	\file_input:n 3604, 4442
\dim_set_eq:NN	4090	\fix@penalty 3472, 3474, 3798
\dim_to_fp:n	3120, 3121	\fontencoding 2642, 3183
\dim_use:N	2114, 3105, 3178	\fontfamily 3460, 3460
\document	92, 95	\fontsize 3188
E		
xeCJKactive	3	\fontspec_set_family:Nnn 2460
\EditInstance	2159	\fontspec_setup_maths: 3353, 3353, 3355, 3359
\else:	146, 214, 216, 241, 246, 267, 296, 305, 760, 1640, 1648, 3765, 4217, 4243, 4264, 4284, 4340, 4460, 4484	\fontspec_visible_space: 3167
\EmboldenFactor	2271	\footnote 3275
\encodingdefault	4454	\footnotemark 3275
\end	3274	\fp_compare:nNnTF 275, 1938, 2019, 2046
\endmath	3365, 3412	\fp_eval:n 198, 3133, 3134
\endminipage	4082	\fp_gset:Nn 2278, 2286
\endtabular	4097, 4103, 4110	\fp_new:N 2269, 2270, 2405, 2406, 3149
\ensuremath	3365, 3387, 3413	\fp_set:Nn 2370, 2384, 3131
\EnvCS	1400	\fp_set_eq:NN 2363, 2377, 2397, 2398
Verb	5	\fp_use:c 1941, 1951, 2052
\etex_currentgrouplevel:D	3015	\fp_use:N 1846, 2028, 2050, 2487, 2499, 3136
\etex_currentgrouptype:D	2997, 3004	\FullLeft 347
\etex_dimexpr:D	1847, 1942, 1952, 2029, 2053	\FullRight 347
\etex_fontcharwd:D	2097, 3105	G
\etex_glueshrink:D	191	\g__xeCJK_after_end_preamble_hook_tl 73, 79, 84, 91
\etex_gluestretch:D	190	\g__xeCJK_after_preamble_hook_tl 72, 77, 83, 87
\etex_iffontchar:D	145, 4459	\g__xeCJK_at_end_preamble_hook_tl 71, 75, 82, 89
\etex_lastnodetype:D	624, 671, 719, 726, 735, 1025, 1032, 1053, 1067, 1245, 3853	\g__xeCJK_auto_fake_bold_bool 2267, 2274, 2277, 2395
\etex_numexpr:D	4459, 4551	\g__xeCJK_auto_fake_slant_bool 2268, 2282, 2285, 2396
NewLineCS	3	\g__xeCJK_base_class_seq 452, 452, 453, 1500
NewLineCS+	3	\g__xeCJK_CJK_class_seq 452, 457, 460, 579, 3524
NewLineCS-	3	\g__xeCJK_CJK_sub_class_seq 1440, 1440, 1516, 1525, 3737, 3739, 4149, 4174, 4188
\exp_after:wN	140, 141, 207, 208, 222, 238, 239, 244, 245, 247, 248, 264, 265, 268, 274, 1291,	\g__xeCJK_class_seq 317, 317, 326, 337, 453, 591
		\g__xeCJK_config_bool 3204, 3207, 3212, 3218, 3655
		\g__xeCJK_config_name_tl 3208, 3213, 3217, 3659
		\g__xeCJK_default_features_clist 2415, 2788, 2790

\g__xeCJK_embolden_factor_fp	. 2269, 2278, 2289, 2363, 2397
\g__xeCJK_family_font_name_prop 2244, 2253, 2470, 2507, 2508, 2512, 2533, 2536, 2568, 2697, 2800, 2848, 2855, 2897, 2910
\g__xeCJK_family_font_options_prop 2507, 2509, 2514, 2539, 2543, 2569, 2822, 2851, 2858, 2862
\g__xeCJK_family_int 2390, 2400, 2775
\g__xeCJK_family_name_prop 2472, 2507, 2507, 2551, 2567, 2639, 2949
\g__xeCJK_fontspec_prop 2770, 2781, 2787
\g__xeCJK_indent_bool 3222, 3286
\g__xeCJK_last_punct_tl	943, 943, 946, 948, 950, 956, 958, 960, 968, 969, 970, 975, 977, 978, 988, 1001, 1019, 1085, 1095, 1102, 1124, 1125, 1126, 1133, 1134, 1135, 1142, 1143, 1144, 1164, 1165, 1166, 1196, 1197, 1198, 1204, 3910, 3912, 3913, 3915, 3937, 3940, 3941, 3951, 3954, 3955
\g__xeCJK_listings_IVS_bool	. 4132, 4316, 4323, 4325, 4331
\g__xeCJK_math_bool 2920, 2936, 3350
\g__xeCJK_new_class_seq 317, 318, 327, 3512, 3516
\g__xeCJK_non_CJK_class_seq 452, 454, 455, 890
\g__xeCJK_number_bool 3221, 3654
\g__xeCJK_punct_bound_width_tl 1596, 1894, 1896
\g__xeCJK_punct_style_seq 1549, 2140, 2146, 2149
\g__xeCJK_punct_width_tl 1595, 1794, 1796
\g__xeCJK_scale_family_prop 3096, 3137, 3150
\g__xeCJK_slant_factor_fp	... 2270, 2286, 2290, 2377, 2398
\g__xeCJK_spacefactor_int 150, 163, 180, 198, 201, 203, 204, 653, 690
\g__xeCJK_special_punct_clist 1601, 1602, 1605, 1642
\g__xeCJK_sub_key_seq	.. 2296, 2296, 2299, 2804, 2810, 2820
\g__xeCJK_under_symbol_box 4018, 4060, 4073
\g__xeCJK_unknown_family_seq 2699, 2701, 2707
