

xeCJK 宏包

ctex.org

2013/01/11 3.1.2*

目录

1	简介	1	5.4	字符类别设定	17
2	基本用法	2	5.5	字符类别处理	20
3	用户手册	2	5.6	字符输出规则	24
3.1	宏包选项	2	5.7	全角右标点后的断行	32
3.2	字体设置与选择	5	5.8	段末孤字处理	33
3.2.1	X _g TeX 的字体名查找	7	5.9	增加 CJK 子分区	37
3.3	CJK 分区字体设置	8	5.10	标点处理	38
3.4	设置 CJK 字符范围	9	5.11	后备字体	48
3.5	标点符号的处理	9	5.12	CJK 字体族声明方式	50
3.5.1	设置特定标点符号的宽度和间距	9	5.13	字体切换	57
3.5.2	定义标点符号处理格式	9	5.14	数学字体设置	63
3.6	其它	11	5.15	抄录环境中的间距调整	64
4	已知问题和兼容性	11	5.16	xeCJK 其它选项	66
5	xeCJK 代码实现	12	5.17	xeCJK 初始化设置	67
5.1	运行环境检查	12	5.18	兼容性修补	69
5.2	内部工具	12	5.19	xeCJKfntef	73
5.3	功能开关	17	5.20	xeCJK.cfg	80
			版本历史	81	
			代码索引	81	

1 简介

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

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

xeCJK 使用了 X_gTeX 的一些最新特性,需要 X_gTeX 0.9995.0 [2009/06/29] 以后的版本。xeCJK 依赖 L^AT_EX3 项目的宏包套件 l3kernel 和 l3packages。xeCJK 还需要通过 fontspec 宏包来调用系统字体。将在 3.1 节介绍的功能选项 CJKnumber 需要 CJK 宏包套件中的 CJKnumb 的支持; indentfirst 选项需要 indentfirst 宏包。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 rev399.

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` 目录下面也有一些例子可以参考。

3 用户手册

3.1 宏包选项

`xeCJK` 以 $\langle key \rangle = \langle var \rangle$ 的形式提供宏包选项,你可以在调用宏包的时候直接设置这些选项,也可以在调用宏包之后使用 `\xeCJKsetup` 来设置这些选项。`xeCJK` 内部调用 `fontspec` 宏包,可以在调用 `xeCJK` 的时候,使用它的宏包选项。`xeCJK` 会将 `fontspec` 的选项传递给它。

<code>\xeCJKsetup</code>	<code>\xeCJKsetup {$\langle key_1 \rangle = \langle var_1 \rangle$, $\langle key_2 \rangle = \langle var_2 \rangle$, ...}</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</code> \star	<code>LocalConfig = {$\langle true \rangle false name \rangle$}</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 `mktexlsr`,刷新文件名数据库以便 \TeX 系统能够找到它。

CJKnumber ★	CJKnumber = $\langle true false \rangle$
	为了与 CJKnumb 宏包兼容,xeCJK 作了一些特别处理。因此,如果需要使用到 CJKnumb,请通过这个选项调用它。
indentfirst ★	indentfirst = $\langle true false \rangle$
Updated: 2012-11-22	是否使用 indentfirst 宏包,使得跟在章节标题后面的第一段首行也缩进。
	<p>请注意,xeCJK 宏包中只有上述 LocalConfig、CJKnumber 和 indentfirst 这三个选项需要在调用 xeCJK 时设置,而不能通过 \xeCJKsetup 来设置。</p>
xeCJKactive	xeCJKactive = $\langle true false \rangle$
	打开/关闭对中文的特殊处理。事实上,这个选项会打开/关闭 \XeTeX 的整个字符类机制,依赖这个机制的宏包都会受到影响。
CJKspace	CJKspace = $\langle true false \rangle$
	缺省状态下,xeCJK 会忽略 CJK 文字之间的空格,使用这一选项来保留它们之间的空格。
CJKmath ★	CJKmath = $\langle true false \rangle$
	是否支持在数学环境中直接输入 CJK 字符。使用这个选项后,可以直接在数学环境中输出 CJK 字符。
CJKglue	CJKglue = $\{\backslash\text{hskip } 0\text{pt plus } 0.08\backslash\text{baselineskip}\}$
	设置 CJK 文字之间插入的 glue,上边是 xeCJK 的默认值。一般来说,除非有特殊需要(例如,改变文字间距等),否则不需要设置这个选项,使用默认值即可。如果要设置这个选项,为了行末的对齐,设置的 glue 最好有一定的弹性。
CJKecglue	CJKecglue = $\{\langle glue \rangle\}$
	设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距,默认值是一个空格。使用这个选项设置的 glue 最好也要用一定的弹性。请注意,这里设置的 glue 只影响 xeCJK 根据需要自动添加的空白,源文件中直接输入的 CJK 文字与西文之间的空格不受影响(直接输出)。有时候 xeCJK 可能不能正确地调整间距,需要手动加空格。
xCJKecglue	xCJKecglue = $\{\langle true false glue \rangle\}$
	缺省状态下,xeCJK 不对源文件中直接输入的 CJK 文字与西文之间的空格进行调整,如果需要调整,请使用这个选项。如果使用这个选项,将使用 CJKecglue 替换源文件中直接输入的 CJK 文字与西文之间的空格。
CheckSingle	CheckSingle = $\langle true false \rangle$
Updated: 2012-12-06	是否避免单个 CJK 文字单独占一个段落的最后一行。需要说明的是,这个选项只有在段末的最后一个字是 CJK 文字或者标点符号,并且倒数第二和第三个字都是文字才能正确处理孤字的问题。如果这倒数三个字有作为控制序列的参数情况,那么一般来说也不能正确处理。
PlainEquation	PlainEquation = $\langle true false \rangle$
New: 2012-12-06	如果使用了 $\$ \$ \dots \$ \$$ 的形式来输入行间数学公式,就需要启用本选项,以便 CheckSingle 选项能够正确识别。推荐使用 $\backslash[\dots \backslash]$ 的形式来输入行间数学公式。
NewLineCS NewLineCS+ NewLineCS-	NewLineCS = $\{\backslash\text{par } \backslash[\}$
New: 2012-12-04	设置造成断行的控制序列,以便 CheckSingle 选项能够正确识别。以上是 xeCJK 的初始设置。
EnvCS EnvCS+ EnvCS-	EnvCS = $\{\backslash\text{begin } \backslash\text{end } \}$
New: 2012-12-04	设置 \LaTeX 环境开始和结束的控制序列,以便 CheckSingle 选项能够正确识别。以上是 xeCJK 的初始设置。

InlineEnv	InlineEnv = { $\langle env_1 \rangle$, $\langle env_2 \rangle$, $\langle env_3 \rangle$, ...}
InlineEnv+	在使用 CheckSingle 选项的时候,xeCJK 会将 CJK 文字后接着的 \LaTeX 环境的开始 $\begin{...}$ 和结束 $\end{...}$ 视为断行的地方,如果有某些特殊的 \LaTeX 环境没有造成断行,可以使用这个选项来声明它,以便 CheckSingle 能正确识别。
InlineEnv-	
Updated: 2012-12-06	
AutoFallBack	AutoFallBack = $\langle true false \rangle$
	当文档中有个别生僻字时,可以使用这个选项,自动使用预先设置好的后备字体来输出这些生僻字。后备字体的设置方法将在 3.2 节中介绍。
AutoFakeBold ☆	AutoFakeBold = { $\langle true false \text{数字} \rangle$ }
	全局设定当没有声明对应的粗体时,是否使用 伪粗体 ;当输入的是数字时,将使用伪粗体,并将使用输入的数字作为伪粗体的默认粗细程度。
AutoFakeSlant ☆	AutoFakeSlant = { $\langle true false \text{数字} \rangle$ }
	全局设定当没有声明对应的斜体时,是否使用 伪斜体 ;当输入的是数字时,将使用伪斜体,并将使用输入的数字作为伪斜体的默认倾斜程度。
EmboldenFactor ☆	EmboldenFactor = { $\langle \text{数字} 4 \rangle$ }
	设置伪粗体的默认粗细程度。
SlantFactor ☆	SlantFactor = { $\langle \text{数字} 0.167 \rangle$ }
	设置伪斜体的粗细程度,范围是 $-0.999 \sim 0.999$ 。
PunctStyle	PunctStyle = { $\langle \textit{quanjiao} \textit{banjiao} \textit{kaiming} \textit{hangmobanjiao} \textit{CCT} \textit{plain} ... \rangle$ }
Updated: 2012-11-10	设置标点处理格式。xeCJK 中预先定义好的格式为
	<div>quanjiao 全角式: 所有标点占一个汉字宽度,相邻两个标点占 1.5 汉字宽度;</div> <div>banjiao 半角式: 所有标点占半个汉字宽度;</div> <div>kaiming 开明式: 句末点号用全角,其他半角;</div> <div>hangmobanjiao 行末半角式: 所有标点占一个汉字宽度,行首行末对齐;</div> <div>CCT CCT 格式;</div> <div>plain 原样(不调整标点间距)。</div>
	可以使用 3.5.2 中介绍的 $\backslash\text{xeCJKDeclarePunctStyle}$ 定义新的标点格式。
KaiMingPunct ☆	KaiMingPunct = { $\langle . \text{ } . \text{ } ? \text{ } ! \text{ } \rangle$ }
KaiMingPunct+	设置开明(kaiming)标点处理格式时的句末点号,KaiMingPunct 后带的 + 与 - 分别表示从已有的开明句末点号中增加或减少标点。
KaiMingPunct-	
LongPunct ☆	LongPunct = { $\langle \text{——} \cdots \rangle$ }
LongPunct+	设置长标点,例如破折号“——”与省略号“……”,允许在长标点前后断行,但是禁止在它们之间断行。
LongPunct-	
MiddlePunct ☆	MiddlePunct = { $\langle \text{——} \cdot \cdot \cdot \rangle$ }
MiddlePunct+	设置居中显示的标点,例如间隔号“·”。对于在 CJK 文字之间的居中标点,xeCJK 会根据不同的标点处理格式,调整居中标点与前后文字之间的空白,保证其确实居中。对于行末出现的居中标点,允许在其后面断行,但禁止在它前面断行。
MiddlePunct-	
PunctWidth ☆	PunctWidth = { $\langle \text{length} \rangle$ }
	缺省状态下,xeCJK 会根据所选择的标点处理格式自动计算标点所占的宽度,如果对缺省设置不满意,可以通过这一选项来改变它。为了使得标点所占的宽度能够适应字体大小的变化,这里设置的 length 的单位最好用 em 等相对距离单位,而不建议使用诸如 pt 之类的绝对距离单位。这里的设置可用于除了 plain 以外的所有标点处理格式。同时,这里的设置对所有的 CJK 标点都生效,如果只要设置部分标点,请使用 3.5.1 节的 $\backslash\text{xeCJKsetwidth}$ 。

AllowBreakBetweenPuncts

AllowBreakBetweenPuncts = $\langle true|false \rangle$

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

CheckFullRight

CheckFullRight = $\langle true|false \rangle$

New: 2012-12-02

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

NoBreakCS
NoBreakCS+
NoBreakCS-

New: 2012-12-02

NoBreakCS = { \footnote \footnotemark \nobreak }

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

3.2 字体设置与选择

\setCJKmainfont ★

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

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

AutoFakeBold
AutoFakeSlant

AutoFakeBold = { $\langle true|false|数字 \rangle$ }**AutoFakeSlant** = { $\langle true|false|数字 \rangle$ }

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

\setCJKsansfont ★

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

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

\setCJKmonofont ★

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

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

\setCJKfamilyfont ★

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

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

\CJKfamily

\CJKfamily [$\langle family \rangle$]**\CJKfamily+** [$\langle family \rangle$]

Updated: 2012-10-27

\CJKfamily- [$\langle family \rangle$]

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

\newCJKfontfamily ★

\newCJKfontfamily [$\langle family \rangle$] [$\langle font-switch \rangle$] [$\langle font\ features \rangle$] [$\langle font\ name \rangle$]

声明新的 CJK 字体族 $\langle family \rangle$ 并指定字体,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 [**] {}

在文档中随机定义新的 CJK 字体族,并马上使用它。

\defaultCJKfontfeatures ☆ **

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 **

临时增加当前使用的 CJK 字体的选项。例如,使用

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

可以将文档中当前使用的 CJK 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 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 ☆ [**] {}

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 ☆ {<family>} [**] {}

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 = {[**]}{}

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

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

如果 FallBack 的值为空,将设置的是备用字体。例如,

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

等价于

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

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

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

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

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

3.2.1 Xe_{La}TeX 的字体名查找

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

Xe_{La}TeX 通常使用 fontconfig 库查找和调用字体,因此,可以用 `fc-list` 命令显示可用的字体。在命令行(Windows 的“命令提示符”,Linux 的 Console)下运行以下命令:

```
fc-list > fontlist.txt
```

可以将系统中所有安装的字体列表存入 `fontlist.txt` 文件中(可能很长)。

`fc-list` 命令列出的信息很多,而且在安装字体较多的 Windows 系统上的输出将非常庞大,如其中可能包含:

```
Times New Roman:style=cursiva,kurzíva,kursiv,Πλάγια,Italic,
Kursivoitu,Italique,Dőlt,Corsivo,Cursief,kursywa,Itálico,Курсив,
Ítalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta cursiva,tučné kurzíva,fed kursiv,
Fett Kursiv,Έντονα Πλάγια,Bold Italic,Negrita Cursiva,
Lihavoitu Kursivoi,Gras Italique,Félkövéř dőlt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrito Itálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın Ítalik,
Krepko poševno,nghiêng đặm,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντονα,Bold,Negrita,
Lihavoitu,Gras,Félkövéř,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
Полужирный,Fet,Kalın,Krepko,đặm,Lodia
Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,
thườg,Arrunta
宋体,SimSun:style=Regular
黑体,SimHei:style=Normal,obyčejné,Standard,Κανονικά,Regular,Normaali,
Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,Arrunta
```

在 fontspec 或 xeCJK 中使用的字体族名是上面列表中冒号前的部分。例如可以使用

```
\setmainfont{Times New Roman}
\setCJKmainfont{SimSun} % 或者 \setCJKmainfont{宋体}
```

来设置字体。

为了方便起见,`fc-list` 命令也可以加上各种选项控制输出格式,例如如果只要列出所有的中文字体的字体族名,可以用命令:

```
fc-list -f "%{family}\n" :lang=zh > zhfont.txt
```

这样就把字体列表保存在文件 `zhfont.txt` 中³。这样列出的字体列表就比较简明易用,如 Windows 下预装的中文字体:

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。

Arial Unicode MS
 FangSong, 仿宋
 KaiTi, 楷体
 Microsoft YaHei, 微软雅黑
 MingLiU, 細明體
 NSimSun, 新宋体
 PMingLiU, 新細明體
 SimHei, 黑体
 SimSun, 宋体

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

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

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

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

3.3 CJK 分区字体设置

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

`\xeCJKDeclareSubCJKBlock` ★

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

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

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

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

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

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

例如, 可以使用

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

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

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

`\xeCJKCancelSubCJKBlock`

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

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

`\xeCJKRestoreSubCJKBlock`

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

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

3.4 设置 CJK 字符范围

`\xeCJKDeclareCharClass` ★

`\xeCJKDeclareCharClass` $\{\langle class \rangle\} \{\langle class\ range \rangle\}$
`\xeCJKDeclareCharClass*` $\{\langle class \rangle\} \{\langle class\ range \rangle\}$

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

`\xeCJKResetCharClass` ★

`\xeCJKResetCharClass`

用于恢复 `\xeCJK` 对各个字符类别的初始化设置。

`\xeCJKResetPunctClass` ★

`\xeCJKResetPunctClass`

用于重置标点符号所属的字符类。

`\normalspacedchars`

`\normalspacedchars` $\{\langle char\ list \rangle\}$

在 $\langle char\ list \rangle$ 中出现的字符两端不自动添加空格,初始设置是 /。

3.5 标点符号的处理

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

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 `plain` 以外的所有标点处理格式。

`\xeCJKsetwidth` ★

`\xeCJKsetwidth` $\{\langle \text{标点列表} \rangle\} \{\langle length \rangle\}$

$\langle \text{标点列表} \rangle$ 可以是单个标点,也可以是多个标点。例如,

`\xeCJKsetwidth{。?}{0.7em}`

将设置句号和问号所占的宽度为 0.7 em。

`\xeCJKsetkern` ★

`\xeCJKsetkern` $\{\langle \text{前标点} \rangle\} \{\langle \text{后标点} \rangle\} \{\langle length \rangle\}$

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

`\xeCJKsetkern{:}{“}{0.3em}`

将设置冒号与左双引号之间的空白宽度为 0.3 em。

3.5.2 定义标点符号处理格式

`\xeCJKDeclarePunctStyle` ★

New: 2012-11-10

`\xeCJKDeclarePunctStyle` $\{\langle style \rangle\} \{\langle options \rangle\}$

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

`\xeCJKEditPunctStyle` ★

New: 2012-11-10

`\xeCJKEditPunctStyle` $\{\langle style \rangle\} \{\langle options \rangle\}$

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 $\langle boolean \rangle$ 类型的选项,将其设置为 `false`,对于 $\langle length \rangle$ 类型的选项,将其设置为 `\maxdimen`,而对于 $\langle real \rangle$ 类型的选项,将其设置为 0。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`add-min-bound-to-margin` *(boolean)* 是否在以上计算结果的基础上再加上标点的左右实际边界宽度中的最小值。这个选项对居中的标点无效。默认值是 `false`。

`optimize-margin` *(boolean)* 使用以上设置空白宽度或比例的选项时, 最终输出的标点符号左/右的空白宽度可能大于原来的实际边界宽度。若此时本选项被设置为 `true`, 则使用原来的实际边界宽度。而使用 `fixed-punct-width` 选项计算得出的左/右宽度可能小于该标点的另一侧宽度, 若此时本选项被启用, 则使用该标点的另一侧宽度。默认值为 `false`。

`margin-minimum` *(length)* 指定标点符号左/右的最小空白宽度。当经过以上选项设置的空白宽度小于这个选项的值时, 则使用这个选项的值。默认值是 `0 pt`。

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

`enabled-kerning` *(boolean)* 是否调整前后相邻的两个标点之间的空白宽度。如果设置为 `false`, 则每个标点都按原来的输出宽度输出。默认值是 `true`。

`min-bound-to-kerning` *(boolean)* 是否使用当前字体中前面标点实际左右边界的最小值与后面标点实际左右边界的最小值中的最大值作为两个标点之间的空白宽度。默认值是 `false`。

`kerning-total-width` *(length)* 设置两个标点的总共宽度。此时 `xeCJK` 会自动计算两个标点之间的空白宽度。默认值是 `\maxdimen`。

`kerning-total-ratio` *(real)* 设置两个标点的总共输出宽度与实际宽度的比例。默认值是 `0.75`。

`same-align-margin` *(length)* 当前后两个标点位于同侧时, 它们之间的空白宽度。默认值是 `\maxdimen`。

`same-align-ratio` *(real)* 当前后两个标点位于同侧时, 它们之间的空白宽度与实际输出宽度的比例。默认值是 `0`。

`different-align-margin` *(length)* 当前后两个标点位于异侧时, 它们之间的空白宽度。默认值是 `\maxdimen`。

`different-align-ratio` *(real)* 当前后两个标点位于异侧时, 它们之间的空白宽度与实际输出宽度的比例。默认值是 `0`。

`kerning-margin-width` *(length)* 设置前后两个标点之间的空白宽度。默认值是 `\maxdimen`。

`kerning-margin-ratio` *(real)* 设置前后两个标点之间的空白宽度与实际输出空白的比例。默认值是 `1.0`。

`optimize-kerning` *(boolean)* 使用以上选项计算出两个标点之间的空白宽度可能小于通过 `min-bound-to-kerning` 选项得出的结果。当出现这一情况时, 若此选项被设置为 `true`, 则使用该选项的空白宽度。默认值为 `false`。

`kerning-margin-minimum` *(length)* 指定两个标点之间的最小空白宽度。当经过以上选项设置的空白宽度小于这个选项的值时, 则使用这个选项的值。默认值是 `0 pt`。

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

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

就定义了一个名为 mine 的标点处理格式。可以在通过

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

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

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

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

3.6 其它

\xeCJKVerbAddon

New: 2012-11-19

\xeCJKVerbAddon

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

\xeCJKnobreak

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。

4 已知问题和兼容性

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

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

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

```
! Forbidden control sequence found while scanning use of \use_i:nn
```

的错误。目前已知的有 cprotect 宏包提供的 \cprotect。它的定义是

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

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

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

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

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

基于历史原因,`xltxtra` 宏包引入的 `realscripts` 定义了 `\dim_max:nn` 和 `\dim_min:nn` 这两个已经在当前版本的 `expl3` 中定义了的函数。但是 `realscripts` 中的定义并不好,并且会覆盖 `expl3` 中的定义,因此破坏了 `xeCJK` 中的相关计算。`xltxtra` 并没有什么实际功能,完全可以不使用它。如果原来通过 `xltxtra` 得到 \XeTeX 这个 Logo,现在可以改用更完善的 `hologo`。如果坚持使用 `xltxtra`,请把它放在 `xeCJK` 之后载入。