\g__xunadd_encname_clist 4394, 4396, 4399, 4402, 4407, 4418, 4441, 4447
\g_fontspec_bfmathrm_tl 3360
\g_fontspec_encoding_tl 3183, 3285
\group_align_safe_begin: 231, 258, 1199, 1282, 1312, 1370, 1385
\group_align_safe_end: 228, 229, 257, 1202, 1203, 1284, 1286, 1315, 1323, 1329, 1373, 1388, 1389, 1395, 1402
\group_begin: 89, 2251, 2409, 2458, 3103, 3344, 3546, 3634, 3646, 4030, 4366, 4498, 4589
\group_end: 89, 2264, 2424, 2464, 3103, 3344, 3547, 3639, 3650, 4038, 4369, 4502, 4594
H	
\HalfLeft 352
\HalfRight 352
\hbox_overlap_right:n 4073
\hbox_set:Nn 207, 4031, 4059, 4093
\hbox_to_zero:n 4063
\hbox_unpack:N 4105
CheckFullRight 5
CheckSingle 3
\hskip 3688, 3695
I	
\icprotect 3593, 3594
MiddlePunct 4
MiddlePunct+ 4
MiddlePunct- 4
\if_case:w 212, 215, 2992
\if_catcode:w 242
\if_cs_exist:w 295, 1647
\if_dim:w 759
\if_false: 4357
\if_int_compare:w 220, 303, 1639
\if_meaning:w 236, 263, 3763
\if_predicate:w 4482
J	
CJKecglue 3
CJKglue 3
CJKmath 3
CJKspace 3
K	
\KaiMingPunct 1572
\keys_define:nn 307, 781, 790, 819, 1170, 1214, 1258, 1407, 1438, 1544, 1572, 2117, 2187, 2271, 2300, 2352, 2357, 2936, 2983, 3200, 3219, 3225, 3287
\keys_set:nn 3255, 3316
\keys_set_known:nVN 2417
\keyval_parse:NNn 2449
\keyval_parse:NNV 2435
\KeyValue 1669, 1670, 1671, 1672, 1675, 1676, 1677, 1678
L	
\l__kernel_expl_bool 4425, 4429
\l__keys_module_tl 3250
\l__peek_search_token 227, 243
\l__xeCJK_add_block_features_clist	2807, 2832, 2839, 2869
\l__xeCJK_add_font_features_clist 2803, 2805, 2829, 2838, 2868
\l__xeCJK_add_min_bound_to_margin_bool 1720, 1954
\l__xeCJK_auto_fake_bold_bool 2362, 2366, 2369, 2395, 2403, 2484

\l__xeCJK_auto_fake_slant_bool	1394
..... 2376, 2380, 2383, 2396, 2404, 2496	
\l__xeCJK_begin_int	1201, 1241
..... 223, 501,	
503, 505, 506, 510, 512, 514, 515, 2974, 2975, 2978, 2979	
\l__xeCJK_bound_dim	1241
..... 1753, 1767, 1827, 1838, 1848, 1872, 1923, 1931, 1952, 1955	
\l__xeCJK_bound_margin_ratio_fp	1727, 2012
..... 1718	
\l__xeCJK_bound_margin_width_dim	1721, 1824, 1837, 1922
..... 1717	
\l__xeCJK_bound_punct_ratio_fp	230,
..... 1716	237, 254, 830, 835, 867, 871, 1303, 1316, 1324, 1336, 1341
\l__xeCJK_bound_punct_width_dim	1332, 1439
..... 1715	
\l__xeCJK_ccglue_skip 786, 789, 3690, 3699, 3830, 3992	1577, 1582, 1600, 4258
\l__xeCJK_check_single_cs_case_tl 1372, 1381, 1397, 1404	747, 821, 840, 878, 1305
\l__xeCJK_CJK_group_bool	\l__xeCJK_restore_listings_toks_tl
..... 585, 587, 3153, 3425 4125, 4133, 4134, 4141, 4151, 4160
\l__xeCJK_current_coor_tl	\l__xeCJK_reverse_bound_dim
..... 2588, 2590, 1755, 1768, 1827, 1831, 1848, 1881, 1917, 1955
2597, 2599, 2602, 2619, 2631, 3095, 3097, 3138, 3176, 3179	\l__xeCJK_same_align_margin_dim
\l__xeCJK_different_align_margin_dim 1728
..... 1730	\l__xeCJK_same_align_ratio_fp
\l__xeCJK_different_align_ratio_fp 1729
..... 1731	\l__xeCJK_scale_factor_fp
\l__xeCJK_ecglue_skip .. 728, 795, 812, 817, 3697, 3701, 3774 3131, 3133, 3136, 3149
\l__xeCJK_embolden_factor_fp . 2363, 2370, 2397, 2405, 2487	\l__xeCJK_shipout_hook_bool
\l__xeCJK_enabled_global_setting_bool 102, 104, 111
..... 1702, 1789, 1889, 1963	\l__xeCJK_shipout_hook_tl
\l__xeCJK_enabled_hanging_bool 105, 109, 110
..... 1719, 1904	\l__xeCJK_slant_factor_fp 2377, 2384, 2398, 2406, 2499
\l__xeCJK_enabled_kerning_bool	\l__xeCJK_sub_cancel_bool 1448, 1451, 1453, 1460, 1462, 1474
..... 1723, 1968	\l__xeCJK_sub_family_name_tl
\l__xeCJK_end_int 224, 502, 503, 510, 512, 2974, 2975 2346, 2521, 2525, 2532, 2537, 2544, 2547, 2555
\l__xeCJK_env_cs_case_tl	\l__xeCJK_sub_font_name_tl
..... 1398, 1401, 1405 2262, 2325, 2326, 2327, 2331, 2337,
\l__xeCJK_env_cs_seq	2339, 2341, 2342, 2347, 2534, 2537, 2849, 2856, 2864, 2874
..... 1401	\l__xeCJK_sub_font_options_clist
\l__xeCJK_fallback_family_tl 2260, 2262, 2322, 2330, 2343,
..... 2213, 2222, 2225, 2231, 2252, 2254, 2258, 2259, 2261, 2266	2348, 2540, 2542, 2544, 2852, 2859, 2863, 2867, 2868, 2873
\l__xeCJK_family_default_init_tl . 2883, 3302, 3304, 3313	\l__xeCJK_sub_key_prop . 2306, 2312, 2328, 2391, 2401, 2519
\l__xeCJK_family_name_tl	\l__xeCJK_sub_key_seq
..... 2307, 2411, 2806, 2812, 2817, 2826
2420, 2421, 2426, 2513, 2515, 2521, 2534, 2540, 2549, 2552	\l__xeCJK_tmp_bool
\l__xeCJK_fixed_margin_ratio_fp 57, 1244, 1247, 1249, 1256
..... 1710	\l__xeCJK_tmp_box
\l__xeCJK_fixed_margin_width_dim 55,
..... 1709	207, 1029, 1033, 1035, 1036, 4059, 4065, 4093, 4099, 4105
\l__xeCJK_fixed_punct_ratio_fp	\l__xeCJK_tmp_clist 59, 472, 474, 475, 2961, 2963, 2964, 2965
..... 1704	\l__xeCJK_tmp_dim
\l__xeCJK_fixed_punct_width_dim 56, 1787, 1810, 1815, 1820,
..... 1703	1831, 1861, 1870, 1872, 1880, 1881, 1887, 1902, 1908,
\l__xeCJK_fntef_shipout_tl	1912, 1917, 1929, 1931, 1959, 1975, 1985, 2091, 2099, 2107
..... 4012, 4043, 4049, 4056	\l__xeCJK_tmp_int 54, 341, 511, 514, 520, 522, 3081, 3083
\l__xeCJK_font_name_bf_tl	\l__xeCJK_tmp_skip
..... 2354, 2392, 2482, 2492 58, 1250, 1252, 1256
\l__xeCJK_font_name_it_tl	\l__xeCJK_tmp_tl
..... 2355, 2393, 2494, 2504 53, 565,
\l__xeCJK_font_name_tl	566, 567, 571, 572, 574, 575, 2321, 2323, 2337, 2342,
..... 2254,	2439, 2440, 2444, 2470, 2472, 2474, 2572, 2573, 2759,
2255, 2326, 2327, 2413, 2427, 2462, 2513, 2801, 2834, 2864	2760, 2761, 2902, 2917, 2943, 2946, 2949, 2951, 4349, 4351
\l__xeCJK_font_options_clist .. 2308, 2412, 2414, 2415,	\l__xeCJK_ulem_hook_used_bool
2416, 2418, 2428, 2515, 2823, 2828, 2829, 2831, 2832, 2834 3673, 3675, 3704, 4001, 4008, 4010, 4014
\l__xeCJK_font_options_prop . 2434, 2437, 2448, 2451, 2453	\l__xeCJK_ulem_skip_punct_bool 3677, 3707, 3709, 3710
\l__xeCJK_fontspec_family_tl . 2460, 2463, 2467, 2552, 2556	\l__xeCJK_underdot_shipout_tl 4036, 4045, 4050, 4051
\l__xeCJK_fontspec_options_clist	\l__xeCJK_verb_addon_bool
..... 2394, 2402, 2418, 2461, 2486, 2491, 2498, 2503 3028, 3030, 3085
\l__xeCJK_inline_env_case_tl 1387, 1433, 1435, 1437	\l__xeCJK_verb_case_int
\l__xeCJK_inline_env_seq 1411, 1418, 1419, 1426, 1430, 1434 2982, 2987, 2992
\l__xeCJK_kerning_margin_minimum_dim	\l__xeCJK_verb_exspace_skip
..... 1734, 2007 3061, 3076, 3078, 3091, 3100, 3109, 3112, 3114, 3116, 3126
\l__xeCJK_kerning_margin_ratio_fp	\l__xeCJK_xecglue_bool
..... 1733, 2050 623, 670, 725, 800, 805, 810, 818, 3048, 3049
\l__xeCJK_kerning_margin_width_dim 1732, 2048, 2049	\l__xunadd_begin_hook_tl
\l__xeCJK_kerning_total_ratio_fp 1726, 2019, 2028 4656, 4657, 4671, 4689
\l__xeCJK_kerning_total_width_dim 1725, 2016, 2017	\l__xunadd_encname_clist
\l__xeCJK_listings_env_bool 4421, 4422, 4427, 4449
..... 3017, 3024, 4128	\l__xunadd_end_hook_tl
\l__xeCJK_listings_flag_int 4666, 4667, 4672, 4699
..... 4239, 4246, 4256, 4257, 4267, 4277, 4287, 4290, 4299, 4310	\l_keys_choice_int
\l__xeCJK_listings_letter_bool 4214, 4218, 4225, 4228, 2987
4237, 4244, 4252, 4265, 4273, 4285, 4293, 4296, 4304, 4307	\l_keys_choice_tl
\l__xeCJK_margin_minimum_dim 2274, 2282
..... 1722, 1813, 1906	\l_keys_key_tl
\l__xeCJK_middle_margin_ratio_fp 3244, 3245, 3289, 3290, 3291
..... 1714, 1846	\l_keys_value_tl
\l__xeCJK_middle_margin_width_dim 1713, 1843, 1844 1547, 2129, 2130, 2131, 2278, 2286, 2370, 2384, 3213
\l__xeCJK_middle_punct_ratio_fp	\l_peek_token