5 xeCJK 代码实现

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

5.1 运行环境检查

`xeCJK` 必须使用 \XeTeX 引擎的支持。

```
3 \msg_new:nnn { xeCJK } { Require-XeTeX }
4 {
5   The~xeCJK~package~requires~XeTeX~to~function.\\
6   You~must~change~your~typesetting~engine~to~"xelatex" \\
7   instead~of~plain~"latex"~or~"pdflatex"~or~"lualatex".\\
8   Loading~xeCJK~will~abort!
9 }
10 \xetex_if_engine:F { \msg_critical:nn { xeCJK } { Require-XeTeX } }
    应该使用较新版本的 expl3 宏包。
```

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

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

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

5.2 内部工具

分配临时变量。

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

```

\__xeCJK_msg_new:nn 各种信息函数的缩略形式。
\__xeCJK_error:n     36 \cs_new_protected_nopar:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:nx    37 \cs_new_protected_nopar:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_warning:nx  38 \cs_new_protected_nopar:Npn \__xeCJK_error:nx { \msg_error:nnx { xeCJK } }
\__xeCJK_info:nx     39 \cs_new_protected_nopar:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\__xeCJK_info:nxx    40 \cs_new_protected_nopar:Npn \__xeCJK_warning:nx { \msg_warning:nnx { xeCJK } }
                     41 \cs_new_protected_nopar:Npn \__xeCJK_warning:nxx { \msg_warning:nnxx { xeCJK } }
                     42 \cs_new_protected_nopar:Npn \__xeCJK_info:nx { \msg_info:nnx { xeCJK } }
                     43 \cs_new_protected_nopar:Npn \__xeCJK_info:nxx { \msg_info:nnxx { xeCJK } }

```

(End definition for __xeCJK_msg_new:nn and others.)

```

\xeCJK_allow_break:
\xeCJK_no_break:    44 \cs_new_protected_nopar:Npn \xeCJK_allow_break: { \tex_penalty:D \c_zero }
                    45 \cs_new_protected_nopar:Npn \xeCJK_no_break: { \tex_penalty:D \c_ten_thousand }

```

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

```

\xeCJK_if_package_loaded_p:n 判断宏包是否被引入,可用于文档正文中。
\xeCJK_if_package_loaded:nTF 46 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
                             47 {
                             48   \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
                             49   { \prg_return_true: } { \prg_return_false: }
                             50 }
                             51 \tl_const:Nx \c__xeCJK_package_ext_tl { \@pkgextension }

```

(End definition for \xeCJK_if_package_loaded:n)

```

\__xeCJK_at_end_preamble:n 在 \document 前后加上各种钩子。
\__xeCJK_after_preamble:n 52 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
\__xeCJK_after_end_preamble:n 53 \tl_new:N \g__xeCJK_after_preamble_hook_tl
                             54 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl
                             55 \cs_new_protected:Npn \__xeCJK_at_end_preamble:n #1
                             56 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
                             57 \cs_new_protected:Npn \__xeCJK_after_preamble:n #1
                             58 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
                             59 \cs_new_protected:Npn \__xeCJK_after_end_preamble:n #1
                             60 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
                             61 \xeCJK_if_package_loaded:nTF { etoolbox }
                             62 {
                             63   \AtEndPreamble { \g__xeCJK_at_end_preamble_hook_tl }
                             64   \AfterPreamble { \g__xeCJK_after_preamble_hook_tl }
                             65   \AfterEndPreamble { \g__xeCJK_after_end_preamble_hook_tl }
                             66 }
                             67 {
                             68   \AtBeginDocument { \g__xeCJK_after_preamble_hook_tl }
                             69   \cs_new_protected_nopar:Npn \__xeCJK_document_left_hook:
                             70   { \group_end: \g__xeCJK_at_end_preamble_hook_tl \group_begin: }
                             71   \cs_new_protected_nopar:Npn \__xeCJK_document_right_hook:
                             72   { \scan_stop: \g__xeCJK_after_end_preamble_hook_tl \tex_ignorespaces:D }
                             73   \cs_gset_nopar:Npx \document
                             74   {
                             75     \__xeCJK_document_left_hook:
                             76     \exp_not:o { \document }
                             77     \__xeCJK_document_right_hook:
                             78   }
                             79 }

```

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

```

\xeCJK_reverse:nnn #1 为 #2 或 #3,若 #1 和 #2 相等,则返回 #3,否则返回 #2。
                     80 \cs_new_nopar:Npn \xeCJK_reverse:nnn #1#2#3
                     81 { \str_if_eq:x:nnTF {#1} {#2} {#3} {#2} }

```

(End definition for \xeCJK_reverse:nnn)

`\xeCJK_tl_remove_outer_braces:N` 去掉 #1 外层的分组括号。

```

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

```

`\xeCJK_cs_clear:N` 让控制序列的意义为空。

`\xeCJK_cs_gclear:N`

```

92 \cs_new_eq:NN \xeCJK_cs_clear:N \tl_clear:N
93 \cs_new_eq:NN \xeCJK_cs_gclear:N \tl_gclear:N
(End definition for \xeCJK_cs_clear:N and \xeCJK_cs_gclear:N)

```

`\xeCJK_swap_cs:NN` 交换 #1 和 #2 的意义。

```

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

```

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

```

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

```

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

`\xeCJK_glyph_if_exist:NTF`

```

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

```

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

```

111 \tl_const:Nn \c_xeCJK_space_skip_tl
112 {
113   \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
114   {
115     \tex_fontdimen:D \c_two \tex_font:D
116     plus \tex_fontdimen:D \c_three \tex_font:D
117     minus \tex_fontdimen:D \c_four \tex_font:D
118   }
119   { \tex_spaceskip:D }
120 }
(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。

```
121 \cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2
122 {
123   \hbox_set:Nn \l__xeCJK_tmp_box { #1 \scan_stop: \exp_after:wN } \exp_after:wN
124   \skip_set:Nn \exp_after:wN #2 \exp_after:wN { \skip_use:N \tex_lastskip:D }
125 }
(End definition for \xeCJK_glue_to_skip:nN)
```

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

```
\xeCJK_if_blank_x:nTF
126 \prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
127 {
128   \if_case:w \pdfTeX_strcmp:D { } {#1} \exp_stop_f:
129   \prg_return_true:
130   \else:
131   \if_case:w \pdfTeX_strcmp:D { ~ } {#1} \exp_stop_f:
132   \prg_return_true: \else: \prg_return_false: \fi:
133   \fi:
134 }
(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
135 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
136 { \__xeCJK_int_until_do:wn \use_none:n { \reverse_if:N \if_int_compare:w #1#2 } }
137 \cs_new_protected:Npn \__xeCJK_int_until_do:wn \use_none:n #1
138 { #1 \exp_after:wN \__xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
(End definition for \xeCJK_int_until_do:nn and \__xeCJK_int_until_do:wn)
```

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

```
139 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
140 {
141   \cs_set_eq:NN \l__peek_search_token #1 \scan_stop:
142   \tl_set:Nn \__xeCJK_peek_catcode_true:w { \group_align_safe_end: #2 }
143   \tl_set:Nn \__xeCJK_peek_catcode_false:w { \group_align_safe_end: #3 }
144   \bool_set_false:N \l__xeCJK_peek_ignore_spaces_bool
145   \group_align_safe_begin:
146   \peek_after:Nw \__xeCJK_peek_catcode_ignore_spaces_branches:w
147 }
148 \cs_new_protected_nopar:Npn \__xeCJK_peek_catcode_ignore_spaces_branches:w
149 {
150   \if_meaning:w \l__peek_token \c_space_token
151   \bool_set_true:N \l__xeCJK_peek_ignore_spaces_bool
152   \exp_after:wN \peek_after:Nw
153   \exp_after:wN \__xeCJK_peek_catcode_ignore_spaces_branches:w
154   \tex_romannumeral:D 0
155   \else:
156   \if_catcode:w
157   \exp_not:N \l__peek_token \exp_not:N \l__peek_search_token
158   \exp_after:wN \exp_after:wN
159   \exp_after:wN \__xeCJK_peek_catcode_true:w
160   \else:
161   \exp_after:wN \exp_after:wN
162   \exp_after:wN \__xeCJK_peek_catcode_false:w
163   \fi:
164   \fi:
165 }
166 \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 `\futurenonSPACElet` 类似,会省略掉后面的空格。

```
167 \cs_new_protected:Npn \xeCJK_peek_after_ignore_spaces:nw #1
168 {
169   \tl_set:Nn \__xeCJK_peek_after_do:w { \group_align_safe_end: #1 }
170   \group_align_safe_begin:
171   \peek_after:Nw \__xeCJK_peek_ignore_spaces_branches:w
172 }
173 \cs_new_protected_nopar:Npn \__xeCJK_peek_ignore_spaces_branches:w
174 {
175   \if_meaning:w \l_peek_token \c_space_token
176     \exp_after:wN \peek_after:Nw
177     \exp_after:wN \__xeCJK_peek_ignore_spaces_branches:w
178     \tex_romannumeral:D 0
179   \else:
180     \exp_after:wN \__xeCJK_peek_after_do:w
181   \fi:
182 }
(End definition for \xeCJK_peek_after_ignore_spaces:nw)
```

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

```
183 \cs_new_nopar:Npn \xeCJK_token_value_class:N #1
184 { \XeTeXcharclass \xeCJK_token_value_charcode:N #1 }
(End definition for \xeCJK_token_value_class:N)
```

`\xeCJK_token_value_charcode:N` 当记号 #1 的 `charcode` 大于等于 `0x10000` 时, \XeTeX 0.9999 版以前的 `\meaning` 的返回结果比较特殊⁴,需要特别处理。

```
185 \cs_new_nopar:Npn \xeCJK_token_value_charcode:N #1
186 { \exp_after:wN \__xeCJK_token_value_charcode:w \token_to_meaning:N #1 \q_stop }
187 \fp_compare:nNnTF { \int_use:N \xetex_XeTeXversion:D \XeTeXrevision } > { 0.9998 }
188 {
189   \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3 \q_stop
190   { '#3 }
191 }
192 {
193   \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
194   { \tl_if_empty:nTF {#4} { '#3 } { "20000 } }
195 }
(End definition for \xeCJK_token_value_charcode:N)
```

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

```
\xeCJK_if_CJK_class:NNTF
196 \prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
197 {
198   \if_cs_exist:w \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 } \cs_end:
199   \prg_return_true: \else: \prg_return_false: \fi:
200 }
201 \cs_new_nopar:Npn \__xeCJK_CJK_class_tl:n #1
202 { c_#xeCJK_CJK_class_ \int_eval:n {#1} _tl }
203 \cs_generate_variant:Nn \__xeCJK_CJK_class_tl:n { c }
(End definition for \xeCJK_if_CJK_class:N These functions are documented on page ??.)
```

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

```
\xeCJK_if_same_class:NNTF
204 \prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
205 {
206   \if_int_compare:w
207     \xeCJK_token_value_class:N #1 = \xeCJK_token_value_class:N #2 \exp_stop_f:
208     \prg_return_true: \else: \prg_return_false: \fi:
209 }
(End definition for \xeCJK_if_same_class:NN)
```

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

5.3 功能开关

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

```
210 \keys_define:nn { xecjk / options }
211 {
212   xecjkActive .choice: ,
213   xecjkActive / true .code:n = { \makexecjkActive } ,
214   xecjkActive / false .code:n = { \makexecjkInactive } ,
215   xecjkActive .default:n = { true }
216 }
```

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

`\makexecjkActive`

`\makexecjkInactive`

```
217 \NewDocumentCommand \makexecjkActive { } { \XeTeXinterchartokenstate = \c_one }
218 \NewDocumentCommand \makexecjkInactive { } { \XeTeXinterchartokenstate = \c_zero }
```

(End definition for \makexecjkActive and \makexecjkInactive)

抑制 BOM。

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

5.4 字符类别设定

`\g__xecjk_class_seq`

`\g__xecjk_new_class_seq`

```
220 \seq_new:N \g__xecjk_class_seq
221 \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。

```
222 \cs_new_protected_nopar:Npn \xecjk_new_class:n #1
223 {
224   \int_if_exist:cTF { \__xecjk_class_csname:n {#1} }
225   { \__xecjk_error:nx { class-already-defined } {#1} }
226   {
227     \exp_args:Nc \newXeTeXintercharclass { \__xecjk_class_csname:n {#1} }
228     \clist_new:c { g__xecjk_#1_range_clist }
229     \seq_gput_right:Nn \g__xecjk_class_seq {#1}
230     \seq_gput_right:Nv \g__xecjk_new_class_seq { \__xecjk_class_csname:n {#1} }
231   }
232 }
```

(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。

```
233 \cs_new_protected_nopar:Npn \xecjk_save_class:nn #1#2
234 {
235   \int_if_exist:cTF { \__xecjk_class_csname:n {#1} }
236   { \__xecjk_error:nx { class-already-defined } {#1} }
237   {
238     \int_const:cn { \__xecjk_class_csname:n {#1} } {#2}
239     \clist_new:c { g__xecjk_#1_range_clist }
240     \seq_gput_right:Nn \g__xecjk_class_seq {#1}
241   }
242 }
```

(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。

```
243 \cs_new_nopar:Npn \__xecjk_class_csname:n #1 { c__xecjk_#1_class_int }
244 \cs_new_eq:cN { \__xecjk_class_csname:n { Others } } \l__xecjk_tmpa_int
245 \__xecjk_msg_new:nn { class-already-defined }
246 {
247   XeTeX~character~class~`#1'~has~been~already~defined.\\\
248   Please~take~another~name. \\
249 }
```

(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 Xe_{La}TeX 中预定义类别,xeCJK 新增加了 HalfLeft、HalfRight、NormalSpace。

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

Default 这五类是 Xe_{La}TeX 预定义的类别。

```

      CJK
FullLeft 250 \xeCJK\_save\_class:nn { Default } { \c\_zero }
FullRight 251 \xeCJK\_save\_class:nn { CJK } { \c\_one }
Boundary 252 \xeCJK\_save\_class:nn { FullLeft } { \c\_two }
253 \xeCJK\_save\_class:nn { FullRight } { \c\_three }
254 \xeCJK\_save\_class:nn { Boundary } { \c\_two\_hundred\_fifty\_five }
```

(End definition for Default and others.)

HalfLeft 新增西文半角左/右标点和前后原始间距的符号类。

```

HalfRight 255 \xeCJK\_new\_class:n { HalfLeft }
NormalSpace 256 \xeCJK\_new\_class:n { HalfRight }
257 \xeCJK\_new\_class:n { NormalSpace }
(End definition for HalfLeft, HalfRight, and NormalSpace)
```

```

\c\_xeCJK\_HalfLeft\_chars\_clist xeCJK 设置的基本字符类的默认范围。
\c\_xeCJK\_HalfRight\_chars\_clist
\c\_xeCJK\_NormalSpace\_chars\_clist
\c\_xeCJK\_FullLeft\_chars\_clist
\c\_xeCJK\_FullRight\_chars\_clist
\c\_xeCJK\_CJK\_chars\_clist
258 \clist\_const:Nn \c\_xeCJK\_HalfLeft\_chars\_clist
259 { "28 , "2D , "5B , "60 , "7B }
260 \clist\_const:Nn \c\_xeCJK\_HalfRight\_chars\_clist
261 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
262 \clist\_const:Nn \c\_xeCJK\_NormalSpace\_chars\_clist { "2F }
263 \clist\_const:Nn \c\_xeCJK\_FullLeft\_chars\_clist
264 {
265     "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
266     "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
267     "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
268     "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6
269 }
270 \clist\_const:Nn \c\_xeCJK\_FullRight\_chars\_clist
271 {
272     "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2030 , "2500 ,
273     "3001 , "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 ,
274     "3015 , "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 ,
275     "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C ,
276     "309D , "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 ,
277     "30E5 , "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE ,
278     "FE50 , "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C ,
279     "FE5E , "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B ,
280     "FF1F , "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 ,
281     "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E ,
282     "FF9F
283 }
284 \clist\_const:Nn \c\_xeCJK\_CJK\_chars\_clist
285 {
```

- Hangul Jamo (谚文字母)

```
286     "1100 -> "11FF ,
```

- CJK Radicals Supplement (中日韩部首补充)
287 "2E80 -> "2EFF ,
- Kangxi Radicals (康熙字典部首)
288 "2F00 -> "2FDF ,
- Ideographic Description Characters (表意文字描述符)
289 "2FF0 -> "2FFF ,
- CJK Symbols and Punctuation (中日韩符号和标点)
290 "3000 -> "303F ,
- Hiragana (日文平假名)
291 "3040 -> "309F ,
- Katakana (日文片假名)
292 "30A0 -> "30FF ,
- Bopomofo (注音字母)
293 "3100 -> "312F ,
- Hangul Compatibility Jamo (谚文兼容字母)
294 "3130 -> "318F ,
- Kanbun (象形字注释标志)
295 "3190 -> "319F ,
- Bopomofo Extended (注音字母扩展)
296 "31A0 -> "31BF ,
- CJK Strokes (中日韩笔画)
297 "31C0 -> "31EF ,
- Katakana Phonetic Extensions (日文片假名语音扩展)
298 "31F0 -> "31FF ,
- Enclosed CJK Letters and Months (带圈中日韩字母和月份)
299 "3200 -> "32FF ,
- CJK Compatibility (中日韩兼容)
300 "3300 -> "33FF ,
- CJK Unified Ideographs Extension-A (中日韩统一表意文字扩展 A)
301 "3400 -> "4DBF ,
- Yijing Hexagrams Symbols (易经六十四卦符号)
302 "4DC0 -> "4DFF ,
- CJK Unified Ideographs (中日韩统一表意文字)
303 "4E00 -> "9FFF ,
- Yi Syllables (彝文音节)
304 "A000 -> "A48F ,
- Yi Radicals (彝文字根)
305 "A490 -> "A4CF ,

- Hangul Jamo Extended-A (谚文扩展 A)

306 "A960 -> "A97F ,

- Hangul Syllables (谚文音节)

307 "AC00 -> "D7AF ,

- Kana Supplement (日文假名增补)

308 "B000 -> "B0FF ,

- Hangul Jamo Extended-B (谚文扩展 B)

309 "D7B0 -> "D7FF ,

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

310 "F900 -> "FAFF ,

- CJK Compatibility Forms (中日韩兼容形式)

311 "FE30 -> "FE4F ,

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

312 "FF00 -> "FFEF ,

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

313 "20000 -> "2A6DF ,

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

314 "2A700 -> "2B73F ,

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

315 "2B740 -> "2B81F ,

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

316 "2F800 -> "2FA1F

317 }

(End definition for \c__xeCJK_HalfLeft_chars_clist and others.)

5.5 字符类别处理

\g__xeCJK_base_class_seq

\g__xeCJK_non_CJK_class_seq

\g__xeCJK_CJK_class_seq

318 \seq_new:N \g__xeCJK_base_class_seq

319 \seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq

320 \seq_new:N \g__xeCJK_non_CJK_class_seq

321 \seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq

322 { Default , HalfLeft , HalfRight , NormalSpace , Boundary }

323 \seq_new:N \g__xeCJK_CJK_class_seq

324 \cs_new_protected_nopar:Npn __xeCJK_save_CJK_class:n #1

325 {

326 \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}

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

328 }

329 \clist_map_function:Nn { CJK , FullLeft , FullRight } __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。

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

(End definition for \xeCJK_class_num:n)

\xeCJKDeclareCharClass

```
331 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
332 {
333   \xeCJK_declare_char_class:nx {#2} {#3}
334   \IfBooleanT {#1} { \xeCJKResetPunctClass }
335 }
```

(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

```
336 \cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
337 {
338   \clist_set:Nn \l__xeCJK_tmpa_clist {#2}
339   \clist_gconcat:ccN
340     { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmpa_clist
341   \clist_map_inline:Nn \l__xeCJK_tmpa_clist
342     {
343       \str_if_eq:nnF {##1} { -> }
344       {
345         \_xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
346         { \xeCJK_class_num:n {#1} }
347       }
348     }
349 }
```

```
350 \NewDocumentCommand \_xeCJK_set_char_class_aux:Nnw
351 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
352 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
353 \cs_generate_variant:Nn \xeCJK_declare_char_class:nn { nx , nV }
```

(End definition for \xeCJK_declare_char_class:nn and _xeCJK_set_char_class_aux:Nnw)

_xeCJK_set_char_class_catcode:nnn

设置字符类别的同时, 设置所有 CJK 字符的 \catcode 为 12。Xe_gTeX 的配置文件 unicode-letters.tex 将所有 CJK 表意文字的 \catcode 设置为 11。我们在作出改变的目的是方便 CheckSingle 的判断。事实上, CheckSingle 的判断也可以不依赖 \catcode, 而改为判断 \XeTeXcharclass。但是判读一个隐性记号(\l_peek_token)的 \XeTeXcharclass 比直接判断它的 \catcode 要低效得多。