..... 1708 236, 243, 263, 657, 694, 839, 840, 849, 877, 878, 1200,
\l__xeCJK_middle_punct_width_dim	1283, 1289, 1290, 1291, 1292, 1313, 1321, 1333, 1353, 1371
..... 1707	\l__xeCJK_current_font_tl
\l__xeCJK_min_bound_to_kerning_bool 1652, 1654, 2581, 2581, 2582, 2586, 2588
..... 1724, 2009	\l__xeCJK_family_tl . 2212, 2228, 2236, 2237, 2582, 2589,
\l__xeCJK_mixed_margin_ratio_fp	2617, 2620, 2621, 2623, 2627, 2630, 2660, 2667, 2668,
..... 1712	
\l__xeCJK_mixed_margin_width_dim	
..... 1711	
\l__xeCJK_mixed_punct_ratio_fp	
..... 1706	
\l__xeCJK_mixed_punct_width_dim	
..... 1705	
\l__xeCJK_new_line_cs_case_tl	
..... 1394, 1398, 1405	

2680, 2681, 2685, 2685, 2771, 2772, 2801, 2823, 2849, 2852, 2863, 3089, 3092, 3095, 3097, 3099, 3125, 3138, 3144	3013, 3025, 3314, 3319, 3321, 3323, 3324, 3325, 3327, 3328, 3329, 3331, 3333, 3335, 3337, 3598, 4559, 4651, 4661
\l_xeCJK_punct_style_tl 1652, 1654, 1741, 1761, 1773, 1779, 2126, 2130, 2134, 4124	\NewEnviron 4085 \NewLineCS 1393
\LA@space 3763	\newXeTeXintercharclass 324
\labelsep 1048, 1056, 1057	InlineEnv 4
PlainEquation 3	InlineEnv+ 4
SlantFactor 4	InlineEnv- 4
\leavevmode 3424	\nobreak 3275, 3424
AllowBreakBetweenPuncts 5	\NoBreakCS 1240
\LocalConfig 3200	\nobreakspace 3423, 3423, 3424
\LongPunct 1572	\normalfont 2737, 2743, 2748
\lst@AddToHook 4120, 4121, 4122, 4128	\NormalSpace 352
\lst@Append 4209, 4300, 4311	\normalspacedchars 9, 524, 524
\lst@AppendLetter 4291, 4291	EnvCS 3
\lst@AppendOther 4291, 4302	EnvCS+ 3
\lst@arg 4351, 4362	EnvCS- 3
\lst@FillOutputBox 4326	
\lst@ifbreaklines 4135, 4299, 4310	O
\lst@ifec 4340	NoBreakCS 5
\lst@ifflexible 4315, 4322	NoBreakCS+ 5
\lst@ifletter 4217, 4230, 4243, 4264, 4284, 4298, 4309	NoBreakCS- 5
\lst@ifNextCharActive 4346	LocalConfig 2
\lst@InlineGJ 4359	LongPunct 4
\lst@InlineGJEnd 4364	LongPunct+ 4
\lst@InsideConvert@ 4353	LongPunct- 4
\lst@lastother 4231, 4311	\or: 2993, 2996, 3003
\lst@length 4208	
\lst@letterfalse 4227, 4230, 4306, 4309	P
\lst@lettertrue 4215, 4217, 4240, 4243, 4261, 4264, 4281, 4284, 4295, 4298	\par 3273
\lst@numberstyle 4133	\PassOptionsToPackage .. 3232, 3239, 3245, 3663, 3664, 4116
\lst@Output 4217, 4227, 4230, 4239, 4243, 4260, 4264, 4280, 4284, 4295, 4306, 4309	\pdfstringdefDisableCommands 3581
\lst@OutputOther 4217, 4243, 4264, 4284, 4298	\pdfTeX_if_engine:T 4383
\lst@ProcessLetter 4197	\pdfTeX_strcmp:D 212, 215
\lst@ProcessOther 4198	\peek_after:Nw 232, 238, 259, 264
\lst@whitespacefalse 4213, 4224, 4236, 4251, 4272	\peek_catcode:NTF 1279, 1361
	\peek_meaning_remove:NTF 654, 691
M	\penalty 3687, 3694
\makeXeCJKactive 310, 314, 314	\Pifont 3573
\makeXeCJKinactive 311, 314, 315, 3554, 3559, 3568, 3574, 3584	\PlainEquation 1438
\math 3365, 3411	\prg_do_nothing: 128, 130, 550, 4413
\mathrm 3353, 3361	\prg_new_conditional:Npnn 11, 143, 210, 293, 301, 757, 1637, 1645, 4457, 4478
EmboldenFactor 4	\prg_new_protected_conditional:Npnn 2648
\mdefault 2953	\prg_return_false: 14, 146, 216, 296, 305, 760, 1640, 1648, 2653, 4460, 4484, 4486
\MiddlePunct 1572	\prg_return_true: 14, 146, 213, 216, 296, 305, 760, 1640, 1648, 2653, 4460, 4484
\minipage 4078	\ProcessKeysOptions 3282
\msg_critical:nn 10, 4391	\ProcessOptions 3665, 4117, 4397
\msg_critical:nnn 47	\prop_clear:N 2306, 2391, 2434
\msg_error:nn 62	\prop_get:Nn 2244
\msg_error:nnn 30, 35, 50, 4444	\prop_get:NnN 2878
\msg_error:nnx 63	\prop_get:NnNT 2551, 2572
\msg_info:nnxx 68	\prop_get:NnNTF 2770, 2879
\msg_new:nnn 3, 17, 22, 38, 60	\prop_get:NoN 2851
\msg_new:nnnn 61, 4385, 4450	\prop_get:NoNTF 2848
\msg_redirect_module:nnn 3229, 3230, 3236, 3237	\prop_get:NVN 2862
\msg_warning:nn 64	\prop_get:NVNF 2253
\msg_warning:nnx 65	\prop_get:NVNT 2533, 2539, 2822, 2949
\msg_warning:nnxx 66	\prop_get:NVNTF 2800, 3096
\msg_warning:nnxxx 67	\prop_get:NxN 2858
	\prop_get:NxNTF 2855
N	\prop_gpop:NnNT 2470, 2472
indentfirst 3	\prop_gput:Nnn 2647, 2781
\newCJKfontfamily 6, 2755, 2757	\prop_gput:NnV 2573
\NewDocumentCommand 100, 314, 315, 465, 485, 524, 529, 536, 1242, 1441, 1449, 1458, 1553, 1567, 2142, 2156, 2247, 2655, 2734, 2740, 2745, 2755, 2757, 2763, 2789, 2792, 2937,	\prop_gput:NVV 2512, 2514, 2536, 2543, 3137
	\prop_gput:Nxx 2639
	\prop_if_empty:NF 2697

\prop_if_empty:NTF	2897	\seq_set_split:Nnn	1218
\prop_map_break:n	2912	\seq_use:Nnnn	2140
\prop_map_inline:Nn	2437, 2519, 2910	\setCJKfallbackfamilyfont	7, 7, 2247, 2247
\prop_new:N	2401, 2448, 2507, 2508, 2509, 2787, 3150	\setCJKfamilyfont	5, 2721, 2755, 2755
\prop_put:Nnn	2312, 2451, 2453	\setCJKmainfont	5, 2717, 2734, 2734, 2739, 2750
\prop_put:Nnx	2328	\setCJKmathfont	7, 2751, 2937, 2937
\protect	3371, 3372	\setCJKmonofont	5, 2719, 2734, 2745, 2753
\ProvideTextCommandDefault	4515	\setCJKromanfont	2739, 2754
\PunctBoundWidth	1572	\setCJKsansfont	5, 2718, 2734, 2740, 2752
\PunctStyle	2117	\SetSymbolFont	2957
\punctstyle	3323, 3323	\sfdefault	2890, 3466
\PunctWidth	1572	\shapedefault	2953, 2955, 2958
Q			
\q_mark	2324, 2334	\silent	3225
\q_nil	2324, 3736	\skip_add:Nn	1252
\q_no_value	2312	\skip_horizontal:N	965, 1256, 3076, 3688, 3695, 3774, 3830
\q_recursion_stop	3736	\skip_horizontal:n	940, 1056, 1057, 1160, 3078, 3257, 3989, 3996
\q_recursion_tail	3736	\skip_if_eq:nnTF	152, 161, 166, 183, 1048, 3061
\q_stop	274, 277, 289, 2324, 2334, 4548, 4679, 4683	\skip_if_eq_p:nn	625, 672, 727, 728, 1044
\quark_if_nil:nTF	2336	\skip_if_finite_p:n	1043
\quark_if_no_value:nTF	2522	\skip_new:N	58, 789, 817, 3109
\quark_if_recursion_tail_stop:N	3757	\skip_set:Nn	208, 3091, 3112, 3997
\quiet	3225	\skip_set_eq:NN	1250, 3992
R			
\r	3560, 3561	\skip_use:N	208, 3126
\ReloadXunicode	4418, 4419, 4419	\skip_vertical:n	4062
\RenewDocumentCommand	3424, 3460, 3573, 3999, 4028, 4419, 4462, 4469, 4488, 4535, 4573, 4580	\skip_zero:N	3100, 3116
\RenewDocumentEnvironment	4076	\SlantFactor	2271
\RequirePackage	48, 51, 52, 3283, 3284, 3286, 3610, 3617, 3620, 3628, 3654, 3666, 3667, 3668, 3669, 4118, 4119, 4393, 4410, 4414	\SplitArgument	486
\reverse_if:N	220, 4298, 4315, 4322	\str_case:nn	3471
\rmdefault	2889, 3465	\str_case:nnTF	2104, 3369
S			
\s_stop	119, 121	\str_case_x:nnn	2715, 2726, 2887
\scan_align_safe_stop:	3384, 3390	\str_case_x:nnTF	1392
\scan_stop:	91, 207, 227, 849, 860, 861, 918, 925, 1849, 1942, 1952, 2032, 2053, 3153, 3381, 3425, 3665, 4053, 4117, 4397, 4435, 4459, 4551	\str_case_x:noTF	1386
\selectfont	2644, 3185, 3197	\str_if_eq:nnF	477, 597, 2606
\seq_clear:N	2806	\str_if_eq:nnT	44, 3632
\seq_count:N	3512	\str_if_eq:nnTF	593, 1504, 2311, 2609, 3741
\seq_gclear:c	1611	\str_if_eq_x:nnTF	113, 2228, 2903, 3146, 3159, 3379
\seq_gclear:N	1549	\str_if_eq_x_p:nn	1289, 3565, 3566
\seq_gput_right:cn	1615, 1625	\sw@slant	3474
\seq_gput_right:Nn	326, 337, 460, 1525, 2299, 2701	T	
\seq_gput_right:Nv	327	\tabular	4095, 4101, 4108
\seq_gput_right:Nx	2146	\tex_afterassignment:D	4544
\seq_gremove_all:cn	1634	\tex_baselineskip:D	3257
\seq_gset_eq:NN	453	\tex_chardef:D	4551
\seq_gset_from_clist:Nn	455	\tex_font:D	141, 145, 154, 155, 156, 164, 170, 171, 176, 177, 186, 2097, 3105, 3113, 3120, 3157, 3178, 3186, 3193, 3346, 4459