```
354 \cs_new_protected_nopar:Npn \_xeCJK_set_char_class_catcode:nnn #1#2#3
355 {
356   \_xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
357   \int_set:Nn \l__xeCJK_tmpc_int {#3}
358   \int_compare:nNnTF
359     { \use:c { \_xeCJK_class_csname:n { CJK } } } = \l__xeCJK_tmpc_int
360     { \cs_set_eq:NN \_xeCJK_set_char_catcode:n \char_set_catcode_other:n }
361     { \cs_set_eq:NN \_xeCJK_set_char_catcode:n \use_none:n }
362   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
363   {
364     \_xeCJK_set_char_catcode:n { \l__xeCJK_tmpa_int }
365     \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
366     \int_incr:N \l__xeCJK_tmpa_int
367   }
368 }
```

(End definition for _xeCJK_set_char_class_catcode:nnn)

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

```
369 \cs_new_protected_nopar:Npn \_xeCJK_set_char_class:nnn #1#2#3
370 {
371   \_xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
372   \int_set:Nn \l__xeCJK_tmpc_int {#3}
373   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
374   {
375     \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
376     \int_incr:N \l__xeCJK_tmpa_int
377   }
378 }
```

(End definition for __xeCJK_set_char_class:nnn)

__xeCJK_check_num_range:nnNN

```
379 \cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
380 {
381   \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
382   {
383     \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
384     \int_set_eq:NN #3 #4
385   }
386   {
387     \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
388     \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
389   }
390 }
```

(End definition for __xeCJK_check_num_range:nnNN)

\xeCJK_set_char_class:nnn #1 和 #2 为字符类别起止的 Unicode, #3 为类别名称对应编号。在宏包内部使用时, 设置 CJK 字符的 \catcode, 而在文档中使用时不设置。

```
391 \cs_new_eq:NN \xeCJK_set_char_class:nnn \__xeCJK_set_char_class_catcode:nnn
392 \AtEndOfPackage
393 { \cs_set_eq:NN \xeCJK_set_char_class:nnn \__xeCJK_set_char_class:nnn }
```

(End definition for \xeCJK_set_char_class:nnn)

__xeCJK_set_char_class_eq:nn

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

```
394 \cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
395 {
396   \int_set:Nn \l__xeCJK_tmpa_int { \xeCJK_class_num:n {#2} }
397   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
398   { \XeTeXcharclass ##1 = \l__xeCJK_tmpa_int }
399 }
```

(End definition for __xeCJK_set_char_class_eq:nn)

\normalspacedchars

声明前后不加间距的字符。

```
400 \NewDocumentCommand \normalspacedchars { m }
401 {
402   \tl_map_inline:nn {#1}
403   { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
404 }
```

(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。

```
405 \NewDocumentCommand \xeCJKResetPunctClass { }
406 {
407   \xeCJK_declare_char_class:nV { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
408   \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
409   \xeCJK_declare_char_class:nV { FullLeft } \c__xeCJK_FullLeft_chars_clist
410   \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
411 }
```

(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。

```
412 \NewDocumentCommand \xeCJKResetCharClass { }
413 {
414   \xeCJK_declare_char_class:nV { CJK } \c__xeCJK_CJK_chars_clist
415   \xeCJK_declare_char_class:nV { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
416   \xeCJKResetPunctClass
417 }
```

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

设置字符类别。

```
418 \xeCJKResetCharClass
```

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

```

419 \cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
420 { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }
421 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc , nnv , 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。

```

422 \cs_new_nopar:Npn \xeCJK_get_inter_class_toks:nn #1#2
423 { \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。

```

424 \cs_new_protected_nopar:Npn \xeCJK_clear_inter_class_toks:nn #1#2
425 { \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。

```

426 \cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
427 {
428   \xeCJK_inter_class_toks:nnx {#1} {#2}
429   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
430 }
431 \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。

```

432 \cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
433 {
434   \xeCJK_inter_class_toks:nnx {#1} {#2}
435   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
436 }
437 \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。

```

438 \cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
439 {
440   \xeCJK_inter_class_toks:nnx {#1} {#2}
441   { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
442 }
(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。

```

443 \cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
444 {
445   \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
446   \tl_replace_all:Nnn \l__xeCJK_tmpa_tl {#3} {#4}
447   \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_tl }
448 }
(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。

```

449 \cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
450 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
451 \cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
452 { \xeCJK_clear_inter_class_toks:nn { Boundary } {#1} }
(End definition for \xeCJK_clear_Boundary_and_CJK_toks:)

```

5.6 字符输出规则

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

```

\xeCJK_class_group_begin:
  \xeCJK_class_group_end:
453 \cs_new_eq:NN \xeCJK_class_group_begin: \c_group_begin_token
454 \cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token
(End definition for \xeCJK_class_group_begin: and \xeCJK_class_group_end:)
455 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
456 {
457   \xeCJK_inter_class_toks:nnn {#1} { CJK }
458   {
459     \xeCJK_class_group_begin:
460     \xeCJK_select_font:
461     \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
462     \xeCJK_clear_Boundary_and_CJK_toks:
463     \CJKsymbol
464   }
465   \xeCJK_inter_class_toks:nnn { CJK } {#1} { \xeCJK_class_group_end: }
466 }
467 \clist_map_inline:nn { Default , HalfLeft , NormalSpace }
468 {
469   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
470   {
471     \bool_if:nTF
472     {
473       \l__xeCJK_xecglue_bool &&
474       \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
475     }
476     {
477       \tex_unskip:D
478       \bool_if:nTF
479       {
480         \xeCJK_if_last_node_p:n { CJK } ||
481         \xeCJK_if_last_node_p:n { CJK-space }
482       }
483       { \CJKecglue } { ~ }
484     }
485     {
486       \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
487       { \xeCJK_if_last_node:nT { CJK-space } { \l__xeCJK_space_or_xecglue: } }
488     }
489   }
490   \str_if_eq:nnF {#1} { NormalSpace }
491   { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
492 }
493 \clist_map_inline:nn { Default , HalfRight , NormalSpace }
494 {
495   \xeCJK_inter_class_toks:nnn {#1} { Boundary }
496   {
497     \peek_meaning_remove:NTF \tex_italiccorrection:D
498     { \tex_italiccorrection:D { \xeCJK_make_node:n { default } } }

```

```

499     {
500         \token_if_space:NTF \l_peek_token
501         { { \xeCJK_make_node:n { default-space } } }
502         { { \xeCJK_make_node:n { default } } }
503     }
504 }
505 \str_if_eq:nnF {#1} { NormalSpace }
506 { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
507 }
508 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
509 {
510     \xeCJK_check_for_glue:
511     \xeCJK_class_group_begin:
512     \xeCJK_clear_Boundary_and_CJK_toks:
513     \xeCJK_select_font:
514     \CJKsymbol
515 }

```

\xeCJK_check_for_glue:

```

516 \cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
517 {
518     \bool_if:nTF
519     { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
520     { \CJKglue }
521     {
522         \bool_if:nTF
523         {
524             \xeCJK_if_last_node_p:n { default } ||
525             \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
526         }
527         { \CJKecglue }
528         {
529             \bool_if:nT
530             {
531                 \l__xeCJK_xecglue_bool &&
532                 ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
533                   \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_ecglue_skip )
534             }
535             {
536                 \tex_unskip:D
537                 \bool_if:nTF
538                 {
539                     \xeCJK_if_last_node_p:n { default-space } ||
540                     \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
541                     \xeCJK_if_last_node_p:n { default }
542                 }
543                 { \CJKecglue }
544                 {
545                     \bool_if:nTF
546                     {
547                         \xeCJK_if_last_node_p:n { CJK } ||
548                         \xeCJK_if_last_node_p:n { CJK-space }
549                     }
550                     { \bool_if:NTF \l__xeCJK_reserve_space_bool { ~ } { \CJKglue } }
551                     { ~ }
552                 }
553             }
554         }
555     }
556 }

```

(End definition for \xeCJK_check_for_glue:)

```

\xeCJK_if_last_node:p:n
\xeCJK_if_last_node:nTF
557 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
558 {
559   \if_dim:w \use:c { c__xeCJK_#1_node_dim } = \tex_lastkern:D
560   \prg_return_true: \else: \prg_return_false: \fi:
561 }
(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

```

562 \cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
563 {
564   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
565   { \dim_gset:cn } { \dim_const:cn }
566   { c__xeCJK_#1_node_dim } {#2}
567 }
568 \cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
569 {
570   \tex_kern:D - \use:c { c__xeCJK_#1_node_dim }
571   \tex_kern:D \use:c { c__xeCJK_#1_node_dim }
572 }
573 \xeCJK_def_node:nn { CJK } { 11 sp }
574 \xeCJK_def_node:nn { CJK-space } { 12 sp }
575 \xeCJK_def_node:nn { default } { 13 sp }
576 \xeCJK_def_node:nn { default-space } { 14 sp }
(End definition for \xeCJK_def_node:nn and \xeCJK_make_node:n)

```

CJKglue CJK 文字之间插入的 glue。

```

577 \keys_define:nn { xeCJK / options }
578 {
579   CJKglue .code:n =
580   {
581     \cs_set_protected_nopar:Npn \CJKglue {#1}
582     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
583   }
584 }
585 \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

```

586 \keys_define:nn { xeCJK / options }
587 {
588   CJKecglue .code:n =
589   {
590     \cs_set_protected_nopar:Npn \CJKecglue {#1}
591     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
592   } ,
593   xCJKecglue .choice: ,
594   xCJKecglue / true .code:n =
595   {
596     \bool_set_true:N \l__xeCJK_xecglue_bool
597     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
598   } ,
599   xCJKecglue / false .code:n =
600   {
601     \bool_set_false:N \l__xeCJK_xecglue_bool
602     \cs_set_eq:NN \__xeCJK_space_or_xecglue: \c_space_tl
603   } ,
604   xCJKecglue / unknown .code:n =
605   {
606     \bool_set_true:N \l__xeCJK_xecglue_bool
607     \cs_set_protected_nopar:Npn \CJKecglue {#1}
608     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip

```



```

609 \cs_set_eq:NN \__xeCJK_space_or_xecglue: \CJKecglue
610 } ,
611 xCJKecglue .default:n = { true }
612 }
613 \skip_new:N \l__xeCJK_ecglue_skip
614 \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。

```

615 \keys_define:nn { xeCJK / options }
616 {
617   CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
618   space .meta:n = { CJKspace = true } ,
619   nospace .meta:n = { CJKspace = false }
620 }

```

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

```

621 \xeCJK_inter_class_toks:nnn { CJK } { Boundary }
622 {
623   \xeCJK_class_group_end:
624   { \xeCJK_make_node:n { CJK } }
625   \xeCJK_ignore_spaces:w
626 }

```

\xeCJK_ignore_spaces:w

```

627 \cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
628 {
629   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
630   {
631     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
632     { \__xeCJK_space_or_xecglue: } { \CJKecglue }
633   }
634   {
635     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
636     {
637       \tex_unkern:D \tex_unkern:D
638       { \xeCJK_make_node:n { CJK-space } }
639       \bool_if:nT
640       {
641         \token_if_macro_p:N \l_peek_token ||
642         ( \l__xeCJK_reserve_space_bool &&
643         \token_if_other_p:N \l_peek_token )
644       }
645       { \__xeCJK_space_or_xecglue: }
646     }
647     { \cs_if_exist:NF \l_peek_token \exp_not:N }
648   }
649 }

```

(End definition for \xeCJK_ignore_spaces:w)

```

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

```

\xeCJK_CJK_and_CJK:N

```

651 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_CJK:N #1 { \CJKglue \CJKsymbol {#1} }
(End definition for \xeCJK_CJK_and_CJK:N)
652 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
653 { \xeCJK_FullLeft_and_CJK: \CJKsymbol }
654 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
655 { \xeCJK_FullRight_and_CJK: \CJKsymbol }
656 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
657 {
658   \clist_map_inline:nn { FullLeft , FullRight }
659   {

```

```

660         \xeCJK_inter_class_toks:nxx {#1} {##1}
661         { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
662         \xeCJK_inter_class_toks:nnc {##1} {#1} { xeCJK_##1_and_Default: }
663     }
664 }

665 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
666 { \xeCJK_Boundary_and_FullLeft:N }
667 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
668 { \xeCJK_Boundary_and_FullRight:N }

\xeCJK_FullRight_and_Boundary:
669 \xeCJK_app_inter_class_toks:nnn { FullLeft } { Boundary } { \tex_ignorespaces:D }
670 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
671 { \xeCJK_FullRight_and_Boundary: }
(End definition for \xeCJK_FullRight_and_Boundary:)

\xeCJK_FullRight_and_Boundary:
672 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Boundary:
673 { \xeCJK_FullRight_and_Default: \tex_ignorespaces:D }
(End definition for \xeCJK_FullRight_and_Boundary:)

674 \clist_map_inline:nn { CJK , FullLeft , FullRight }
675 {
676     \clist_map_inline:nn { FullLeft , FullRight }
677     { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
678 }

__xeCJK_punct_rule:NN 用于抹去标点符号的左/右空白。
679 \cs_new_protected_nopar:Npn __xeCJK_punct_rule:NN #1#2
680 {
681     \tex_vrule:D
682     width - \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
683     depth \c_zero_dim
684     height \c_zero_dim \scan_stop:
685 }
(End definition for __xeCJK_punct_rule:NN)

__xeCJK_punct_glue:NN 根据所选的标点处理方式在标点符号左/右增加的空白。
686 \cs_new_protected_nopar:Npn __xeCJK_punct_glue:NN #1#2
687 {
688     \__xeCJK_punct_hskip:n
689     {
690         \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
691         minus \dim_eval:n { ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} ) / \c_two }
692     }
693 }
694 \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。
695 \cs_new_protected_nopar:Npn __xeCJK_punct_kern:NN #1#2
696 { \tex_kern:D \__xeCJK_use_punct_dim:nnn { kern } {#1} {#2} }
(End definition for __xeCJK_punct_kern:NN)

\g__xeCJK_last_punct_tl 用于记录当前的标点符号。
697 \tl_new:N \g__xeCJK_last_punct_tl
(End definition for \g__xeCJK_last_punct_tl)

```

\xeCJK_FullLeft_and_CJK:

```
698 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_CJK:
699 {
700   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
701   {
702     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
703     \xeCJK_no_break:
704     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
705   }
706   { \xeCJK_no_break: }
707 }
(End definition for \xeCJK_FullLeft_and_CJK:)
```

\xeCJK_FullLeft_and_Default:

```
708 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_Default:
709 {
710   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
711   {
712     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
713     \xeCJK_class_group_end: \xeCJK_no_break:
714     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
715   }
716   { \xeCJK_class_group_end: \xeCJK_no_break: }
717 }
(End definition for \xeCJK_FullLeft_and_Default:)
```

\xeCJK_FullRight_and_CJK:

```
718 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_CJK:
719 {
720   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
721   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
722   \CJKglue
723 }
(End definition for \xeCJK_FullRight_and_CJK:)
```

\xeCJK_FullRight_and_Default:

```
724 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Default:
725 {
726   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
727   \xeCJK_class_group_end:
728   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
729 }
(End definition for \xeCJK_FullRight_and_Default:)
```

\xeCJK_Default_and_FullLeft:nN

```
730 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullLeft:nN #1#2
731 {
732   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#2}
733   \__xeCJK_Default_and_FullLeft_glue:N {#2}
734   \xeCJK_class_group_begin:
735   \xeCJK_select_font:
736   \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
737   \xeCJK_clear_Boundary_and_CJK_toks:
738   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
739   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2}
740   \CJKpunctsymbol {#2}
741 }
742 \cs_new_protected_nopar:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
743 { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_Default_and_FullLeft:nN)
```

\xeCJK_CJK_and_FullLeft:N

```
744 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
745 {
746   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
747   \__xeCJK_CJK_and_FullLeft_glue:N {#1}
748   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
749   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
750   \CJKpunctsymbol {#1}
751 }
752 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
753 { \CJKglue \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
(End definition for \xeCJK_CJK_and_FullLeft:N)
```

\xeCJK_Boundary_and_FullLeft:N

```
754 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullLeft:N #1
755 {
756   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
757   \__xeCJK_Boundary_and_FullLeft_glue:N {#1}
758   \xeCJK_class_group_begin:
759   \xeCJK_select_font:
760   \xeCJK_clear_Boundary_and_CJK_toks:
761   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
762   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
763   \CJKpunctsymbol {#1}
764 }
765 \cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
766 {
767   \int_compare:nNf \etex_lastnodetype:D = \c_one
768   { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
769 }
(End definition for \xeCJK_Boundary_and_FullLeft:N)
```

\xeCJK_Default_and_FullRight:nN

```
770 \cs_new_protected_nopar:Npn \xeCJK_Default_and_FullRight:nN #1#2
771 {
772   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#2}
773   \__xeCJK_Default_and_FullRight_glue:N {#2}
774   \xeCJK_class_group_begin:
775   \xeCJK_select_font:
776   \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
777   \xeCJK_clear_Boundary_and_CJK_toks:
778   \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
779   \__xeCJK_punct_if_middle:NT {#2}
780   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2} }
781   \xeCJK_FullRight_symbol:N {#2}
782 }
(End definition for \xeCJK_Default_and_FullRight:nN)
```

\xeCJK_Boundary_and_FullRight:N

```
783 \cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullRight:N #1
784 {
785   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
786   \__xeCJK_Default_and_FullRight_glue:N {#1}
787   \xeCJK_class_group_begin:
788   \xeCJK_select_font:
789   \xeCJK_clear_Boundary_and_CJK_toks:
790   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
791   \__xeCJK_punct_if_middle:NT {#1}
792   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
793   \xeCJK_FullRight_symbol:N {#1}
794 }
(End definition for \xeCJK_Boundary_and_FullRight:N)
```

\xeCJK_CJK_and_FullRight:N

```
795 \cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
796 {
797   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
798   \__xeCJK_CJK_and_FullRight_glue:N {#1}
799   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
800   \__xeCJK_punct_if_middle:NT {#1}
801   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1} }
802   \xeCJK_FullRight_symbol:N {#1}
803 }
```

(End definition for \xeCJK_CJK_and_FullRight:N)

__xeCJK_CJK_and_FullRight_glue:N
__xeCJK_Default_and_FullRight_glue:N

```
804 \cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
805 {
806   \__xeCJK_punct_if_long:NTF {#1}
807   { \CJKglue }
808   {
809     \__xeCJK_punct_if_middle:NTF {#1}
810     { \xeCJK_no_break: \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1} }
811     { \xeCJK_no_break: }
812   }
813 }
```

814 \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

```
815 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
816 {
817   \xeCJK_no_break:
818   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
819   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
820   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
821   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
822   \CJKpunctsymbol {#1}
823 }
```

(End definition for \xeCJK_FullLeft_and_FullLeft:N)

\xeCJK_FullLeft_and_FullRight:N

```
824 \cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
825 {
826   \xeCJK_no_break:
827   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
828   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
829   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
830   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
831   \xeCJK_no_break:
832   \xeCJK_FullRight_symbol:N {#1}
833 }
```

(End definition for \xeCJK_FullLeft_and_FullRight:N)

\xeCJK_FullRight_and_FullLeft:N

```
834 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
835 {
836   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
837   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
838   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
839   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
840   \__xeCJK_punct_break:
841   \CJKpunctsymbol {#1}
842 }
```

(End definition for \xeCJK_FullRight_and_FullLeft:N)

\xeCJK_FullRight_and_FullRight:N

```
843 \cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
844 {
845   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
846   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
847   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
848   \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
849   \xeCJK_no_break:
850   \xeCJK_FullRight_symbol:N {#1}
851 }
```

(End definition for \xeCJK_FullRight_and_FullRight:N)

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```
852 \keys_define:nn { xecjk / options }
853 {
854   CheckFullRight .choice: ,
855   CheckFullRight / true .code:n =
856   {
857     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
858     {
859       \cs_set_eq:NN \__xeCJK_save_FullRight_check: \xeCJK_FullRight_and_Boundary:
860       \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N \xeCJK_FullRight_symbol:N
861       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
862       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \xeCJK_check_FullRight_symbol:Nw
863     }
864   } ,
865   CheckFullRight / false .code:n =
866   {
867     \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
868     {
869       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \__xeCJK_save_FullRight_check:
870       \cs_set_eq:NN \xeCJK_FullRight_symbol:N \__xeCJK_save_FullRight_symbol:N
871     }
872   } ,
873   CheckFullRight .default:n = { true }
874 }
```

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

\xeCJK_FullRight_symbol:N

```
875 \cs_new_nopar:Npn \xeCJK_FullRight_symbol:N { \CJKpunctsymbol }
(End definition for \xeCJK_FullRight_symbol:N)
```

\xeCJK_check_FullRight:

```
876 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight:
877 {
878   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
879   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
880   \tl_case:Non \l_peek_token { \l__xeCJK_no_break_cs_case_tl } { }
881   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
882   \xeCJK_class_group_end:
883 }
884 \cs_generate_variant:Nn \tl_case:Nnn { No }
(End definition for \xeCJK_check_FullRight:)
```

\xeCJK_check_FullRight_symbol:Nw

```
885 \cs_new_protected_nopar:Npn \xeCJK_check_FullRight_symbol:Nw #1
886 { \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

```
887 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
888 {
889   \tl_new:N #2
890   \seq_new:N #3
891   \keys_define:nn { xeCJK / options }
892   {
893     #1 .code:n =
894     {
895       \seq_set_split:Nnn #3 { } {##1}
896       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
897     } ,
898     #1+ .code:n =
899     {
900       \tl_map_inline:nn {##1}
901       { \seq_if_in:NnF #2 {###1} { \seq_put_right:Nn #3 {###1} } }
902       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
903     } ,
904     #1- .code:n =
905     {
906       \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {###1} }
907       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
908     }
909   }
910 }
911 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
912 {
913   \tl_clear:N #1
914   \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
915   #4
916 }
```

(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。