\seq_if_empty_p:N	2817, 2826	\tex_fontdimen:D	154, 155, 156, 164, 170, 171, 176, 177, 186, 3113, 3120, 3178, 3186, 3193
\seq_if_in:cnF	1622	\tex_futurelet:D	4311
\seq_if_in:NnF	1224, 1418, 2699, 3516	\tex_global:D	2977
\seq_if_in:NnTF	2810	\tex_hfill:D	4079, 4087
\seq_map_function:NN	579, 2819	\tex_hss:D	4066, 4327
\seq_map_inline:cn	1609	\tex_iftrue:D	3609
\seq_map_inline:Nn	591, 890, 1237, 1434, 1500, 1516, 2804, 3524, 3737, 3739, 4149, 4174, 4188	\tex_ignorespaces:D	91, 903, 907, 2674, 2766, 2795, 3317, 4016, 4039
\seq_new:c	1606	\tex_italiccorrection:D	654, 655, 691, 692
\seq_new:N	317, 318, 452, 454, 457, 1213, 1430, 1440, 2149, 2296, 2707	\tex_kern:D	770, 771, 930, 942, 4065
\seq_put_right:Nn	1224, 1419, 2812	\tex_lastkern:D	759, 3995
\seq_remove_all:Nn	1229, 1426	\tex_lastpenalty:D	1054, 1064
\seq_set_from_clist:Nn	1411	\tex_lastskip:D	208, 625, 672, 727, 728, 1043, 1044, 1048, 1250, 1252
		\tex_let:D	3594
		\tex_noindent:D	4131
		\tex_number:D	201
		\tex_parindent:D	1033
		\tex_penalty:D	69, 70, 1068, 1069, 3687, 3694
		\tex_romannumeral:D	240, 266

\use:n	1036, 1057, 1059, 1069, 1071, 1402, 1795, 1799, 1829, 1839, 1895, 1899, 1905, 1924, 2014, 2050, 2052, 2904, 3303, 3488, 3494, 4375
\use:x	2248, 2431, 2765, 3117, 3601, 4078, 4423
\use_i:nn	2441, 2444, 2526, 2651, 3766
\use_ii:nn	2439, 2526, 3764
\use_iii:nnn	1395
\use_iiii:nnn	1373
\use_none:n	220, 221, 222, 1035, 1038, 1046, 1056, 1068, 1792, 1796, 1892, 1896, 2049, 3496, 4532, 4691, 4697
\use_none:nn	174, 188, 2459, 2659
\use_none:nnnnn	2947
\use_none_delimit_by_q_stop:w	4544
\usefont	3574
\UseInstance	1761, 1779
\usepackage	3296, 4454
\UTFencname	3423, 4398, 4399, 4402, 4405, 4406, 4407, 4440, 4462, 4469, 4488, 4535, 4559
AutoFakeBold	4, 5
AutoFakeSlant	4, 5
AutoFallBack	4
V	
\vbox_gset_to_ht:Nnn	4060
\Verb	2982
\verbatim@font	3010, 3011
X	
\x@protect	3371
\xCJKecglue	790
\xe@alloc@intercharclass	3512, 3514
\xCJK@family	2668, 2681, 2772, 3099
\xCJK@fix@penalty	3472, 3472, 3473, 3474, 3726
\xCJK@italiccorr	3473, 3475, 3475
\xCJK@setfont	2603
\xCJK_add_font_features:Nnn	2798, 2798, 2840
\xCJK_add_font_features:Nnx	3135
\xCJK_add_font_features:Nxx	2794
\xCJK_allow_break:	69, 69
\xCJK_app_inter_class_toks:nnn	557, 557, 562, 617, 903
\xCJK_app_inter_class_toks:nnx	3528
\xCJK_Boundary_and_Default:	616, 619, 619, 3724
\xCJK_Boundary_and_FullLeft:N	900, 1011, 1011
\xCJK_Boundary_and_FullRight:N	902, 1088, 1088
\xCJK_Boundary_and_NormalSp:	665, 666, 666, 3725
\xCJK_calc_punct_dimen:f	1752
\xCJK_calc_punct_dimen:N	2085, 2085, 2111
\xCJK_char_from_charcode:Nn	3637, 3644
\xCJK_check_for_glue:	701, 707, 707, 3038, 3054, 3070
\xCJK_check_FullRight:	1175, 1179, 1185, 1194, 1194
\xCJK_check_FullRight_symbol:Nw	1180, 1208, 1208
\xCJK_check_single:NNw	1280, 1284, 1294, 1299, 1299
\xCJK_check_single:Nw	1263, 1266, 1271, 1277, 1277
\xCJK_check_single_cs:NNn	1325, 1326, 1368, 1368
\xCJK_check_single_env:nnNn	1374, 1383, 1383
\xCJK_check_single_equation:NNnNw	1337, 1338, 1359, 1359
\xCJK_CJK_and_Boundary:w	825, 826, 826, 3039, 3055, 3071
\xCJK_CJK_and_CJK:N	884, 885, 885, 1263, 1265, 1266, 1271, 1272, 3723
\xCJK_CJK_and_FullLeft:N	997, 997
\xCJK_CJK_and_FullRight:N	1098, 1098
\xCJK_class_group_begin:	582, 582, 606, 702, 984, 1016, 1081, 1092, 2211, 3814, 3824
\xCJK_class_group_end:	582, 588, 612, 831, 832, 843, 846, 856, 959, 962, 976, 1205, 2214, 3074, 3789, 3805, 3822, 3864, 3900, 3914, 3917, 3938, 3952, 3979
\xCJK_class_num:n	464, 464, 480, 483, 520, 527, 545, 548, 1639, 3081
\xCJK_clear_Boundary_and_CJK_toks:	578, 578, 609, 703, 987, 1018, 1084, 1094, 3815, 3825
\xCJK_clear_inter_class_toks:nn	549, 549, 581, 608, 986, 1083
\xCJK_copy_inter_class_toks:nnnn	563, 563, 594, 596, 598, 1502, 1503, 1515, 1518, 1519, 3526, 3527, 3535, 3541
\xCJK_cs_case_keys_define:nNnNn	1210, 1210, 1240, 1393, 1400
\xCJK_cs_clear:N	127, 127, 3063, 3064, 3584, 3585, 3586, 3680, 3681, 3712, 3715, 4326
\xCJK_cs_gclear:N	127, 129
\xCJK_declare_char_class:nn	470, 470, 488, 1495
\xCJK_declare_char_class:nV	531, 532, 533, 534, 538, 539, 540
\xCJK_declare_char_class:nx	467, 1473
\xCJK_declare_sub_char_class:nnn	1487, 1487, 1497
\xCJK_declare_sub_char_class:nxx	1444
\xCJK_def_node:nn	762, 762, 775, 776, 777, 778, 779, 780
\xCJK_Default_and_FullLeft:nN	980, 980
\xCJK_Default_and_FullRight:nN	1077, 1077
\xCJK_fallback_loop:Nn	2213, 2217, 2217, 2225, 2237
\xCJK_fallback_test_glyph:N	2192, 2195, 2200, 2206, 2206
\xCJK_family_if_exist:x	2648
\xCJK_family_if_exist:xF	2621, 2900
\xCJK_family_if_exist:xT	2549, 2563
\xCJK_family_if_exist:XTF	2219, 2648, 2665, 2678, 2691, 2906, 2942, 2945
\xCJK_family_if_exist_use:x	2660, 2663, 2689, 2689
\xCJK_font_gset_to_current:c	138, 138, 2590, 2599, 2631, 3179
\xCJK_fontspec:nn	2765, 2768, 2768, 2785
\xCJK_fontspec:VV	2834
\xCJK_FullLeft_and_CJK:	887, 944, 944, 3720
\xCJK_FullLeft_and_Default:	954, 954, 3719, 3928
\xCJK_FullLeft_and_FullLeft:N	1120, 1120
\xCJK_FullLeft_and_FullRight:N	1129, 1129
\xCJK_FullRight_and_Boundary:	903, 905, 906, 906, 1175, 1177, 1179, 1185, 1187
\xCJK_FullRight_and_CJK:	889, 966, 966, 3722
\xCJK_FullRight_and_Default:	907, 973, 973, 3721
\xCJK_FullRight_and_FullLeft:N	1139, 1139
\xCJK_FullRight_and_FullRight:N	1161, 1161
\xCJK_FullRight_symbol:N	1086, 1096, 1103, 1137, 1168, 1178, 1180, 1188, 1193, 1193
\xCJK_get_inter_class_toks:nn	547, 547, 554, 560, 565, 571, 3529, 3531, 3533, 3539, 4164
\xCJK_get_punct_bounds:NN	982, 999, 1013, 1079, 1090, 1100, 1123, 1132, 1141, 1163, 1196, 1737, 1737, 3912
\xCJK_get_punct_kerning:NN	1769, 1769, 1784
\xCJK_get_punct_kerning:oN	1124, 1133, 1142, 1164
\xCJK_get_to_skip:nN	205, 205, 786, 795, 812, 3684, 3691
\xCJK_glyph_bounds:NN	2088, 2090, 2112, 2112
\xCJK_glyph_if_exist:N	143
\xCJK_glyph_if_exist:NTF	143, 2208, 2223, 3154
\xCJK_glyph_if_exist_p:N	143
\xCJK_gset_mathcode:nnnn	2967, 2972, 2972
\xCJK_hook_for_ulem:	3670, 3671, 3671
\xCJK_if_blank_x:n	210
\xCJK_if_blank_x:nT	3532, 3538
\xCJK_if_blank_x:nTF	210, 493, 2657
\xCJK_if_blank_x_p:n	210, 491
\xCJK_if_CJK_class:N	293
\xCJK_if_CJK_class:NTF	293, 1351, 1353
\xCJK_if_CJK_class_p:N	293
\xCJK_if_last_node:n	757
\xCJK_if_last_node:nT	644, 684, 873, 3775, 3782
\xCJK_if_last_node:nTF	713, 757, 3479, 3485, 3491, 3773
\xCJK_if_last_node_p:n	631, 632, 639, 640, 678, 679, 710, 718, 734, 736, 742, 743, 757

\xeCJK_if_package_loaded:n	11	\xeCJK_space_or_xecglue:	
\xeCJK_if_package_loaded:nF	3231, 3238, 3614		645, 685, 801, 806, 813, 831, 844, 868, 880
\xeCJK_if_package_loaded:nT	29, 3448, 3571, 3579, 3619, 3627	\xeCJK_swap_cs:NN	131, 131, 3758, 4023, 4371
\xeCJK_if_package_loaded:nTF		\xeCJK_switch_family:n	
	11, 34, 80, 3243, 3432, 3438, 3616		2676, 2760, 2783, 2919, 3465, 3466, 3467, 3468
\xeCJK_if_package_loaded_p:n	11, 3416, 3417, 3593	\xeCJK_tl_remove_outer_braces:N	114, 114
\xeCJK_if_same_class:NN	301	\xeCJK_tl_remove_outer_braces:n	
\xeCJK_if_same_class:NNTF	301, 2021		114, 115, 116, 124, 2321, 2340
\xeCJK_if_same_class_p:NN	301	\xeCJK_token_value_charcode:N	272, 273, 273, 2097, 2115
\xeCJK_if_ulem_patch:TF	3761, 3761, 3771, 3781, 3787, 3797, 3803, 3820, 3837, 3849, 3862, 3877, 3898, 3908, 3926, 3935, 3949, 3965, 3971, 3977, 3987, 4021	\xeCJK_token_value_class:N	271, 271, 295, 304, 1639