```
917 \xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
918 \l__xeCJK_no_break_cs_case_tl \l__xeCJK_no_break_cs_seq { \xeCJK_no_break: } { }
```

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

\xeCJKnobreak 为保险起见,我们在这里用了一个循环。

```
919 \NewDocumentCommand \xeCJKnobreak { }
920 {
921   \int_zero:N \l__xeCJK_tmpa_int
922   \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
923   {
924     \int_if_even:nTF \l__xeCJK_tmpa_int
925     {
926       \int_incr:N \l__xeCJK_tmpa_int
927       \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
928     }
929     { \skip_add:Nn \l__xeCJK_tmp_skip \tex_lastskip:D }
930     \tex_unskip:D
931   }
932   \xeCJK_no_break:
933   \int_if_even:nF \l__xeCJK_tmpa_int
934   { \skip_horizontal:N \l__xeCJK_tmp_skip }
935 }
```

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

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```

936 \keys_define:nn { xeCJK / options }
937 {
938   CheckSingle .choice: ,
939   CheckSingle / true .code:n =
940   {
941     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
942     {
943       \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
944       \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
945     }
946   } ,
947   CheckSingle / false .code:n =
948   {
949     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
950     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
951   } ,
952   CheckSingle .default:n = { true } ,
953   CJKchecksingle .meta:n = { CheckSingle = true }
954 }

```

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

\xeCJK_check_single:Nw

```

955 \cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
956 {
957   \peek_catcode:NTF \c_catcode_other_token
958   { \xeCJK_check_single:NNw #1 }
959   {
960     \bool_if:nTF
961     {
962       \xeCJK_if_blank_x_p:n { \token_get_arg_spec:N \l_peek_token } &&
963       \exp_args:No \tl_if_single_token_p:n \l_peek_token &&
964       \exp_after:wN \token_if_other_p:N \l_peek_token
965     }
966     { \exp_after:wN \xeCJK_check_single:NNw \exp_after:wN #1 }
967     { \__xeCJK_check_single_save:N #1 }
968   }
969 }

```

(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。

```

970 \cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
971 {
972   \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_other_token
973   {
974     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
975     { \__xeCJK_check_single_space:NN #1#2 }
976     { \__xeCJK_check_single_save:N #1 #2 }
977   }
978   {
979     \group_align_safe_begin:
980     \token_if_cs:NTF \l_peek_token
981     {
982       \group_align_safe_end:
983       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
984       { \xeCJK_check_single_cs:NNn #1#2 { ~ } }
985       { \xeCJK_check_single_cs:NNn #1#2 { } }
986     }
987     {
988       \group_align_safe_end:
989       \bool_if:nTF
990       {
991         \l__xeCJK_plain_equation_bool &&

```

```

992         \token_if_math_toggle_p:N \l_peek_token
993     }
994     {
995         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
996         { \xeCJK_check_single_equation:NNnNw #1 #2 { ~ } }
997         { \xeCJK_check_single_equation:NNnNw #1 #2 { } }
998     }
999     {
1000         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1001         { \__xeCJK_check_single_save:N #1 #2 ~ }
1002         { \__xeCJK_check_single_save:N #1 #2 }
1003     }
1004 }
1005 }
1006 }
(End definition for \xeCJK_check_single:NNw)

```

__xeCJK_check_single_space:NN

```

1007 \cs_new_protected_nopar:Npn \__xeCJK_check_single_space:NN #1#2
1008 {
1009     \xeCJK_if_CJK_class:NTF #2
1010     {
1011         \xeCJK_if_CJK_class:NTF \l_peek_token
1012         { \__xeCJK_check_single_save:N #1 #2 }
1013         { \__xeCJK_check_single_save:N #1 #2 ~ }
1014     }
1015     { \__xeCJK_check_single_save:N #1 #2 ~ }
1016 }
(End definition for \__xeCJK_check_single_space:NN)

```

\xeCJK_check_single_equation:NNnNw

```

1017 \cs_new_protected_nopar:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
1018 {
1019     \peek_catcode:NTF \c_math_toggle_token
1020     {
1021         \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1022         \xeCJK_no_break: #2 #4
1023     }
1024     { \__xeCJK_check_single_save:N #1 #2#3#4 }
1025 }
(End definition for \xeCJK_check_single_equation:NNnNw)

```

\xeCJK_check_single_cs:NNn

```

1026 \cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
1027 {
1028     \tl_case:Non \l_peek_token
1029     { \l__xeCJK_check_single_cs_case_tl }
1030     { \use_iii:nnn }
1031     { \xeCJK_check_single_env:nnNn }
1032     {
1033         \xeCJK_no_break: \__xeCJK_check_single_save:N #1
1034         \xeCJK_no_break: #2
1035     }
1036     { \__xeCJK_check_single_save:N #1 #2#3 }
1037 }
1038 \tl_new:N \l__xeCJK_check_single_cs_case_tl
(End definition for \xeCJK_check_single_cs:NNn)

```

\xeCJK_check_single_env:nnNn

```

1039 \cs_new_protected_nopar:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
1040 {
1041     \str_case:x:non {#4}

```

```

1042     { \l__xeCJK_inline_env_case_tl }
1043     { \use_i:nn }
1044     {#1} {#2}
1045     \scan_stop: #3 {#4}
1046   }
1047   \cs_generate_variant:Nn \str_case:x:nnn { no }
(End definition for \xeCJK_check_single_env:nnNn)

```

NewLineCS

```

1048   \xeCJK_cs_case_keys_define:nNNnn { NewLineCS }
1049   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq { \use_ii:nnn }
1050   {
1051     \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1052     \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1053   }
(End definition for NewLineCS This function is documented on page 3.)

```

EnvCS

```

1054   \xeCJK_cs_case_keys_define:nNNnn { EnvCS }
1055   \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq { \use:n }
1056   {
1057     \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
1058     \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
1059   }
(End definition for EnvCS This function is documented on page 3.)

```

InlineEnv

```

1060   \keys_define:nn { xeCJK / options }
1061   {
1062     InlineEnv .code:n =
1063     {
1064       \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
1065       \__xeCJK_update_inline_env_case_tl:
1066     } ,
1067     InlineEnv+ .code:n =
1068     {
1069       \clist_map_inline:nn {#1}
1070       {
1071         \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
1072         { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
1073       }
1074       \__xeCJK_update_inline_env_case_tl:
1075     } ,
1076     InlineEnv- .code:n =
1077     {
1078       \clist_map_inline:nn {#1}
1079       { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
1080       \__xeCJK_update_inline_env_case_tl:
1081     }
1082   }
1083   \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:

```

1084   \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
1085   {
1086     \tl_clear:N \l__xeCJK_inline_env_case_tl
1087     \seq_map_inline:Nn \l__xeCJK_inline_env_seq
1088     { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { \use_ii:nn } } }
1089   }
1090   \tl_new:N \l__xeCJK_inline_env_case_tl
(End definition for \__xeCJK_update_inline_env_case_tl: This function is documented on page ??.)

```

PlainEquation

```
1091 \keys_define:nn { xeCJK / options }
1092 { 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

```
1093 \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。

```
1094 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
1095 { s > { \TrimSpaces } m > { \TrimSpaces } m }
1096 {
1097   \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
1098   \IfBooleanT {#1} { \xeCJKResetPunctClass }
1099 }
1100 \@onlypreamble \xeCJKDeclareSubCJKBlock
(End definition for \xeCJKDeclareSubCJKBlock This function is documented on page 8.)
```

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

\xeCJKRestoreSubCJKBlock

```
1101 \bool_new:N \l__xeCJK_sub_cancel_bool
1102 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
1103 {
1104   \bool_if:NF \l__xeCJK_sub_cancel_bool
1105   {
1106     \bool_set_true:N \l__xeCJK_sub_cancel_bool
1107     \__xeCJK_sub_restore_or_cancel:x {#2}
1108     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1109   }
1110 }
1111 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
1112 {
1113   \bool_if:NT \l__xeCJK_sub_cancel_bool
1114   {
1115     \bool_set_false:N \l__xeCJK_sub_cancel_bool
1116     \__xeCJK_sub_restore_or_cancel:x {#2}
1117     \IfBooleanT {#1} { \xeCJKResetPunctClass }
1118   }
1119 }
```

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

__xeCJK_sub_restore_or_cancel:n

```
1120 \cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
1121 {
1122   \clist_map_inline:nn {#1}
1123   {
1124     \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
1125     {
1126       \xeCJK_declare_char_class:nx
1127       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
1128       { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
1129     }
1130     { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
1131   }
1132 }
1133 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
1134 \__xeCJK_msg_new:nn { SubBlock-undefined }
1135 {
1136   The~CJK~sub~block~`#1'~is~undefined.\\
```

```

1137     Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
1138     to~declare~it.
1139   }
(End definition for \__xeCJK_sub_restore_or_cancel:n)

```

\xeCJK_declare_sub_char_class:nnn

```

1140 \cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
1141 {
1142   \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
1143   {
1144     \xeCJK_new_class:n { #1/#2 }
1145     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
1146     \xeCJK_new_sub_key:n {#2}
1147   }
1148   \xeCJK_declare_char_class:nn { #1/#2 } {#3}
1149 }
1150 \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

```

1151 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
1152 {
1153   \seq_map_inline:Nn \g__xeCJK_base_class_seq
1154   {
1155     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
1156     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
1157     \str_if_eq:nnTF {##1} { CJK }
1158     {
1159       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
1160       { \__xeCJK_switch_font:nn {#1} {#2} }
1161     }
1162     {
1163       \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
1164       { \CJKsymbol }
1165       { \__xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
1166     }
1167   }
1168   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
1169   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
1170   {
1171     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
1172     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
1173     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
1174     { \__xeCJK_switch_font:nn {#2} {##1} }
1175     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
1176     { \__xeCJK_switch_font:nn {##1} {#2} }
1177   }
1178   \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
1179   \__xeCJK_save_CJK_class:n { #1/#2 }
1180   \clist_map_inline:nn { CJK , FullLeft , FullRight }
1181   {
1182     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
1183     { \__xeCJK_switch_font:nn {#2} {#1} }
1184   }
1185 }
(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。

```

1186 \__xeCJK_msg_new:nn { XeTeX-too-old }

```

```

1187 {
1188   \token_to_str:N \XeTeXglyphbounds \isnotdefined.\
1189   CJK~punctuation~kerning~will~not~be~available.\
1190   You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
1191 }
1192 \cs_if_exist:NF \XeTeXglyphbounds
1193 {
1194   \__xeCJK_error:n { XeTeX-too-old }
1195   \__xeCJK_after_end_preamble:n { \xeCJKsetup { PunctStyle = plain } }
1196 }

```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```

1197 \NewDocumentCommand \xeCJKsetwidth { m m }
1198 { \tl_map_inline:xn {#1} { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#2} } }
1199 \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。

```

1200 \NewDocumentCommand \xeCJKsetkern { m m m }
1201 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
(End definition for \xeCJKsetkern This function is documented on page 9.)

```

\c__xeCJK_left_tl

\c__xeCJK_right_tl

```

1202 \tl_const:Nn \c__xeCJK_left_tl { left }
1203 \tl_const:Nn \c__xeCJK_right_tl { right }
(End definition for \c__xeCJK_left_tl and \c__xeCJK_right_tl)

```

AllowBreakBetweenPuncts 相关选项声明。

KaiMingPunct

LongPunct

MiddlePunct

PunctWidth

```

1204 \keys_define:nn { xeCJK / options }
1205 {
1206   AllowBreakBetweenPuncts .choice: ,
1207   AllowBreakBetweenPuncts / true .code:n =
1208     { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_allow_break: } ,
1209   AllowBreakBetweenPuncts / false .code:n =
1210     { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_no_break: } ,
1211   AllowBreakBetweenPuncts .default:n = { true } ,
1212   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
1213   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
1214   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
1215   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
1216   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
1217   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
1218   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
1219   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
1220   MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
1221   PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl
1222 }

```

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

\g__xeCJK_special_punct_clist

__xeCJK_set_special_punct:nn

__xeCJK_add_special_punct:nn

__xeCJK_sub_special_punct:nn

```

1223 \clist_new:N \g__xeCJK_special_punct_clist
1224 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
1225 \cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
1226 \cs_new_nopar:Npn \__xeCJK_special_punct_tl:n #1#2 { g__xeCJK_special_punct_#1_#2_tl }
1227 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1228 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
1229 \cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
1230 {
1231   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
1232   { \cs_undefine:c { \__xeCJK_special_punct_tl:n {#1} {##1} } }
1233   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
1234   \tl_map_inline:xn {#2}

```

```

1235     {
1236         \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1237         \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1238     }
1239 }
1240 \cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
1241 {
1242     \tl_map_inline:xn {#2}
1243     {
1244         \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
1245         {
1246             \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1247             \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1248         }
1249     }
1250 }
1251 \cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
1252 {
1253     \tl_map_inline:xn {#2}
1254     {
1255         \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
1256         \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
1257     }
1258 }

```

(End definition for \g__xeCJK_special_punct_clist and others.)

```

\__xeCJK_punct_if_right_p:N
\__xeCJK_punct_if_long_p:N
\__xeCJK_punct_if_middle_p:N
\__xeCJK_punct_if_mixed_width_p:N
\__xeCJK_punct_if_right:NTF
\__xeCJK_punct_if_long:NTF
\__xeCJK_punct_if_middle:NTF
\__xeCJK_punct_if_mixed_width:NTF
1259 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
1260 {
1261     \if_int_compare:w \xeCJK_token_value_class:N #1 = \xeCJK_class_num:n { FullRight }
1262     \prg_return_true: \else: \prg_return_false: \fi:
1263 }
1264 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
1265 {
1266     \exp_args:Nc
1267     \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
1268     {
1269         \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
1270         \prg_return_true: \else: \prg_return_false: \fi:
1271     }
1272 }

```

(End definition for __xeCJK_punct_if_right:N and others.)

```

\__xeCJK_punct_dim_csname:nn
\__xeCJK_punct_dim_csname:nnn
\__xeCJK_punct_dim_csname:nnnn
\__xeCJK_use_punct_dim:nnn
\__xeCJK_use_punct_dim:nn
\__xeCJK_gset_punct_dim:nnx
\__xeCJK_gset_punct_dim:nnnx
\__xeCJK_gset_punct_dim:nnnnx
1273 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
1274 { g__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
1275 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn
1276 { \__xeCJK_punct_dim_csname:nnnn { \l_xeCJK_punct_style_tl } }
1277 \cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnnn #1#2#3#4
1278 { g__\l_xeCJK_current_font_tl/#1/#2/#3/#4/tl }
1279 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
1280 { \tl_use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
1281 \cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
1282 { \tl_use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
1283 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnx #1#2#3
1284 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } {#3} }
1285 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnx #1#2#3#4
1286 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } {#4} }
1287 \cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnnx #1#2#3#4#5
1288 { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnnn {#1} {#2} {#3} {#4} } {#5} }

```

(End definition for __xeCJK_punct_dim_csname:nn and others.)

定义标点处理模板。


```

1289 \DeclareObjectType { xeCJK / punctuation } { \c_three }
1290 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_three }
1291 {
1292     enabled-global-setting : boolean = true ,
1293     fixed-punct-width     : length  = \c_max_dim ,
1294     fixed-punct-ratio     : real    = \c_one_fp ,
1295     mixed-punct-width     : length  = \KeyValue { fixed-punct-width } ,
1296     mixed-punct-ratio     : real    = \KeyValue { fixed-punct-ratio } ,
1297     middle-punct-width    : length  = \KeyValue { fixed-punct-width } ,
1298     middle-punct-ratio    : real    = \KeyValue { fixed-punct-ratio } ,
1299     fixed-margin-width    : length  = \c_max_dim ,
1300     fixed-margin-ratio    : real    = \c_one_fp ,
1301     mixed-margin-width    : length  = \KeyValue { fixed-margin-width } ,
1302     mixed-margin-ratio    : real    = \KeyValue { fixed-margin-ratio } ,
1303     middle-margin-width   : length  = \KeyValue { fixed-margin-width } ,
1304     middle-margin-ratio   : real    = \KeyValue { fixed-margin-ratio } ,
1305     add-min-bound-to-margin : boolean = false ,
1306     optimize-margin       : boolean = false ,
1307     margin-minimum        : length  = \c_zero_dim ,
1308     enabled-kerning       : boolean = true ,
1309     min-bound-to-kerning  : boolean = false ,
1310     kerning-total-width   : length  = \c_max_dim ,
1311     kerning-total-ratio   : real    = 0.75 ,
1312     optimize-kerning      : boolean = false ,
1313     same-align-margin     : length  = \c_max_dim ,
1314     same-align-ratio      : real    = \c_zero_fp ,
1315     different-align-margin : length  = \c_max_dim ,
1316     different-align-ratio : real    = \c_zero_fp ,
1317     kerning-margin-width  : length  = \c_max_dim ,
1318     kerning-margin-ratio  : real    = \c_one_fp ,
1319     kerning-margin-minimum : length  = \c_zero_dim
1320 }
1321 \DeclareTemplateCode { xeCJK / punctuation } { basic } { \c_three }
1322 {
1323     enabled-global-setting = \l__xeCJK_enabled_global_setting_bool ,
1324     fixed-punct-width     = \l__xeCJK_fixed_punct_width_dim ,
1325     fixed-punct-ratio     = \l__xeCJK_fixed_punct_ratio_fp ,
1326     mixed-punct-width     = \l__xeCJK_mixed_punct_width_dim ,
1327     mixed-punct-ratio     = \l__xeCJK_mixed_punct_ratio_fp ,
1328     middle-punct-width    = \l__xeCJK_middle_punct_width_dim ,
1329     middle-punct-ratio    = \l__xeCJK_middle_punct_ratio_fp ,
1330     fixed-margin-width    = \l__xeCJK_fixed_margin_width_dim ,
1331     fixed-margin-ratio    = \l__xeCJK_fixed_margin_ratio_fp ,
1332     mixed-margin-width    = \l__xeCJK_mixed_margin_width_dim ,
1333     mixed-margin-ratio    = \l__xeCJK_mixed_margin_ratio_fp ,
1334     middle-margin-width   = \l__xeCJK_middle_margin_width_dim ,
1335     middle-margin-ratio   = \l__xeCJK_middle_margin_ratio_fp ,
1336     add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
1337     optimize-margin       = \l__xeCJK_optimize_margin_bool ,
1338     margin-minimum        = \l__xeCJK_margin_minimum_dim ,
1339     enabled-kerning       = \l__xeCJK_enabled_kerning_bool ,
1340     min-bound-to-kerning  = \l__xeCJK_min_bound_to_kerning_bool ,
1341     kerning-total-width   = \l__xeCJK_kerning_total_width_dim ,
1342     kerning-total-ratio   = \l__xeCJK_kerning_total_ratio_fp ,
1343     optimize-kerning      = \l__xeCJK_optimize_kerning_bool ,
1344     same-align-margin     = \l__xeCJK_same_align_margin_dim ,
1345     same-align-ratio      = \l__xeCJK_same_align_ratio_fp ,
1346     different-align-margin = \l__xeCJK_different_align_margin_dim ,
1347     different-align-ratio = \l__xeCJK_different_align_ratio_fp ,
1348     kerning-margin-width  = \l__xeCJK_kerning_margin_width_dim ,
1349     kerning-margin-ratio  = \l__xeCJK_kerning_margin_ratio_fp ,
1350     kerning-margin-minimum = \l__xeCJK_kerning_margin_minimum_dim

```

```

1351 }
1352 {
1353     \AssignTemplateKeys
1354     \tl_if_blank:nTF {#3}
1355         { \xeCJK_punct_margin_process:NN {#1} {#2} }
1356         { \xeCJK_punct_kerning_process:NN {#2} {#3} }
1357 }

```

\xeCJK_punct_margin_process:NN

```

1358 \cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
1359 {
1360     \tl_set:Nx \l__xeCJK_tmpa_tl { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
1361     \tl_set:Nx \l__xeCJK_tmpb_tl
1362     {
1363         \__xeCJK_use_punct_dim:nnn { bound }
1364         { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl } {#2}
1365     }
1366     \dim_set:Nn \l__xeCJK_tmp_dim
1367     {
1368         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
1369         {
1370             \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
1371             { \use_none:n }
1372             {
1373                 \xeCJK_if_blank_x:nTF \g__xeCJK_punct_width_tl
1374                 { \use:n }
1375                 { \g__xeCJK_punct_width_tl \use_none:n }
1376             }
1377         }
1378         { \use:n }
1379         {
1380             \__xeCJK_punct_if_middle:NTF {#2}
1381             { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
1382             {
1383                 \__xeCJK_punct_if_mixed_width:NTF {#2}
1384                 { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
1385                 { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }
1386             }
1387         }
1388     }
1389     \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2}
1390     {
1391         \dim_max:nn
1392         { \l__xeCJK_margin_minimum_dim }
1393         {
1394             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
1395             {
1396                 \__xeCJK_punct_if_middle:NTF {#2}
1397                 {
1398                     (
1399                         \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1400                     ) / \c_two
1401                 }
1402                 {
1403                     \bool_if:NTF \l__xeCJK_optimize_margin_bool
1404                     { \dim_max:nn { \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl } }
1405                     { \use:n }
1406                     {
1407                         \l__xeCJK_tmp_dim - ( \l__xeCJK_tmpb_tl )
1408                         - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1409                     }
1410                 }
1411             }
1412         }
1413     }

```

```

1412         {
1413             \bool_if:NTF \l__xeCJK_optimize_margin_bool
1414             { \dim_min:nn { \l__xeCJK_tmpa_tl } }
1415             { \use:n }
1416             {
1417                 \__xeCJK_punct_if_middle:NTF {#2}
1418                 {
1419                     \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
1420                     { \l__xeCJK_middle_margin_width_dim }
1421                     {
1422                         \fp_use:N \l__xeCJK_middle_margin_ratio_fp
1423                         \etex_dimexpr:D
1424                         ( \l__xeCJK_tmpa_tl + \l__xeCJK_tmpb_tl ) / \c_two
1425                         \scan_stop:
1426                     }
1427                 }
1428                 {
1429                     \__xeCJK_punct_if_mixed_width:NTF {#2}
1430                     { \__xeCJK_margin_width_or_ratio:n { mixed } }
1431                     { \__xeCJK_margin_width_or_ratio:n { fixed } }
1432                 }
1433             }
1434         }
1435     }
1436 }
1437 }

```

(End definition for \xeCJK_punct_margin_process:NN)

__xeCJK_punct_width_or_ratio:nN

```

1438 \cs_new_nopar:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
1439 {
1440     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
1441     { \use:c { l__xeCJK_#1_punct_width_dim } }
1442     {
1443         \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } = \c_zero_fp
1444         { \c_max_dim }
1445         {
1446             \fp_use:c { l__xeCJK_#1_punct_ratio_fp }
1447             \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
1448         }
1449     }
1450 }

```

(End definition for __xeCJK_punct_width_or_ratio:nN)

__xeCJK_margin_width_or_ratio:n

```

1451 \cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
1452 {
1453     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
1454     { \use:c { l__xeCJK_#1_margin_width_dim } }
1455     {
1456         \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
1457         \etex_dimexpr:D \l__xeCJK_tmpa_tl \scan_stop:
1458     }
1459     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
1460     { + \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl }
1461 }

```

(End definition for __xeCJK_margin_width_or_ratio:n)

\xeCJK_punct_kerning_process:NN

```

1462 \cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
1463 {
1464     \__xeCJK_gset_punct_dim:nnx { kern } {#1} {#2}

```

```

1465 {
1466   \dim_eval:n
1467   {
1468     \bool_if:nTF
1469     {
1470       \l__xeCJK_enabled_global_setting_bool &&
1471       \tl_if_exist:c { g__xeCJK_punct/kern/#1/#2/tl }
1472     }
1473     { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }
1474     {
1475       \bool_if:NTF \l__xeCJK_enabled_kerning_bool
1476       { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
1477       { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
1478     }
1479     - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1480     - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} )
1481   }
1482 }
1483 }

```

(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。

```

1484 \cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
1485 {
1486   \dim_eval:n
1487   {
1488     \__xeCJK_use_punct_dim:nnn
1489     { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
1490     { \c__xeCJK_right_tl } {#1} +
1491     \__xeCJK_use_punct_dim:nnn
1492     { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1493     { \c__xeCJK_left_tl } {#2}
1494   }
1495 }

```

(End definition for __xeCJK_original_kerning_margin:NN)

__xeCJK_calc_kerning_margin:NN

```

1496 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:NN #1#2
1497 {
1498   \dim_max:nn
1499   { \l__xeCJK_kerning_margin_minimum_dim }
1500   {
1501     \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
1502     { \__xeCJK_punct_min_bound:NN {#1} {#2} }
1503     {
1504       \bool_if:NTF \l__xeCJK_optimize_kerning_bool
1505       { \dim_max:nn { \__xeCJK_punct_min_bound:NN {#1} {#2} } }
1506       { \use:n }
1507       {
1508         \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
1509         { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
1510         {
1511           \fp_compare:nNnTF \l__xeCJK_kerning_total_ratio_fp = \c_zero_fp
1512           {
1513             \xeCJK_if_same_class:NNTF {#1} {#2}
1514             { \__xeCJK_kerning_width_or_ratio:nNN { same } }
1515             { \__xeCJK_kerning_width_or_ratio:nNN { different } }
1516           }
1517           {
1518             \__xeCJK_calc_kerning_margin:nNN
1519             {
1520               \fp_use:N \l__xeCJK_kerning_total_ratio_fp

```

```

1521 \etex_dimexpr:D
1522 \__xeCJK_use_punct_dim:nn { width } {#1} +
1523 \__xeCJK_use_punct_dim:nn { width } {#2}
1524 \scan_stop:
1525 }
1526 }
1527 }
1528 {#1} {#2}
1529 }
1530 }
1531 }
1532 }

```

(End definition for __xeCJK_calc_kerning_margin:NN)

__xeCJK_kerning_width_or_ratio:nnN

```

1533 \cs_new_nopar:Npn \__xeCJK_kerning_width_or_ratio:nnN #1#2#3
1534 {
1535   \dim_compare:nNtF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
1536   { \use:c { l__xeCJK_#1_align_margin_dim } \use_none:nn }
1537   {
1538     \fp_compare:nNtF { \use:c { l__xeCJK_#1_align_ratio_fp } } = \c_zero_fp
1539     { \use:n }
1540     { \fp_use:c { l__xeCJK_#1_align_ratio_fp } \use_ii:nn }
1541   }
1542   {
1543     \dim_compare:nNtF { l__xeCJK_kerning_margin_width_dim } < \c_max_dim
1544     { \l__xeCJK_kerning_margin_width_dim \use_none:n }
1545     { \fp_use:N { l__xeCJK_kerning_margin_ratio_fp } \use:n }
1546   }
1547   { \etex_dimexpr:D \__xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
1548 }

```

(End definition for __xeCJK_kerning_width_or_ratio:nnN)

__xeCJK_punct_min_bound:NN

```

1549 \cs_new_nopar:Npn \__xeCJK_punct_min_bound:NN #1#2
1550 {
1551   \dim_max:nn
1552   {
1553     \dim_min:nn
1554     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} }
1555     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
1556   }
1557   {
1558     \dim_min:nn
1559     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#2} }
1560     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
1561   }
1562 }

```

(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。

```

1563 \cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:nnN #1#2#3
1564 {
1565   \dim_eval:n
1566   {
1567     (#1)
1568     - ( \__xeCJK_use_punct_dim:nnn
1569         { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
1570         { \c__xeCJK_left_tl } {#2} )
1571     - ( \__xeCJK_use_punct_dim:nnn
1572         { \__xeCJK_punct_if_right:NTF {#3} { glue } { bound } }
1573         { \c__xeCJK_right_tl } {#3} )

```

```

1574         - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
1575         - ( \__xeCJK_use_punct_dim:nn { dimen } {#3} )
1576     }
1577 }
(End definition for \__xeCJK_calc_kerning_margin:nnn)

```

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

```

1578 \cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
1579 {
1580     \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }
1581     {
1582         \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1583         {
1584             \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim }
1585             \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
1586             \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
1587         }
1588         {
1589             { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
1590             \__xeCJK_punct_if_long:NTF {#2}
1591             { \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim } }
1592             {
1593                 \UseInstance { xeCJK / punctuation }
1594                 { \l_xeCJK_punct_style_tl } {#1} {#2} { }
1595             }
1596         }
1597     }
1598 }
(End definition for \xeCJK_get_punct_bounds:NN)

```

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

```

1599 \cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
1600 {
1601     \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl {#1}
1602     { \xeCJK_glyph_bounds:NN \c_one {#1} }
1603     \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_right_tl {#1}
1604     { \xeCJK_glyph_bounds:NN \c_three {#1} }
1605     \dim_set:Nn \l__xeCJK_tmp_dim
1606     {
1607         ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl {#1} ) +
1608         ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
1609     }
1610     \__xeCJK_gset_punct_dim:nnx { width } {#1}
1611     { \dim_use:N \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
1612     \__xeCJK_gset_punct_dim:nnx { dimen } {#1}
1613     { \dim_eval:n { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim } }
1614     \__xeCJK_punct_if_long:NT {#1}
1615     {
1616         \seq_map_inline:Nn \g__xeCJK_punct_style_seq
1617         {
1618             \__xeCJK_gset_punct_dim:nnnnx {##1} { kern } {#1} {#1}
1619             {
1620                 \bool_if:nTF
1621                 {
1622                     \str_if_eq_p:nn {#1} { ^^^^2025 } ||
1623                     \str_if_eq_p:nn {#1} { ^^^^2026 }
1624                 }
1625                 { \c_zero_dim }
1626                 { - \dim_use:N \l__xeCJK_tmp_dim }
1627             }
1628         }
1629     }
1630 }

```

```

1629     }
1630   }
1631   \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。

```

1632   \cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
1633   {
1634     \dim_use:N \XeTeXglyphbounds #1 ~
1635     \XeTeXcharglyph \xeCJK_token_value_charcode:N #2 \exp_stop_f:
1636   }
(End definition for \xeCJK_glyph_bounds:NN)

```

`\xeCJK_get_punct_kerning:NN` 标点挤压。

```

1637   \cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
1638   {
1639     \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
1640     {
1641       \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
1642       { \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#2} { \c_zero_dim } }
1643       {
1644         \UseInstance { xeCJK / punctuation }
1645         { \l_xeCJK_punct_style_tl } { } {#1} {#2}
1646       }
1647     }
1648   }
1649   \cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
(End definition for \xeCJK_get_punct_kerning:NN)

```

PunctStyle

```

1650   \keys_define:nn { xeCJK / options }
1651   {
1652     PunctStyle .choice: ,
1653     PunctStyle .default:n = { quanjiao } ,
1654     PunctStyle / halfwidth .meta:n = { PunctStyle = banjiao } ,
1655     PunctStyle / fullwidth .meta:n = { PunctStyle = quanjiao } ,
1656     PunctStyle / mixedwidth .meta:n = { PunctStyle = kaiming } ,
1657     PunctStyle / marginkerning .meta:n = { PunctStyle = hangmoban_jiao } ,
1658     PunctStyle / plain .code:n =
1659     { \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl } ,
1660     PunctStyle / unknown .code:n =
1661     {
1662       \IfInstanceExistTF { xeCJK / punctuation } { \l_keys_value_tl }
1663       { \tl_set:Nx \l_xeCJK_punct_style_tl { \l_keys_value_tl } }
1664       { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
1665     }
1666   }
1667   \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
1668   \__xeCJK_msg_new:nn { punct-style-unknown }
1669   {
1670     Punctuation~style~"#1"~is~unknown. \\\
1671     The~available~styles~are~listed~as~follow.\\\
1672     "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
1673   }

```

(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。

```

1674   \NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
1675   {
1676     \IfInstanceExistTF { xeCJK / punctuation } {#1}
1677     { \__xeCJK_warning:nx { punct-style-already-defined } {#1} }
1678     { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }

```

```

1679 \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
1680 }
1681 \seq_new:N \g__xeCJK_punct_style_seq
1682 \__xeCJK_msg_new:nn { punct-style-already-defined }
1683 {
1684   Punctuation~style~"#1"~is~already~defined!. \\\
1685   The~existing~style~of~"#1"~will~be~overwritten.\
1686 }
1687 \@onlypreamble \xeCJKDeclarePunctStyle
(End definition for \xeCJKDeclarePunctStyle This function is documented on page 9.)

```

\xeCJKEditPunctStyle 对已有的标点处理风格进行修改。

```

1688 \NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
1689 {
1690   \IfInstanceExistTF { xeCJK / punctuation } {#1}
1691     { \EditInstance { xeCJK / punctuation } {#1} {#2} }
1692     { \__xeCJK_error:nx { punct-style-unknown } {#1} }
1693 }
1694 \@onlypreamble \xeCJKEditPunctStyle
(End definition for \xeCJKEditPunctStyle This function is documented on page 9.)
默认设置即为全角格式。

1695 \xeCJKDeclarePunctStyle { quanjiao } { }
1696 \xeCJKDeclarePunctStyle { hangmobanjiang } { enabled-kerning = false }
1697 \xeCJKDeclarePunctStyle { banjiao }
1698 {
1699   fixed-punct-ratio = 0.5 ,
1700   optimize-margin   = true ,
1701   kerning-total-ratio = 0.5 ,
1702   optimize-kerning   = true
1703 }
1704 \xeCJKDeclarePunctStyle { kaiming }
1705 {
1706   fixed-punct-ratio = 0.5 ,
1707   mixed-punct-ratio = 0.8 ,
1708   optimize-margin   = true ,
1709   kerning-total-ratio = 0.5 ,
1710   optimize-kerning   = true
1711 }
1712 \xeCJKDeclarePunctStyle { CCT }
1713 {
1714   fixed-punct-ratio = 0.7 ,
1715   optimize-margin   = true ,
1716   kerning-total-ratio = 0.6 ,
1717   optimize-kerning   = true
1718 }

```

5.11 后备字体

AutoFallback 后备字体的宏包选项声明。

```

1719 \keys_define:nn { xeCJK / options }
1720 {
1721   AutoFallback .choice: ,
1722   AutoFallback / true .code:n =
1723   {
1724     \cs_if_eq:NNF \CJKsymbol \xeCJK_fallback_test_glyph:N
1725     {
1726       \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
1727       \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
1728     }
1729   } ,

```



```

1730     AutoFallback / false .code:n =
1731     {
1732         \cs_if_eq:NNT \CJKsymbol \xeCJK_fallback_test_glyph:N
1733         { \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N }
1734     } ,
1735     AutoFallback .default:n = { true } ,
1736     fallback .meta:n = { AutoFallback = true }
1737 }

```

(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。

```

1738 \cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
1739 {
1740     \xeCJK_glyph_if_exist:NTF {#1}
1741     { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1742     { { \xeCJK_fallback_loop:Nn {#1} { \l_xeCJK_family_tl/FallBack } } }
1743 }

```

(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。

```

1744 \cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
1745 {
1746     \xeCJK_family_if_exist:XTF {#2}
1747     {
1748         \xeCJK_select_font:x {#2}
1749         \xeCJK_glyph_if_exist:NTF {#1}
1750         { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
1751         { \xeCJK_fallback_loop:Nn {#1} { #2/FallBack } }
1752     }
1753     {
1754         \xeCJK_family_if_exist:XT { \l_xeCJK_family_tl/FallBack }
1755         { \__xeCJK_warning:nxx { fallback } {#1} {#2} }
1756         \__xeCJK_fallback_save_CJKsymbol:N {#1}
1757     }
1758 }
1759 \__xeCJK_msg_new:nn { fallback }
1760 {
1761     CJKfamily~`#2'~
1762     ( \prop_get:No \g__xeCJK_family_font_name_prop {#2} )~
1763     does~not~contain~glyph~`#1'~(U+\int_to_hexadecimal:n {'#1}).\\
1764 }
1765 \bool_new:N \l__xeCJK_fallback_first_bool
1766 \cs_generate_variant:Nn \prop_get:Nn { No }

```

(End definition for \xeCJK_fallback_loop:Nn)

\setCJKfallbackfamilyfont

```

1767 \NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
1768 {
1769     \clist_set:Nx \l__xeCJK_tmpa_clist {#3}
1770     \int_compare:nNnTF { \clist_count:N \l__xeCJK_tmpa_clist } > \c_one
1771     { \xeCJK_set_family_fallback:xxN {#1} {#2} \l__xeCJK_tmpa_clist }
1772     { \xeCJK_set_family:xxx { #1/FallBack } {#2} {#3} }
1773 }

```

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

\xeCJK_set_family_fallback:nnN

```

1774 \cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnN #1#2#3
1775 {
1776     \group_begin:
1777     \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
1778     \prop_get:NVNF \g__xeCJK_family_font_name_prop
1779     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl

```

```

1780     { \tl_clear:N \l__xeCJK_font_name_tl }
1781 \clist_map_inline:Nn #3
1782 {
1783     \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /Fallback }
1784     \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
1785     \clist_put_left:cn
1786     { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist } {#2}
1787     \xeCJK_set_family:Vvv \l__xeCJK_fallback_family_tl
1788     { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist }
1789     { \l__xeCJK_ \l__xeCJK_fallback_family_tl _font_name_tl }
1790 }
1791 \group_end:
1792 }
1793 \cs_generate_variant:Nn \xeCJK_set_family_fallback:nnN { xx }
(End definition for \xeCJK_set_family_fallback:nnN)

```

5.12 CJK 字体族声明方式

```

1794 \bool_new:N \g__xeCJK_auto_fake_bold_bool
1795 \bool_new:N \g__xeCJK_auto_fake_slant_bool
1796 \fp_new:N \g__xeCJK_embolden_factor_fp
1797 \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。

```

1798 \keys_define:nn { xeCJK / options }
1799 {
1800     AutoFakeBold .choices:nn = { true , false }
1801     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,
1802     AutoFakeBold / unknown .code:n =
1803     {
1804         \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
1805         \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1806     } ,
1807     AutoFakeBold .default:n = { true } ,
1808     AutoFakeSlant .choices:nn = { true , false }
1809     { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
1810     AutoFakeSlant / unknown .code:n =
1811     {
1812         \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
1813         \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
1814     } ,
1815     AutoFakeSlant .default:n = { true } ,
1816     EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
1817     SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
1818     BoldFont .meta:n = { AutoFakeBold = true } ,
1819     boldfont .meta:n = { AutoFakeBold = true } ,
1820     SlantFont .meta:n = { AutoFakeSlant = true } ,
1821     slantfont .meta:n = { AutoFakeSlant = true }
1822 }

```

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

\xeCJK_new_sub_key:n
\g__xeCJK_sub_key_seq

用于定义 CJK 子区字体和备用字体的选项。

```

1823 \seq_new:N \g__xeCJK_sub_key_seq
1824 \cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
1825 {
1826     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
1827     \keys_define:nn { xeCJK / features }
1828     {
1829         #1 .code:n =
1830         {
1831             \tl_if_blank:nTF {##1}
1832             {

```

```

1833         \bool_set_false:c { l__xeCJK_copy_#1_bool }
1834         \bool_set_false:c { l__xeCJK_add_#1_bool }
1835         \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
1836     }
1837     {
1838         \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
1839         \str_if_eq:nnTF {##1} { * }
1840         {
1841             \bool_set_true:c { l__xeCJK_copy_#1_bool }
1842             \bool_set_false:c { l__xeCJK_add_#1_bool }
1843         }
1844         {
1845             \bool_set_false:c { l__xeCJK_copy_#1_bool }
1846             \bool_set_true:c { l__xeCJK_add_#1_bool }
1847             \__xeCJK_get_sub_features:nn {#1} {##1}
1848         }
1849     }
1850 },
1851 #1 .default:n = { }
1852 }
1853 }

```

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

__xeCJK_get_sub_features:nn

__xeCJK_get_sub_features:wn

```

1854 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
1855 {
1856     \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
1857     \clist_clear:N \l__xeCJK_sub_font_options_clist
1858     \tl_if_head_eq_meaning:VNTF \l__xeCJK_tmpa_tl [ % ]
1859     {
1860         \exp_after:wN \__xeCJK_get_sub_features:wn \l__xeCJK_tmpa_tl \c_empty_tl
1861         \tl_if_blank:VT \l__xeCJK_sub_font_name_tl
1862         {
1863             \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl
1864             \clist_clear:N \l__xeCJK_sub_font_options_clist
1865         }
1866     }
1867     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl }
1868     \tl_if_blank:VTF \l__xeCJK_sub_font_name_tl
1869     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
1870     { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
1871     \clist_set_eq:cN { l__xeCJK_#1_font_options_clist } \l__xeCJK_sub_font_options_clist
1872     \tl_set_eq:cN { l__xeCJK_#1_font_name_tl } \l__xeCJK_sub_font_name_tl
1873 }
1874 \cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:wn [#1] #2
1875 {
1876     \clist_set:Nn \l__xeCJK_sub_font_options_clist {#1}
1877     \str_if_eq:nnTF {#2} { \c_empty_tl }
1878     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \c_empty_tl }
1879     { \tl_set:Nn \l__xeCJK_sub_font_name_tl {#2} }
1880 }
1881 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
1882 \cs_generate_variant:Nn \tl_if_head_eq_meaning:nNTF { V }
1883 \cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }

```

(End definition for __xeCJK_get_sub_features:nn and __xeCJK_get_sub_features:wn)

FallBack

```

1884 \xeCJK_new_sub_key:n { FallBack }

```

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

\g__xeCJK_features_id_prop

BoldFont

ItalicFont

BoldItalicFont

SlantedFont

BoldSlantedFont

调用字体的属性声明,同 fontspec 宏包。