\xeCJK_ignore_spaces:w	863, 863, 3503	\xeCJK_visible_space:	3151, 3151, 3167
\xeCJK_int_until_do:nn	219, 219, 503, 512, 2975	\xeCJK_visible_space_fallback:	3163, 3168, 3168
\xeCJK_inter_class_toks:nnc	896, 911	\xeCJK_xetex_mathcode:w	286, 291, 2977
\xeCJK_inter_class_toks:nnn	544, 544, 546, 550, 604, 612, 616, 651, 664, 688, 699, 825, 884, 886, 888, 899, 901, 904, 3743, 3745, 3749, 4154, 4156, 4163, 4168, 4170, 4172, 4176, 4182, 4184, 4186, 4190	\xeCJKactive	307
\xeCJK_inter_class_toks:nmx	553, 559, 567, 575, 894	\xeCJKallowbreakbetweenpuncts	3329, 3329
\xeCJK_make_node:n	655, 658, 659, 692, 695, 696, 762, 768, 843, 846, 856, 874, 1364, 1377, 3482, 3488, 3494, 3791	\xeCJKCancelSubCJKBlock	9, 1448, 1449
\xeCJK_new_class:n	319, 319, 352, 353, 354, 355, 1491	\xeCJKcaption	3596, 3598
\xeCJK_new_sub_key:n	1493, 2296, 2297, 2351	\xeCJKDeclareCharClass	9, 465, 465
\xeCJK_no_break:	69, 70, 714, 949, 952, 959, 962, 1112, 1116, 1122, 1131, 1136, 1150, 1167, 1202, 1255, 1363, 1376, 3076, 3078, 3886, 3890, 3914, 3919, 4074	\xeCJKDeclarePunctStyle	
\xeCJK_peek_after_ignore_spaces:nw	255, 255, 1209		10, 2142, 2142, 2155, 2163, 2164, 2165, 2172, 2180
\xeCJK_peek_catcode_ignore_spaces:NTF		\xeCJKDeclareSubCJKBlock	8, 1441, 1441, 1447, 1484
	225, 225, 828, 865, 1301	\xeCJKdisablefallback	3333, 3335
\xeCJK_pre_inter_class_toks:nnn		\xeCJKEditPunctStyle	10, 2156, 2156, 2162
	551, 551, 556, 662, 1506, 1520, 1522, 1529	\xeCJKenablefallback	3333, 3333
\xeCJK_pre_inter_class_toks:nmx	3530	\xeCJKknobreak	12, 1242, 1242
\xeCJK_punct_kern:NN	1143, 1152, 1578, 1583	\xeCJKknobreakbetweenpuncts	3329, 3331
\xeCJK_punct_kerning_process:NN	1780, 1957, 1957	\xeCJKKOffVerbAddon	12, 3013, 3025, 3040, 3058
\xeCJK_punct_margin_process:NN	1762, 1785, 1785	\xeCJKKplainchr	3323, 3324
\xeCJK_punct_offset_process:NN	1763, 1885, 1885	\xeCJKResetCharClass	9, 536, 536, 543
\xeCJK_remove_node:	634, 642, 645, 681, 685, 711, 714, 721, 738, 746, 773, 874, 3481, 3487, 3493, 3774, 3775, 3782	\xeCJKResetPunctClass	
\xeCJK_replace_inter_class_toks:nnnn	569, 569, 1510		9, 468, 529, 529, 541, 1445, 1455, 1464, 3340
\xeCJK_reverse:nnn	112, 112, 1758	\xeCJKRestoreSubCJKBlock	9, 1448, 1458
\xeCJK_save_class:nn	330, 330, 347, 348, 349, 350, 351	\xeCJKsetcharclass	3337, 3337
\xeCJK_select_font:	607, 704, 985, 1017, 1082, 1093, 1752, 2581, 2584, 2603, 2610, 3103, 3344, 3816	\xeCJKsetecglue	3326
\xeCJK_select_font:x	2221, 2581, 2593	\xeCJKsetemboldenfactor	3319, 3319
\xeCJK_set_char_class:nnn	479, 483, 508, 508, 3339	\xeCJKsetkern	10, 1567, 1567, 1569
\xeCJK_set_family:nnn	2407, 2407, 2429, 2431, 2782	\xeCJKsetslantfactor	3319, 3321
\xeCJK_set_family:Voo	2525	\xeCJKsetup	2, 1550, 3059, 3297, 3314, 3314, 3320, 3322, 3323, 3324, 3325, 3327, 3328, 3330, 3332, 3334, 3336, 3676
\xeCJK_set_family:VVV	2261	\xeCJKsetwidth	9, 1553, 1553, 1565
\xeCJK_set_family:xxx	2430, 2736, 2742, 2747, 2756, 2761, 2938	\xeCJKShipoutHook	12, 99, 99, 100
\xeCJK_set_family_fallback:nnn	2248, 2249, 2249	\xeCJKVerbAddon	12, 2998, 3005, 3013, 3013, 3086, 3142
\xeCJK_set_mathfont:	2920, 2940, 2940	\xetex_if_engine:F	10
\xeCJK_set_visible_space_font:	3171, 3174, 3174	\xetex_if_engine:TF	4404, 4409
		\xetex_XeTeXversion:D	275
		\XeTeXcharclass	272, 514, 522, 527, 3083
		\XeTeXcharglyph	2115
		\XeTeXdefaultencoding	3600, 3607
		\XeTeXfonttype	3157
		\XeTeXglyphbounds	1533, 1537, 2114
		\XeTeXinterchartokenstate	314, 315, 3477
		\XeTeXinterchartoks	545, 548
		\XeTeXmathcode	291
		\XeTeXpicfile	4412, 4413, 4415
		\XeTeXrevision	275