```

1885 \prop_new:N \g__xeCJK_features_id_prop

```

```

1886 \prop_put:Nnn \g__xeCJK_features_id_prop { bf } { Bold }
1887 \prop_put:Nnn \g__xeCJK_features_id_prop { it } { Italic }
1888 \prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
1889 \prop_put:Nnn \g__xeCJK_features_id_prop { sl } { Slanted }
1890 \prop_put:Nnn \g__xeCJK_features_id_prop { bfsl } { BoldSlanted }
1891 \prop_map_inline:Nn \g__xeCJK_features_id_prop
1892 {
1893   \keys_define:nn { xeCJK / features }
1894   {
1895     #2Font .tl_set:c = { l__xeCJK_font_name_#1_tl } ,
1896     #2Features .clist_set:c = { l__xeCJK_font_feat_#1_clist }
1897   }
1898 }
(End definition for \g__xeCJK_features_id_prop and others.)

```

AutoFakeBold
AutoFakeSlant

```

1899 \keys_define:nn { xeCJK / features }
1900 {
1901   AutoFakeBold .choice: ,
1902   AutoFakeBold / false .code:n =
1903     { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
1904   AutoFakeBold / unknown .code:n =
1905     {
1906       \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
1907       \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
1908     } ,
1909   AutoFakeBold .default:n = { \g__xeCJK_embolden_factor_fp } ,
1910   AutoFakeSlant .choice: ,
1911   AutoFakeSlant / false .code:n =
1912     { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
1913   AutoFakeSlant / unknown .code:n =
1914     {
1915       \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
1916       \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
1917     } ,
1918   AutoFakeSlant .default:n = { \g__xeCJK_slant_factor_fp }
1919 }
(End definition for AutoFakeBold and AutoFakeSlant)

```

__xeCJK_set_family_initial:

```

1920 \cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
1921 {
1922   \int_gincr:N \g__xeCJK_family_int
1923   \prop_clear:N \l__xeCJK_add_font_prop
1924   \prop_map_inline:Nn \g__xeCJK_features_id_prop
1925   {
1926     \tl_clear:c { l__xeCJK_font_name_##1_tl }
1927     \clist_clear:c { l__xeCJK_font_feat_##1_clist }
1928   }
1929   \clist_clear:N \l__xeCJK_fontspec_options_clist
1930   \seq_clear:N \l__xeCJK_sub_key_seq
1931   \seq_map_inline:Nn \g__xeCJK_sub_key_seq
1932   {
1933     \bool_set_false:c { l__xeCJK_copy_##1_bool }
1934     \bool_set_false:c { l__xeCJK_add_##1_bool }
1935   }
1936   \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
1937   \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
1938   \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
1939   \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
1940 }
1941 \int_new:N \g__xeCJK_family_int

```

(End definition for __xeCJK_set_family_initial:)

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

```
1942 \cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
1943 {
1944   \group_begin:
1945   \__xeCJK_set_family_initial:
1946   \tl_set:Nn \l__xeCJK_family_name_tl {#1}
1947   \clist_set:Nn \l__xeCJK_font_options_clist {#2}
1948   \tl_set:Nn \l__xeCJK_font_name_tl {#3}
1949   \clist_concat:NNN \l__xeCJK_font_features_clist
1950   \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
1951   \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_features_clist
1952   \keys_set_known:nVN { xeCJK / features }
1953   \l__xeCJK_font_features_clist \l__xeCJK_fontspec_options_clist
1954   \__xeCJK_parse_features:
1955   \__xeCJK_pass_features:
1956   \__xeCJK_check_family:V \l__xeCJK_family_name_tl
1957   \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
1958   \__xeCJK_save_family_info:
1959   \__xeCJK_set_sub_block_family:
1960   \group_end:
1961 }
1962 \cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , xxx , Vvv }
(End definition for \xeCJK_set_family:nnn)
```

__xeCJK_remove_duplicate_keys:N

```
1963 \cs_new_protected_nopar:Npn \__xeCJK_remove_duplicate_keys:N #1
1964 {
1965   \prop_clear:N \l__xeCJK_tmp_prop
1966   \keyval_parse:NNV \__xeCJK_prop_put_aux:n \__xeCJK_prop_put_aux:nn #1
1967   \clist_clear:N #1
1968   \prop_map_inline:Nn \l__xeCJK_tmp_prop
1969   {
1970     \tl_if_blank:nTF {##2}
1971     { \clist_put_right:Nn #1 {##1} }
1972     { \clist_put_right:Nn #1 { ##1 = {##2} } }
1973   }
1974 }
1975 \cs_generate_variant:Nn \keyval_parse:NNn { NNV }
1976 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
1977 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} { } }
1978 \cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
1979 { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} {#2} }
(End definition for \__xeCJK_remove_duplicate_keys:N)
```

__xeCJK_gset_family_cs:x

```
1980 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_cs:x #1
1981 {
1982   \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n {#1} }
1983   {
1984     \group_begin:
1985     \exp_not:n { \cs_set_eq:NN \__xeCJK_update_family:n \use_none:n }
1986     \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
1987     { \exp_not:V \l__xeCJK_fontspec_options_clist }
1988     { \exp_not:V \l__xeCJK_font_name_tl }
1989     \__xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
1990     \group_end:
1991   }
1992 }
(End definition for \__xeCJK_gset_family_cs:x)
```

__xeCJK_check_family:n

```
1993 \cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
1994 {
1995   \prop_get:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmpa_tl
1996   {
1997     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl
1998     { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
1999     \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmpa_tl }
2000   }
2001 }
2002 \cs_generate_variant:Nn \__xeCJK_check_family:n { V }
2003 \__xeCJK_msg_new:nn { CJKfamily-redef }
2004 { Redefining~CJKfamily~\__xeCJK_msg_family_map:n {#1}'~(#{2}). }
(End definition for \__xeCJK_check_family:n)
```

__xeCJK_add_font:nn

__xeCJK_add_font_if_new:nn

```
2005 \cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
2006 { \prop_put:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2007 \cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
2008 { \prop_put_if_new:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
2009 \cs_generate_variant:Nn \__xeCJK_add_font:nn { nx , nV , nv }
2010 \cs_generate_variant:Nn \__xeCJK_add_font_if_new:nn { nx , nV , nv }
(End definition for \__xeCJK_add_font:nn and \__xeCJK_add_font_if_new:nn)
```

__xeCJK_add_fake_bold:n

__xeCJK_add_fake_slant:n

```
2011 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
2012 {
2013   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2014   { FakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
2015 }
2016 \cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
2017 {
2018   \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
2019   { FakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
2020 }
(End definition for \__xeCJK_add_fake_bold:n and \__xeCJK_add_fake_slant:n)
```

__xeCJK_parse_features:

__xeCJK_parse_features:nn

```
2021 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:
2022 { \prop_map_function:NN \g__xeCJK_features_id_prop \__xeCJK_parse_features:nn }
2023 \cs_new_protected_nopar:Npn \__xeCJK_parse_features:nn #1#2
2024 {
2025   \__xeCJK_if_font_select:nTF {#1}
2026   {
2027     \__xeCJK_add_font:nv {#2} { l__xeCJK_font_name_#1_tl }
2028     \__xeCJK_if_it_or_sl:nTF {#1}
2029     {
2030       \__xeCJK_if_font_select:nF { bf #1 }
2031       { \__xeCJK_add_font_if_new:nv {#2} { l__xeCJK_font_name_#1_tl } }
2032     }
2033     {
2034       \str_if_eq:nnT {#1} { bf }
2035       {
2036         \__xeCJK_if_font_select:nF { bfit }
2037         { \__xeCJK_add_font:nV { BoldItalic } \l__xeCJK_font_name_bf_tl }
2038         \__xeCJK_if_font_select:nF { bfsi }
2039         { \__xeCJK_add_font:nV { BoldSlanted } \l__xeCJK_font_name_bf_tl }
2040       }
2041     }
2042   }
2043   { \__xeCJK_set_fake_font:nn {#1} {#2} }
2044 }
```

(End definition for __xeCJK_parse_features: and __xeCJK_parse_features:nn)

```
\__xeCJK_if_font_select_p:n
  \__xeCJK_if_it_or_sl_p:n 2045 \prg_new_conditional:Npnn \__xeCJK_if_font_select:n #1 { p , T , F , TF }
\__xeCJK_if_font_select:nTF 2046 {
  \__xeCJK_if_it_or_sl:nTF 2047   \tl_if_blank:vTF { l__xeCJK_font_name_#1_tl }
  2048   { \prg_return_false: } { \prg_return_true: }
  2049 }
  2050 \prg_new_conditional:Npnn \__xeCJK_if_it_or_sl:n #1 { p , T , F , TF }
  2051 {
  2052   \if_predicate:w
  2053     \bool_if_p:n { \str_if_eq_p:nn { it } {#1} || \str_if_eq_p:nn { sl } {#1} }
  2054     \prg_return_true: \else: \prg_return_false: \fi:
  2055 }
  2056 \cs_generate_variant:Nn \tl_if_blank:nTF { v }
(End definition for \__xeCJK_if_font_select:n and \__xeCJK_if_it_or_sl:n)
```

```
\__xeCJK_set_fake_font:nn
2057 \cs_new_protected_nopar:Npn \__xeCJK_set_fake_font:nn #1#2
2058 {
2059   \str_if_eq:nnTF {#1} { bf }
2060   { \bool_if:NT \l__xeCJK_auto_fake_bold_bool { \__xeCJK_add_fake_bold:n {#1} } }
2061   {
2062     \bool_if:NTF \l__xeCJK_auto_fake_slant_bool
2063     {
2064       \bool_if:nT
2065       { \__xeCJK_if_it_or_sl_p:n {#1}
2066         || ( \str_if_eq_p:nn {#1} { bfit }
2067           && ! ( \__xeCJK_if_font_select_p:n { it } ) )
2068         || ( \str_if_eq_p:nn {#1} { bfsl }
2069           && ! ( \__xeCJK_if_font_select_p:n { sl } ) )
2070       }
2071       { \__xeCJK_add_fake_slant:n {#1} }
2072     }
2073     { \__xeCJK_if_it_or_sl:nT {#1} { \__xeCJK_map_it_sl:nn {#1} {#2} } }
2074     \bool_if:nT
2075     { \l__xeCJK_auto_fake_bold_bool
2076       && ! ( \__xeCJK_if_it_or_sl_p:n {#1} )
2077       && ! ( \__xeCJK_if_font_select_p:n { bf } )
2078     }
2079     { \__xeCJK_add_fake_bold:n {#1} }
2080   }
2081   \__xeCJK_add_font_if_new:nn {#2} { * }
2082 }
(End definition for \__xeCJK_set_fake_font:nn)
```

```
\__xeCJK_map_it_sl:n
2083 \cs_new_protected_nopar:Npn \__xeCJK_map_it_sl:nn #1#2
2084 {
2085   \__xeCJK_if_map_font_select:nT {#1}
2086   {
2087     \__xeCJK_add_font:nx {#2} { \__xeCJK_get_map_font:n {#1} }
2088     \__xeCJK_if_font_select:nF { bf #1 }
2089     { \__xeCJK_add_font_if_new:nx { Bold #2 } { \__xeCJK_get_map_font:n {#1} } }
2090   }
2091 }
(End definition for \__xeCJK_map_it_sl:n)
```

```
\__xeCJK_get_map_font:n
2092 \cs_new_nopar:Npn \__xeCJK_get_map_font:n #1
2093 { \exp_not:v { l__xeCJK_font_name_ \xeCJK_reverse:nnn {#1} { it } { sl } _tl } }
(End definition for \__xeCJK_get_map_font:n)
```

```

\__xeCJK_if_map_font_select_p:n
\__xeCJK_if_map_font_select:nTF
2094 \prg_new_conditional:Npnn \__xeCJK_if_map_font_select:n #1 { p , T , F , TF }
2095 {
2096   \xeCJK_if_blank_x:nTF { \__xeCJK_get_map_font:n {#1} }
2097   { \prg_return_false: } { \prg_return_true: }
2098 }
(End definition for \__xeCJK_if_map_font_select:n)

\__xeCJK_pass_features:

2099 \cs_new_protected_nopar:Npn \__xeCJK_pass_features:
2100 {
2101   \prop_map_inline:Nn \g__xeCJK_features_id_prop
2102   {
2103     \clist_if_empty:cF { l__xeCJK_font_feat_##1_clist }
2104     {
2105       \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
2106       { ##2Features = { \exp_not:v { l__xeCJK_font_feat_##1_clist } } }
2107     }
2108   }
2109   \prop_map_inline:Nn \l__xeCJK_add_font_prop
2110   { \clist_put_right:Nn \l__xeCJK_fontspec_options_clist { ##1 = {##2} } }
2111 }
(End definition for \__xeCJK_pass_features:)

\g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop
2112 \prop_new:N \g__xeCJK_family_name_prop
2113 \prop_new:N \g__xeCJK_family_font_name_prop
2114 \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:

2115 \cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
2116 {
2117   \group_begin:
2118   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
2119   { \clist_remove_all:Nn \l__xeCJK_font_options_clist {##1} }
2120   \prop_gput:NVV \g__xeCJK_family_font_name_prop
2121   \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
2122   \prop_gput:NVV \g__xeCJK_family_font_options_prop
2123   \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
2124   \group_end:
2125 }
(End definition for \__xeCJK_save_family_info:)

\__xeCJK_set_sub_block_family:

2126 \cs_new_protected_nopar:Npn \__xeCJK_set_sub_block_family:
2127 {
2128   \seq_map_inline:Nn \l__xeCJK_sub_key_seq
2129   {
2130     \tl_set:Nx \l__xeCJK_sub_family_tl { \l__xeCJK_family_name_tl/##1 }
2131     \bool_if:cT { l__xeCJK_copy_##1_bool } { \__xeCJK_copy_sub_family:n {##1} }
2132     \bool_if:cT { l__xeCJK_add_##1_bool }
2133     {
2134       \xeCJK_set_family:Vvv \l__xeCJK_sub_family_tl
2135       { l__xeCJK_##1_font_options_clist } { l__xeCJK_##1_font_name_tl }
2136     }
2137   }
2138 }
2139 \cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
2140 {
2141   \__xeCJK_check_family:V \l__xeCJK_sub_family_tl
2142   \prop_get:NVT \g__xeCJK_family_font_name_prop

```



```

2143     \l__xeCJK_family_name_tl \l__xeCJK_tmpa_tl
2144     {
2145         \prop_gput:NVV \g__xeCJK_family_font_name_prop
2146         \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl
2147     }
2148     \prop_get:NVNT \g__xeCJK_family_font_options_prop
2149     \l__xeCJK_family_name_tl \l__xeCJK_tmpa_clist
2150     {
2151         \clist_remove_all:Nn \l__xeCJK_tmpa_clist { #1 = * }
2152         \prop_gput:NVV \g__xeCJK_family_font_options_prop
2153         \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
2154     }
2155     \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_tl } }
2156     {
2157         \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }
2158         {
2159             \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
2160             { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_tmpa_tl
2161             {
2162                 \__xeCJK_gset_family_nfss_cs:xx
2163                 { \l__xeCJK_sub_family_tl } { \exp_not:N \l__xeCJK_tmpa_tl }
2164             }
2165         }
2166     }
2167 }
(End definition for \__xeCJK_set_sub_block_family:)

```

__xeCJK_copy_family:nn

```

2168 \cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
2169 {
2170     \xeCJK_family_if_exist:xT {#2}
2171     {
2172         \tl_map_inline:nn
2173         {
2174             \g__xeCJK_family_name_prop
2175             \g__xeCJK_family_font_name_prop
2176             \g__xeCJK_family_font_options_prop
2177         }
2178         {
2179             \prop_get:NnNT ##1 {#2} \l__xeCJK_tmpa_tl
2180             { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_tl }
2181         }
2182         \cs_gset_eq:cc
2183         { \__xeCJK_family_nfss_csname:n {#1} }
2184         { \__xeCJK_family_nfss_csname:n {#2} }
2185     }
2186 }
2187 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { xx }
(End definition for \__xeCJK_copy_family:nn)

```

5.13 字体切换

\l_xeCJK_current_font_tl

缓存当前字体的原始格式,以加速编译。

\xeCJK_select_font:

```

2188 \tl_new:N \l_xeCJK_current_font_tl
2189 \tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
2190 \cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
2191 \cs_new_protected_nopar:Npn \xeCJK_select_font:
2192 {
2193     \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
2194     {
2195         \tl_set:Nx \l__xeCJK_current_coord_tl { \l_xeCJK_current_font_tl }
2196         \__xeCJK_family_use:x { \l_xeCJK_family_tl }

```

```

2197         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2198     }
2199 }
2200 \cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
2201 {
2202     \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
2203     {
2204         \tl_set:Nx \l__xeCJK_current_coor_tl { \__xeCJK_font_csname:n {#1} }
2205         \__xeCJK_family_use:x {#1}
2206         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2207     }
2208 }
2209 \cs_new_eq:NN \xeCJK@setfont \xeCJK_select_font:
(End definition for \l__xeCJK_current_font_tl, \xeCJK_select_font:, and \xeCJK_select_font:x)

```

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

```

2210 \cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
2211 {
2212     \str_if_eq:nnF {#1} {#2}
2213     {
2214         \__xeCJK_info:nxx { CJK-block } {#1} {#2}
2215         \str_if_eq:nnTF {#2} { CJK }
2216         { \xeCJK_select_font: }
2217         { \__xeCJK_block_select_font:n {#2} }
2218     }
2219 }
2220 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~`#1'~to~`#2'. }
(End definition for \__xeCJK_switch_font:nn)

```

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

```

2221 \cs_new_protected_nopar:Npn \__xeCJK_block_select_font:n #1
2222 {
2223     \cs_if_exist_use:cF { \__xeCJK_font_csname:n { \l__xeCJK_family_tl/#1 } }
2224     {
2225         \tl_set:Nx \l__xeCJK_current_coor_tl
2226         { \__xeCJK_font_csname:n { \l__xeCJK_family_tl/#1 } }
2227         \xeCJK_family_if_exist:xF { \l__xeCJK_family_tl/#1 }
2228         {
2229             \__xeCJK_copy_family:xx { \l__xeCJK_family_tl/#1 }
2230             {
2231                 \cs_if_exist:cTF
2232                 { \__xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
2233                 { \CJKfamilydefault/#1 } { \l__xeCJK_family_tl }
2234             }
2235         }
2236         \__xeCJK_family_use:x { \l__xeCJK_family_tl/#1 }
2237         \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2238     }
2239 }
(End definition for \__xeCJK_block_select_font:n)

```

__xeCJK_family_csname:n

__xeCJK_family_nfss_csname:n

__xeCJK_family_use:x

__xeCJK_gset_family_nfss_cs:xx

```

2240 \cs_new_nopar:Npn \__xeCJK_family_csname:n #1 { xeCJK/family/#1 }
2241 \cs_new_nopar:Npn \__xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
2242 \cs_new_nopar:Npn \__xeCJK_family_use:x #1 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
2243 \cs_new_protected_nopar:Npn \__xeCJK_gset_family_nfss_cs:xx #1#2
2244 {
2245     \prop_gput:Nxx \g__xeCJK_family_name_prop {#1} {#2}
2246     \cs_gset_protected_nopar:cpx { \__xeCJK_family_nfss_csname:n {#1} }
2247     {
2248         \tl_set:Nx \exp_not:N \f@encoding { \c__xeCJK_encoding_tl }

```

```

2249         \tl_set:Nx \exp_not:N \f@family {#2}
2250         \exp_not:N \selectfont
2251     }
2252 }
2253 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
(End definition for \__xeCJK_family_csname:n and others.)

```

\xeCJK_family_if_exist:xTF

```

2254 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
2255 {
2256     \cs_if_exist:cTF { \__xeCJK_family_nfss_csname:n {#1} }
2257     { \use_i:nn }
2258     { \cs_if_exist_use:cTF { \__xeCJK_family_csname:n {#1} } }
2259     { \prg_return_true: } { \prg_return_false: }
2260 }
(End definition for \xeCJK_family_if_exist:xTF)

```

\CJKfamily 用于切换 CJK 字体族。

```

2261 \NewDocumentCommand \CJKfamily { t+ t- m }
2262 {
2263     \xeCJK_if_blank_x:nTF {#3}
2264     {
2265         \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
2266         \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
2267     }
2268     {
2269         \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
2270         {
2271             \xeCJK_family_if_exist:xTF {#3}
2272             {
2273                 \tl_set:Nx \l_xeCJK_family_tl {#3}
2274                 \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
2275                 \IfBooleanT {#1} { \__xeCJK_family_use:x {#3} }
2276             }
2277             { \__xeCJK_family_unknown_warning:x {#3} }
2278         }
2279     }
2280     \tex_ignorespaces:D
2281 }
(End definition for \CJKfamily This function is documented on page 5.)

```

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

```

2282 \tl_new:N \l_xeCJK_family_tl
2283 \tl_set:Nn \l_xeCJK_family_tl { \CJKfamilydefault }
(End definition for \l_xeCJK_family_tl)

```

__xeCJK_gobble_CJKfamily:

```

2284 \cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
2285 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
2286 \DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }
(End definition for \__xeCJK_gobble_CJKfamily:)

```

\xeCJK_family_if_exist_use:x

```

2287 \cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
2288 {
2289     \xeCJK_family_if_exist:xTF {#1}
2290     { \__xeCJK_family_use:x {#1} }
2291     { \__xeCJK_family_unknown_warning:x {#1} }
2292 }
(End definition for \xeCJK_family_if_exist_use:x)

```

_xeCJK_family_unknown_warning:n

```
2293 \cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:n #1
2294 {
2295   \prop_if_empty:NF \g__xeCJK_family_font_name_prop
2296   {
2297     \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
2298     {
2299       \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
2300       \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
2301     }
2302   }
2303 }
2304 \cs_generate_variant:Nn \__xeCJK_family_unknown_warning:n { x }
2305 \seq_new:N \g__xeCJK_unknown_family_seq
2306 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
2307 {
2308   Unknown~CJK~family~\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\\
2309   Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2310 }
2311 \cs_new_nopar:Npn \__xeCJK_msg_def_family_map:n #1
2312 {
2313   \str_case:x:nnn {#1}
2314   {
2315     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
2316     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
2317     \CJKttdefault { \token_to_str:N \setCJKmonofont }
2318   }
2319   { \token_to_str:N \setCJKfamilyfont {#1} }
2320   [...]{}
2321 }
2322 \cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1
2323 {
2324   \str_case:x:nnn {#1}
2325   {
2326     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
2327     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
2328     \CJKttdefault { \token_to_str:N \CJKttdefault }
2329   }
2330   {#1}
2331 }
```

(End definition for __xeCJK_family_unknown_warning:n)

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

```
\setCJKsansfont 2332 \NewDocumentCommand \setCJKmainfont { O{} m }
\setCJKmonofont 2333 { \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2} }
2334 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
2335 \NewDocumentCommand \setCJKsansfont { O{} m }
2336 { \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2} }
2337 \NewDocumentCommand \setCJKmonofont { O{} m }
2338 { \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2} }
```

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

```
2339 \@onlypreamble \setCJKmainfont
2340 \@onlypreamble \setCJKmathfont
2341 \@onlypreamble \setCJKsansfont
2342 \@onlypreamble \setCJKmonofont
2343 \@onlypreamble \setCJKromanfont
```

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

```
\newCJKfontfamily 2344 \NewDocumentCommand \setCJKfamilyfont { m O{} m }
\CJKfontspec 2345 { \xeCJK_set_family:xxx {#1} {#2} {#3} }
2346 \NewDocumentCommand \newCJKfontfamily { o m O{} m }
2347 {
```

```

2348 \tl_set:Nx \l__xeCJK_tmpa_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
2349 \cs_new_protected_nopar:Npx #2 { \CJKfamily { \l__xeCJK_tmpa_tl } }
2350 \xeCJK_set_family:xxx { \l__xeCJK_tmpa_tl } {#3} {#4}
2351 }
2352 \NewDocumentCommand \CJKfontspec { 0{ } m }
2353 {
2354 \xeCJK_fontspec:xx {#1} {#2}
2355 \tex_ignorespaces:D
2356 }

```

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

\xeCJK_fontspec:nn

```

2357 \cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
2358 {
2359 \tl_set:Nx \l__xeCJK_tmpa_tl { \tl_to_str:n { CJKfontspec/#1/#2/id } }
2360 \tl_if_exist:cF { \l__xeCJK_tmpa_tl }
2361 {
2362 \tl_set:Nx \l__xeCJK_tmpp_tl
2363 { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
2364 \tl_gset_eq:cN { \l__xeCJK_tmpa_tl } \l__xeCJK_tmpp_tl
2365 \xeCJK_set_family:Vnn \l__xeCJK_tmpp_tl {#1} {#2}
2366 }
2367 \exp_args:Nc \CJKfamily { \l__xeCJK_tmpa_tl }
2368 }
2369 \cs_generate_variant:Nn \xeCJK_fontspec:nn { VV , xx }

```

(End definition for \xeCJK_fontspec:nn)

\defaultCJKfontfeatures

分别用于设置 CJK 字体的默认属性和增加当前 CJK 字体的属性。

\addCJKfontfeatures

```

2370 \clist_new:N \g__xeCJK_default_features_clist
2371 \NewDocumentCommand \defaultCJKfontfeatures { m }
2372 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
2373 \@onlypreamble \defaultCJKfontfeatures
2374 \NewDocumentCommand \addCJKfontfeatures { m }
2375 {
2376 \xeCJK_add_font_features:x {#1}
2377 \tex_ignorespaces:D
2378 }
2379 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

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

\xeCJK_add_font_features:n

```

2380 \cs_new_protected_nopar:Npn \xeCJK_add_font_features:n #1
2381 {
2382 \prop_get:NVNTF \g__xeCJK_family_font_name_prop \l_xeCJK_family_tl \l__xeCJK_tmpa_tl
2383 {
2384 \clist_set:Nn \l__xeCJK_tmpa_clist {#1}
2385 \seq_map_function:NN
2386 \g__xeCJK_CJK_sub_class_seq \__xeCJK_parse_sub_class_features:n
2387 \prop_get:NVNT \g__xeCJK_family_font_options_prop
2388 \l_xeCJK_family_tl \l__xeCJK_tmpp_clist
2389 {
2390 \clist_concat:NNN
2391 \l__xeCJK_tmpa_clist \l__xeCJK_tmpp_clist \l__xeCJK_tmpa_clist
2392 }
2393 \xeCJK_fontspec:VV \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_tl
2394 }
2395 { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
2396 }
2397 \cs_generate_variant:Nn \xeCJK_add_font_features:n { x }
2398 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
2399 {
2400 \token_to_str:N \addCJKfontfeature (s)~ignored.\\\

```

```

2401     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
2402 }
(End definition for \xeCJK_add_font_features:n)

```

_xeCJK_parse_sub_class_features:n

```

2403 \cs_new_protected_nopar:Npn \__xeCJK_parse_sub_class_features:n #1
2404 {
2405     \clist_if_in:NnT \l__xeCJK_tmpa_clist {#1}
2406     {
2407         \clist_remove_all:Nn \l__xeCJK_tmpa_clist {#1}
2408         \prop_get:NoNF \g__xeCJK_family_font_name_prop
2409             { \l_xeCJK_family_tl/#1 } \l__xeCJK_tmpb_tl
2410         {
2411             \prop_get:NxNF \g__xeCJK_family_font_name_prop
2412                 { \CJKfamilydefault/#1 } \l__xeCJK_tmpb_tl
2413             { \tl_set_eq:NN \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
2414         }
2415         \clist_set:Nx \l__xeCJK_tmpa_clist
2416             { #1 = { [ \exp_not:V \l__xeCJK_tmpa_clist ] { \exp_not:V \l__xeCJK_tmpb_tl } } }
2417         \clist_map_break:
2418     }
2419 }
2420 \cs_generate_variant:Nn \prop_get:NnNF { Nx }
(End definition for \__xeCJK_parse_sub_class_features:n)

```

在导言区结束的时候,若没有声明 CJK 字体,则给出一个警告。如果 \CJKfamilydefault 没有被更改,则在此时根据西文字体的情况更新 \CJKfamilydefault。如果 \CJKfamilydefault 对应的字体族没有定义,则使用 \CJKrmdefault 作为默认字体族。若 \CJKrmdefault 也没有定义,则使用在导言区设置的第一个 CJK 字体作为默认字体族。最后设置数学字体。

```

2421 \__xeCJK_at_end_preamble:n
2422 {
2423     \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
2424     \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
2425     {
2426         \tl_gset:Nx \CJKfamilydefault
2427         {
2428             \str_case:x:nnn { \familydefault }
2429             {
2430                 { \rmdefault } { \exp_not:N \CJKrmdefault }
2431                 { \sfdefault } { \exp_not:N \CJKsfdefault }
2432                 { \ttdefault } { \exp_not:N \CJKttdefault }
2433             }
2434             { \CJKfamilydefault }
2435         }
2436     }
2437     \cs_undefine:N \__xeCJK_family_default_wrap:n
2438     \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
2439         { \__xeCJK_warning:nx { no-CJKfamily } { \CJKfamilydefault } }
2440         {
2441             \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2442             {
2443                 \tl_set_eq:NN \l__xeCJK_tmpa_tl \CJKfamilydefault
2444                 \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
2445                 { \use:n }
2446                 {
2447                     \xeCJK_family_if_exist:xF { \CJKrmdefault }
2448                     { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
2449                 }
2450             }
2451             \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
2452                 { \prop_map_break:n { \tl_gset:Nn \CJKfamilydefault {#1} } }
2453         }
2454     \__xeCJK_warning:nxx { CJKfamilydefault-undefined }

```

```

2455         { \l__xeCJK_tmpa_tl } { \CJKfamilydefault }
2456     }
2457     \CJKfamily { \CJKfamilydefault }
2458     \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
2459 }
2460 }
2461 \__xeCJK_msg_new:nn { no-CJKfamily }
2462 {
2463     It~seems~that~you~have~not~declare~a~CJKfamily.\\
2464     If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
2465     '\__xeCJK_msg_def_family_map:n {#1}'\\
2466     in~the~preamble~to~declare~the~default~CJKfamily.\\
2467 }
2468 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
2469 {
2470     Undefined~CJK~default~family~'\__xeCJK_msg_family_map:n {#1}'~
2471     has~been~replaced~by~'\__xeCJK_msg_family_map:n {#2}'\\.\\
2472     Try~to~use~'\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
2473 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```

2474 \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。

```

2475 \NewDocumentCommand \setCJKmathfont { O{} m }
2476 { \xeCJK_set_family:xxx { \c__xeCJK_math_tl } {#1} {#2} }
2477 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }

```

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

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

```

2478 \cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
2479 {
2480     \xeCJK_family_if_exist:xF { \c__xeCJK_math_tl }
2481     { \tl_set:Nx \l__xeCJK_tmpa_tl { \c__xeCJK_math_tl } }
2482     {
2483         \xeCJK_family_if_exist:xF { \CJKfamilydefault }
2484         { \tl_set:Nx \l__xeCJK_tmpa_tl { \CJKfamilydefault } }
2485         { \use_none:nnnn }
2486     }
2487     \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_tmppb_tl
2488     {
2489         \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmppb_tl }
2490         \DeclareSymbolFont \c__xeCJK_math_tl
2491         \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \mddefault \shapedefault
2492         \SetSymbolFont \c__xeCJK_math_tl { bold }
2493         \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \bfdefault \shapedefault
2494         \int_const:Nn \c__xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }
2495         \clist_concat:NNN
2496         \l__xeCJK_tmpa_clist \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
2497         \clist_concat:NNN
2498         \l__xeCJK_tmpa_clist \l__xeCJK_tmppa_clist \c__xeCJK_FullRight_chars_clist
2499         \clist_map_inline:Nn \l__xeCJK_tmppa_clist
2500         {
2501             \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
2502             { \c_seven } { \c__xeCJK_math_fam_int }
2503         }
2504     }
2505 }

```

(End definition for \xeCJK_set_mathfont:)

\xeCJK_gset_mathcode:nnnn

```
2506 \cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
2507 {
2508   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
2509   \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
2510   {
2511     \tex_global:D \XeTeXmathcode \l__xeCJK_tmpa_int = #3 ~ #4 \l__xeCJK_tmpa_int
2512     \int_incr:N \l__xeCJK_tmpa_int
2513   }
2514 }
(End definition for \xeCJK_gset_mathcode:nnnn)
```

5.15 抄录环境中的间距调整

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

```
2515 \NewDocumentCommand \xeCJKVerbAddon { }
2516 {
2517   \int_compare:nNf \etex_currentgrouplevel:D = \c_zero
2518   {
2519     \__xeCJK_set_verb_exspace:
2520     \bool_if:NF \l__xeCJK_verb_addon_bool
2521     {
2522       \bool_set_true:N \l__xeCJK_verb_addon_bool
2523       \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
2524       \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
2525       \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
2526       \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
2527       \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
2528       \xeCJKsetup { xCJKecglue = false }
2529       \cs_set_protected_nopar:Npn \CJKglue
2530       { \skip_horizontal:N \g__xeCJK_verb_exspace_skip }
2531       \cs_set_protected_nopar:Npn \CJKecglue
2532       { \skip_horizontal:n { 0.5 \g__xeCJK_verb_exspace_skip } }
2533       \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
2534       \cs_set_eq:NN \xeCJK_ignore_spaces:w \CJKecglue
2535       \skip_set:Nn \tex_rightskip:D { \c_zero_dim plus 1 fil }
2536     }
2537   }
2538 }
2539 \bool_new:N \l__xeCJK_verb_addon_bool
2540 \cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon
(End definition for \xeCJKVerbAddon This function is documented on page 11.)
```

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

```
2541 \__xeCJK_after_preamble:n { \tl_put_right:Nn \verbatim@font { \xeCJKVerbAddon } }
```

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

```
2542 \cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
2543 {
2544   \tl_if_exist:cF { xeCJK/verb/\l__xeCJK_family_tl/\curr@fontshape/\f@size }
2545   {
2546     \group_begin:
2547     \xeCJK_select_font:
2548     \use:x
2549     {
2550       \group_end:
2551       \tl_gset:cx
```



```

2552         \exp_not:n { { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2553     {
2554         \exp_not:N \dim_max:nn { \c_zero_dim }
2555     {
2556         \c_two \tex_fontdimen:D \c_two \tex_font:D -
2557         \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00
2558     }
2559 }
2560 }
2561 }
2562 \skip_gset:Nn \g__xeCJK_verb_exspace_skip
2563 { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
2564 }
2565 \skip_new:N \g__xeCJK_verb_exspace_skip
(End definition for \__xeCJK_set_verb_exspace:)

```

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

```

2566 \cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
2567 { {
2568     \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
2569     { \xeCJK_set_visible_space_font: }
2570     \textvisiblespace
2571 } }
2572 \AtEndOfPackage
2573 {
2574     \tl_put_left:Nn \fontspec_visible_space: { \scan_stop: }
2575     \cs_gset_eq:NN \fontspec_visible_space_fallback: \xeCJK_visible_space_fallback:
2576 }
(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。

```

2577 \cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
2578 {
2579     \tl_set:Nx \l__xeCJK_current_coor_tl { xeCJK/space/\curr@fontshape/\f@size }
2580     \exp_after:wN \__xeCJK_set_visible_space_size:n
2581     \exp_after:wN { \dim_use:N \tex_fontdimen:D \c_two \tex_font:D }
2582     \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
2583 }
2584 \cs_new_protected_nopar:Npn \__xeCJK_set_visible_space_size:n #1
2585 {
2586     \tl_set:Nx \f@encoding { \g_fontspec_encoding_tl }
2587     \tl_set:Nx \f@family { lmtt }
2588     \selectfont
2589     \dim_compare:nNf {#1} = { \tex_fontdimen:D \c_two \tex_font:D }
2590     {
2591         \fontsize
2592         {
2593             \dim_eval:n
2594             {
2595                 \f@size pt *
2596                 \dim_ratio:nn {#1} { \tex_fontdimen:D \c_two \tex_font:D }
2597             }
2598         }
2599         { \f@baselineskip }
2600     \selectfont
2601 }
2602 }

```

(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。

```
2603 \keys_define:nn { xeCJK / options }
2604 {
2605   LocalConfig .choice: ,
2606   LocalConfig / false .code:n =
2607     { \bool_gset_false:N \g__xeCJK_config_bool } ,
2608   LocalConfig / true .code:n =
2609     {
2610       \bool_gset_true:N \g__xeCJK_config_bool
2611       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
2612     } ,
2613   LocalConfig / unknown .code:n =
2614     {
2615       \bool_gset_true:N \g__xeCJK_config_bool
2616       \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
2617     } ,
2618   LocalConfig .default:n = { true }
2619 }
2620 \tl_new:N \g__xeCJK_config_name_tl
2621 \bool_new:N \g__xeCJK_config_bool
```

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

CJKnumber 是否启用 CJKnumb 宏包和首行是否缩进。

indentfirst

```
2622 \keys_define:nn { xeCJK / options }
2623 {
2624   CJKnumber .bool_gset:N = \g__xeCJK_number_bool ,
2625   indentfirst .bool_gset:N = \g__xeCJK_indent_bool ,
2626   normalindentfirst .meta:n = { indentfirst = false }
2627 }
```

(End definition for CJKnumber and indentfirst These functions are documented on page 3.)

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

silent

```
2628 \keys_define:nn { xeCJK / options }
2629 {
2630   quiet .code:n =
2631     {
2632       \msg_redirect_module:nnn { xeCJK } { warning } { info }
2633       \msg_redirect_module:nnn { xeCJK } { info } { none }
2634       \xeCJK_if_package_loaded:nF { fontspec }
2635       { \PassOptionsToPackage { quiet } { fontspec } }
2636     } ,
2637   silent .code:n =
2638     {
2639       \msg_redirect_module:nnn { xeCJK } { warning } { none }
2640       \msg_redirect_module:nnn { xeCJK } { info } { none }
2641       \xeCJK_if_package_loaded:nF { fontspec }
2642       { \PassOptionsToPackage { silent } { fontspec } }
2643     } ,
2644   unknown .code:n =
2645     {
2646       \xeCJK_if_package_loaded:nTF { fontspec }
2647       { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
2648       { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
2649     }
2650 }
2651 \__xeCJK_msg_new:nn { key-unknown }
2652 {
```

```

2653     Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~`#1'.\\
2654     The~key~`#1'~is~being~ignored.
2655   }
(End definition for quiet and silent)

```

5.17 xeCJK 初始化设置

```

\CJKsymbol
\CJKpunctsymbol
2656 \cs_new_nopar:Npn \CJKsymbol      #1 {#1}
2657 \cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}
(End definition for \CJKsymbol and \CJKpunctsymbol)
xeCJK 宏包的初始化设置。

2658 \keys_set:nn { xeCJK / options }
2659 {
2660   CJKglue      = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
2661   CJKecglue    = { ~ } ,
2662   xCJKecglue   = false ,
2663   CheckSingle  = false ,
2664   PlainEquation = false ,
2665   CheckFullRight = false ,
2666   CJKspace     = false ,
2667   CJKmath      = false ,
2668   CJKnumber    = false ,
2669   xeCJKactive  = true ,
2670   LocalConfig  = true ,
2671   indentfirst  = true ,
2672   EmboldenFactor = 4 ,
2673   SlantFactor  = 0.167 ,
2674   PunctStyle   = quanjiao ,
2675   NewLineCS    = { \par \[ } ,
2676   EnvCS        = { \begin \end } ,
2677   NoBreakCS    = { \footnote \footnotemark \nobreak } ,
2678   KaiMingPunct = { ^^^^3002 ^^^^ff0e ^^^^ff1f ^^^^ff01 } ,
2679   LongPunct    = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^2025 ^^^^2026 } ,
2680   MiddlePunct  = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
2681   AllowBreakBetweenPuncts = false
2682 }
2683 \defaultCJKfontfeatures { Script = CJK }
执行宏包选项,并载入 fontspec 宏包。

2684 \ProcessKeysOptions { xeCJK / options }
2685 \RequirePackage { fontspec } [ 2011/09/13 ]

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

2686 \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。

2687 \bool_if:NT \g__xeCJK_indent_bool { \RequirePackage { indentfirst } }
对不能通过 \xeCJKsetup 设置的选项给出警告。

2688 \keys_define:nn { xeCJK / options }
2689 {
2690   LocalConfig .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2691   CJKnumber   .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } } ,
2692   indentfirst .code:n = { \__xeCJK_warning:nx { option-invalid } { \l_keys_key_tl } }
2693 }
2694 \__xeCJK_msg_new:nn { option-invalid }
2695 {
2696   The~`#1'~option~only~can~be~set~in~the~optional~argument~to~the\\
2697   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
2698   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
2699 }

```

```

\CJKrmdefault
\JKsfdefault 2700 \tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
\JKttdefault 2701 \tl_if_exist:NF \JKsfdefault { \tl_gset:Nn \JKsfdefault { sf } }
\CJKfamilydefault 2702 \tl_if_exist:NF \JKttdefault { \tl_gset:Nn \CJKttdefault { tt } }
2703 \tl_new:N \l__xeCJK_family_default_init_tl
2704 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
2705 \tl_gset:Nx \l__xeCJK_family_default_init_tl
2706 {
2707   \exp_not:N \__xeCJK_family_default_wrap:n
2708   {
2709     \tl_if_exist:NTF \CJKfamilydefault
2710       { \exp_not:V \CJKfamilydefault }
2711       { \exp_not:N \CJKrmdefault }
2712   }
2713 }
2714 \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。

```

2715 \NewDocumentCommand \xeCJKsetup { +m }
2716 {
2717   \keys_set:nn { xeCJK / options } {#1}
2718   \tex_ignorespaces:D
2719 }
(End definition for \xeCJKsetup This function is documented on page 2.)

```

\xeCJKsetemboldenfactor

\xeCJKsetslantfactor

```

2720 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
2721 { \xeCJKsetup { EmboldenFactor = {#1} } }
2722 \NewDocumentCommand \xeCJKsetslantfactor { m }
2723 { \xeCJKsetup { SlantFactor = {#1} } }
(End definition for \xeCJKsetemboldenfactor and \xeCJKsetslantfactor)

```

\punctstyle

\xeCJKplainchr

```

2724 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
2725 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }
(End definition for \punctstyle and \xeCJKplainchr)

```

\CJKsetecglue

```

2726 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
2727 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue
(End definition for \CJKsetecglue)

```

\CJKspace

\CJKnospace

```

2728 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
2729 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }
(End definition for \CJKspace and \CJKnospace)

```

\xeCJKallowbreakbetweenpuncts

\xeCJKnobreakbetweenpuncts

```

2730 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
2731 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
2732 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
2733 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }
(End definition for \xeCJKallowbreakbetweenpuncts and \xeCJKnobreakbetweenpuncts)

```

\xeCJKenablefallback

\xeCJKdisablefallback

```

2734 \NewDocumentCommand \xeCJKenablefallback { }
2735 { \xeCJKsetup { AutoFallback = true } }
2736 \NewDocumentCommand \xeCJKdisablefallback { }
2737 { \xeCJKsetup { AutoFallback = false } }
(End definition for \xeCJKenablefallback and \xeCJKdisablefallback)

```

\xeCJKsetcharclass

```
2738 \NewDocumentCommand \xeCJKsetcharclass { m m m }
2739 {
2740   \xeCJK_set_char_class:nnn {#1} {#2} {#3}
2741   \xeCJKResetPunctClass
2742 }
(End definition for \xeCJKsetcharclass)
```

5.18 兼容性修补

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

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

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

```
2743 \cs_if_exist:cT { ? - \token_to_str:N \nobreakspace }
2744 {
2745   \exp_args:NNv \ProvideTextCommandDefault \nobreakspace
2746   { ? - \token_to_str:N \nobreakspace }
2747 }
(End definition for \nobreakspace)

2748 \__xeCJK_msg_new:nn { conflict-package }
2749 {
2750   The~`#1'~package~is~in~conflict~with~xeCJK.  \
2751   Please~do~not~use~it~or~load~it~after~xeCJK.
2752 }
2753 \xeCJK_if_package_loaded:nTF { realscripts }
2754 {
2755   \ifpackagelater { realscripts } { 2010/10/10 } { }
2756   {
2757     \__xeCJK_error:nx { conflict-package }
2758     {
2759       \xeCJK_if_package_loaded:nTF { xltextra }
2760       { xltextra } { realscripts }
2761     }
2762   }
2763 }
2764 {
2765   \cs_new_eq:NN \__xeCJK_dim_max:nn \dim_max:nn
2766   \cs_new_eq:NN \__xeCJK_dim_min:nn \dim_min:nn
2767   \__xeCJK_at_end_preamble:n
2768   {
2769     \xeCJK_if_package_loaded:nT { realscripts }
2770     {
2771       \ifpackagelater { realscripts } { 2010/10/10 } { }
2772       {
2773         \cs_gset_eq:NN \dim_max:nn \__xeCJK_dim_max:nn
2774         \cs_gset_eq:NN \dim_min:nn \__xeCJK_dim_min:nn
2775       }
2776     }
2777   }
}
```

```

2777         \cs_undefine:N \__xeCJK_dim_max:nn
2778         \cs_undefine:N \__xeCJK_dim_min:nn
2779     }
2780 }

```

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

```

2781 \RenewDocumentCommand \fontfamily { m }
2782 {
2783     \tl_set:Nx \f@family {#1}
2784     \__xeCJK_update_family:n
2785     {
2786         \str_case:nnn {#1}
2787         {
2788             { \rmdefault } { \CJKfamily { \CJKrmdefault } }
2789             { \sfdefault } { \CJKfamily { \CJKsfdefault } }
2790             { \ttdefault } { \CJKfamily { \CJKttdefault } }
2791             { \familydefault } { \CJKfamily { \CJKfamilydefault } }
2792         }
2793     { }
2794 }
2795 }
2796 \cs_new_eq:NN \__xeCJK_update_family:n \use:n
(End definition for \fontfamily)

```

`\xeCJK@fix@penalty` 对 $\text{\LaTeX 2}_{\epsilon}$ 内核中的 `\fix@penalty` 被用于诸如 `\textit` 之类的文档字体转换命令的定义之中。这里对它进行补丁的目的是修复其中的倾斜校正, 并使得这些文档命令与紧随其后的汉字之间可以正确的插入 `\CJKecglue` 或者忽略其中的空格。例如 这是 $\text{\emph{强调}}$ 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。

```

2798 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
2799 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
2800 \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。

```

2801 \cs_new_protected_nopar:Npn \xeCJK@italiccorr
2802 {
2803     \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
2804     {
2805         \xeCJK_if_last_node:nTF { default }
2806         {
2807             \tex_unkern:D \tex_unkern:D \@@italiccorr
2808             { \xeCJK_make_node:n { default } }
2809         }
2810         {
2811             \xeCJK_if_last_node:nTF { CJK }
2812             {
2813                 \tex_unkern:D \tex_unkern:D \@@italiccorr
2814                 { \xeCJK_make_node:n { CJK } } \use:n
2815             }
2816             {
2817                 \xeCJK_if_last_node:nTF { CJK-space }
2818                 {
2819                     \tex_unkern:D \tex_unkern:D \@@italiccorr
2820                     { \xeCJK_make_node:n { CJK-space } } \use:n
2821                 }
2822                 { \@@italiccorr \use_none:n }
2823             }

```

\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:)。

```

2824         {
2825             \exp_after:wN \exp_after:wN \exp_after:wN
2826             \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2827             \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2828             \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
2829             \xeCJK_ignore_spaces:w
2830         }
2831     }
2832 }
2833 { \@italiccorr }
2834 }

```

(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。

```

2836 \_xeCJK_after_end_preamble:n
2837 {
2838     \int_compare:nNnF
2839     { \c_three + \seq_count:N \g__xeCJK_new_class_seq } = \xe@alloc@intercharclass
2840     {
2841         \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
2842         {
2843             \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
2844             { \_xeCJK_set_others_toks:n {#1} }
2845         }
2846     }
2847 }
2848 \cs_new_protected_nopar:Npn \_xeCJK_set_others_toks:n #1
2849 {
2850     \int_set:cn { \_xeCJK_class_csname:n { Others } } {#1}
2851     \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
2852     {
2853         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
2854         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
2855         \xeCJK_app_inter_class_toks:nnx {##1} { Others }
2856         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
2857         \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
2858         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
2859         \xeCJK_if_blank_x:nT
2860         { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
2861         {
2862             \xeCJK_copy_inter_class_toks:nnnn
2863             { Others } { Boundary } { Default } { Boundary }
2864         }
2865         \xeCJK_if_blank_x:nT
2866         { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
2867         {
2868             \xeCJK_copy_inter_class_toks:nnnn
2869             { Boundary } { Others } { Boundary } { Default }
2870         }
2871     }
2872 }

```

(End definition for _xeCJK_set_others_toks:n)

单独处理宽度有分歧的几个标点: 包括省略号、破折号、间隔号、引号等中西文混用的符号,保证其命令形式输出的是西文字体。并对一些编码的符号宏包做特殊处理。

```

2873 \_xeCJK_after_preamble:n
2874 {

```

```

2875 \tl_map_inline:nn
2876 {
2877   \textellipsis \textemdash \textperiodcentered \textcentereddot
2878   \textquotelleft \textquoteright \textquotedblleft \textquotedblright
2879 }
2880 { \cs_gset_nopar:Npx #1 { { \makexeCJKinactive \exp_not:o {#1} } } }
2881 \tl_put_left:Nn \tipaencoding { \makexeCJKinactive }
2882 \cs_new_eq:NN \__xeCJK_aux_r:n \r
2883 \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \__xeCJK_aux_r:n {#1} } }
2884 \xeCJK_if_package_loaded:nT { pifont }
2885 {
2886   \RenewDocumentCommand \Pifont { m }
2887   { \makexeCJKinactive \usefont { U } {#1} { m } { n } }
2888 }
2889 }

```

简单处理与 hyperref 宏包的兼容问题。

```

2890 \__xeCJK_after_end_preamble:n
2891 {
2892   \xeCJK_if_package_loaded:nT { hyperref }
2893   {
2894     \tl_gput_right:Nn \pdfstringdefPreHook
2895     {
2896       \__xeCJK_gobble_CJKfamily:
2897       \xeCJK_cs_clear:N \makexeCJKinactive
2898     }
2899   }
2900 }

```

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

```

2901 \__xeCJK_after_end_preamble:n
2902 {
2903   \bool_if:nT
2904   { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
2905   { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
2906 }

```

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

```

2907 \tl_set:cn { ver@CJK.sty } { 9999/99/99 }
2908 \tl_set_eq:cc { ver@CJKulem.sty } { ver@CJK.sty }
2909 \tl_set_eq:cc { ver@CJKpunct.sty } { ver@CJK.sty }

```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```

2910 \cs_if_exist:NF \CJK@ifundefined
2911 { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
2912 \NewDocumentCommand \xeCJKcaption { o m }
2913 {
2914   \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
2915   \use:x
2916   {
2917     \char_set_catcode_letter:n { 64 }
2918     \file_input:n { #2.cpx }
2919     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
2920   }
2921   \XeTeXdefaultencoding "UTF-8"
2922 }

```

(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。

```

2923 \cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
2924 { \RequirePackage { xeCJKfntef } }
2925 {
2926   \__xeCJK_at_end_preamble:n

```



```

2927 {
2928   \xeCJK_if_package_loaded:nF { xeCJKfntef }
2929   {
2930     \xeCJK_if_package_loaded:nTF { CJKfntef }
2931     { \RequirePackage { xeCJKfntef } }
2932     {
2933       \xeCJK_if_package_loaded:nT { ulem }
2934       { \RequirePackage { xeCJKfntef } }
2935     }
2936   }
2937 }
2938 }

```

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。

```

2939 \bool_if:NT \g__xeCJK_number_bool
2940 {
2941   \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
2942   \cs_set_protected:Npn \CJKaddEncHook #1#2
2943   { \cs_set:cpn { __xeCJK_enc_#1_hook: } {#2} }
2944   \cs_set_protected_nopar:Npn \Unicode #1#2
2945   { \tex_char:D \etex_numexpr:D (#1) * \c_two_hundred_fifty_six + (#2) \scan_stop: }
2946   \RequirePackage { CJKnumb }
2947   \cs_if_exist_use:c { __xeCJK_enc_ \CJK@UnicodeEnc _hook: }
2948   \tl_set:Nn \CJK@tenthousand { \Unicode { 78 } { 7 } }
2949   \tl_set:Nn \CJK@hundredmillion { \Unicode { 78 } { 191 } }
2950 }

```

最后引入本地配置文件。

```

2951 \bool_if:NT \g__xeCJK_config_bool
2952 {
2953   \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
2954   \onefilewithoptions
2955   { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
2956 }
2957 </package>

```

5.19 xeCJKfntef

```

2958 <*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。

```

2959 \PassOptionsToPackage { normalem } { ulem }
2960 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
2961 \ProcessOptions \scan_stop:
2962 \@ifpackageloaded { xeCJK }
2963 { }
2964 { \RequirePackage { xeCJK } }
2965 \RequirePackage { ulem }
2966 \RequirePackage { CJKfntef }
2967 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

```

\xeCJK_hook_for_ulem:

```

2968 \cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
2969 {
2970   \bool_if:NF \l__xeCJK_ulem_hook_used_bool
2971   {
2972     \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
2973     \xeCJKsetup { CheckFullRight = false }
2974     \bool_if:NTF \l__xeCJK_ulem_skip_punct_bool
2975     { \cs_set_eq:NN \__xeCJK_ulem_leader_type: \UL@leadtype }
2976     {
2977       \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:

```

```

2978         \xCJK_cs_clear:N \__xCJK_ulem_skip_punct_end:
2979     }
2980     \__xCJK_ulem_initial:
2981     \xCJK_glue_to_skip:nN
2982     {
2983         \cs_set_eq:NN \ \tex_space:D
2984         \cs_set_eq:NN \penalty \tex_penalty:D
2985         \cs_set_eq:NN \hskip \skip_horizontal:N
2986         \CJKglue
2987     } \l__xCJK_ccglue_skip
2988     \xCJK_glue_to_skip:nN
2989     {
2990         \cs_set_eq:NN \ \tex_space:D
2991         \cs_set_eq:NN \penalty \tex_penalty:D
2992         \cs_set_eq:NN \hskip \skip_horizontal:N
2993         \CJKecglue
2994     } \l__xCJK_ecglue_skip
2995     \cs_set_protected_nopar:Npn \CJKglue
2996     { \__xCJK_ulem_glue:n \l__xCJK_ccglue_skip }
2997     \cs_set_protected_nopar:Npn \CJKecglue
2998     { \__xCJK_ulem_glue:n \l__xCJK_ecglue_skip }
2999 }
3000 }
3001 \bool_new:N \l__xCJK_ulem_hook_used_bool
(End definition for \xCJK_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 3003 \cs_set_eq:NN \CJK@UL \CJK@@UL
3004 \tl_replace_once:Nnn \CJK@UL { \ULon }
3005 { \bool_set_true:N \l__xCJK_ulem_skip_punct_bool \ULon }
3006 \tl_replace_once:Nnn \CJK@@UL { \ULon }
3007 { \bool_set_false:N \l__xCJK_ulem_skip_punct_bool \ULon }
3008 \bool_new:N \l__xCJK_ulem_skip_punct_bool
(End definition for \CJK@UL and \CJK@@UL)

```

```

\__xCJK_ulem_skip_punct_begin:
\__xCJK_ulem_skip_punct_end: 3010 \cs_new_protected_nopar:Npn \__xCJK_ulem_skip_punct_begin:
3011 { \xCJK_cs_clear:N \UL@leadtype }
3012 \cs_new_protected_nopar:Npn \__xCJK_ulem_skip_punct_end:
3013 { \cs_set_eq:NN \UL@leadtype \__xCJK_ulem_leader_type: }
3014 \xCJK_cs_clear:N \__xCJK_ulem_leader_type:
(End definition for \__xCJK_ulem_skip_punct_begin: and \__xCJK_ulem_skip_punct_end:)

```

__xCJK_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。

```

3015 \cs_new_protected_nopar:Npn \__xCJK_ulem_initial:
3016 {
3017     \xCJK_swap_cs:NN
3018     \__xCJK_Default_and_FullLeft_glue:N \__xCJK_ulem_Default_and_FullLeft_glue:N
3019     \xCJK_swap_cs:NN
3020     \__xCJK_Default_and_FullRight_glue:N \__xCJK_ulem_Default_and_FullRight_glue:N
3021     \xCJK_swap_cs:NN
3022     \__xCJK_CJK_and_FullLeft_glue:N \__xCJK_ulem_CJK_and_FullLeft_glue:N
3023     \xCJK_swap_cs:NN
3024     \__xCJK_CJK_and_FullRight_glue:N \__xCJK_ulem_CJK_and_FullRight_glue:N
3025     \xCJK_swap_cs:NN
3026     \__xCJK_Boundary_and_FullLeft_glue:N \__xCJK_ulem_Boundary_and_FullLeft_glue:N
3027     \xCJK_swap_cs:NN \__xCJK_punct_hskip:n \__xCJK_ulem_punct_hskip:n
3028     \xCJK_swap_cs:NN \xCJK_FullLeft_and_Default: \__xCJK_ulem_FullLeft_and_Default:
3029     \xCJK_swap_cs:NN \xCJK_FullLeft_and_CJK: \__xCJK_ulem_FullLeft_and_CJK:
3030     \xCJK_swap_cs:NN \xCJK_FullRight_and_Default: \__xCJK_ulem_FullRight_and_Default:
3031     \xCJK_swap_cs:NN \xCJK_FullRight_and_CJK: \__xCJK_ulem_FullRight_and_CJK:

```

```

3032 \xeCJK_swap_cs:NN \xeCJK_CJK_and_CJK:N \_\_xeCJK_ulem_CJK_and_CJK:N
3033 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3034 {
3035   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
3036   {
3037     \str_if_eq:nnTF {##1} {####1}
3038     {
3039       \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
3040       { \_\_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3041       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
3042       { \_\_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
3043     }
3044     {
3045       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
3046       { \_\_xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
3047     }
3048   }
3049 }
3050 }

```

(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。

```

3051 \cs_new_nopar:Npn \xeCJK_if_ulem_patch:TF
3052 {
3053   \if_meaning:w \ \LA@space
3054   \exp_after:wN \use_ii:nn
3055   \else:
3056   \exp_after:wN \use_i:nn
3057   \fi:
3058 }

```

(End definition for \xeCJK_if_ulem_patch:TF)

__xeCJK_ulem_CJK_and_CJK:N

```

3059 \cs_new_protected_nopar:Npn \_\_xeCJK_ulem_CJK_and_CJK:N
3060 {
3061   \xeCJK_if_ulem_patch:TF
3062   {
3063     \xeCJK_class_group_end:
3064     \UL@stop \_\_xeCJK_ulem_ccglue: \UL@start
3065     \_\_xeCJK_ulem_class_group_begin:
3066     \CJKsymbol
3067   }
3068   { \_\_xeCJK_ulem_CJK_and_CJK:N }
3069 }

```

(End definition for __xeCJK_ulem_CJK_and_CJK:N)

__xeCJK_ulem_class_group_begin:

```

3070 \cs_new_protected_nopar:Npn \_\_xeCJK_ulem_class_group_begin:
3071 {
3072   \xeCJK_class_group_begin:
3073   \xeCJK_clear_Boundary_and_CJK_toks:
3074   \xeCJK_select_font:
3075 }

```

(End definition for __xeCJK_ulem_class_group_begin:)

__xeCJK_ulem_between_CJK_blocks:nnN

```

3076 \cs_new_protected_nopar:Npn \_\_xeCJK_ulem_between_CJK_blocks:nnN #1#2
3077 {
3078   \xeCJK_if_ulem_patch:TF
3079   {
3080     \xeCJK_class_group_end:

```

```

3081      \UL@stop \__xeCJK_ulem_ccglue: \UL@start
3082      \xeCJK_class_group_begin:
3083      \xeCJK_clear_Boundary_and_CJK_toks:
3084      \__xeCJK_switch_font:nn {#1} {#2}
3085      \CJKsymbol
3086    }
3087    {
3088      \skip_horizontal:N \l__xeCJK_ccglue_skip
3089      \__xeCJK_switch_font:nn {#1} {#2}
3090      \CJKsymbol
3091    }
3092  }

```

(End definition for __xeCJK_ulem_between_CJK_blocks:nnN)

_xeCJK_ulem_Default_and_FullLeft_glue:N

```

3093 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
3094 {
3095   \xeCJK_if_ulem_patch:TF
3096   {
3097     \UL@stop
3098     \__xeCJK_ulem_skip_punct_begin:
3099     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3100     \UL@start
3101   }
3102   { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
3103 }

```

(End definition for __xeCJK_ulem_Default_and_FullLeft_glue:N)

_xeCJK_ulem_Boundary_and_FullLeft_glue:N

```

3104 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
3105 {
3106   \xeCJK_if_ulem_patch:TF
3107   {
3108     \UL@stop
3109     \__xeCJK_ulem_skip_punct_begin:
3110     \int_compare:nNf \etex_lastnodetype:D = \c_one
3111     { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
3112     \UL@start
3113   }
3114   { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
3115 }

```

(End definition for __xeCJK_ulem_Boundary_and_FullLeft_glue:N)

_xeCJK_ulem_CJK_and_FullLeft_glue:N

```

3116 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
3117 {
3118   \xeCJK_if_ulem_patch:TF
3119   {
3120     \xeCJK_class_group_end:
3121     \UL@stop
3122     \__xeCJK_ulem_skip_punct_begin:
3123     \__xeCJK_ulem_ccglue:
3124     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
3125     \UL@start
3126     \__xeCJK_ulem_class_group_begin:
3127   }
3128   { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
3129 }

```

(End definition for __xeCJK_ulem_CJK_and_FullLeft_glue:N)

__xeCJK_ulem_Default_and_FullRight_glue:N

```
3130 \cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
3131 {
3132   \xeCJK_if_ulem_patch:TF
3133   {
3134     \UL@stop
3135     \__xeCJK_ulem_skip_punct_begin:
3136     \__xeCJK_punct_if_long:NTF {#1}
3137     { \__xeCJK_ulem_ccglue: }
3138     {
3139       \__xeCJK_punct_if_middle:NT {#1}
3140       {
3141         \xeCJK_no_break:
3142         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
3143       }
3144       \xeCJK_no_break:
3145     }
3146     \UL@start
3147   }
3148   { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
3149 }
```

(End definition for __xeCJK_ulem_Default_and_FullRight_glue:N)

__xeCJK_ulem_CJK_and_FullRight_glue:N

```
3150 \cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
3151 {
3152   \xeCJK_if_ulem_patch:TF
3153   {
3154     \xeCJK_class_group_end:
3155     \__xeCJK_ulem_Default_and_FullRight_glue:N {#1}
3156     \__xeCJK_ulem_class_group_begin:
3157   }
3158   { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
3159 }
```

(End definition for __xeCJK_ulem_CJK_and_FullRight_glue:N)

__xeCJK_ulem_FullLeft_and_Default:

```
3160 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_Default:
3161 {
3162   \xeCJK_if_ulem_patch:TF
3163   {
3164     \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3165     {
3166       \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3167       \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3168       \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
3169       \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3170     }
3171     { \xeCJK_class_group_end: \UL@stop }
3172     \__xeCJK_ulem_skip_punct_end:
3173     \xeCJK_no_break:
3174     \UL@start
3175   }
3176   { \__xeCJK_ulem_FullLeft_and_Default: }
3177 }
```

(End definition for __xeCJK_ulem_FullLeft_and_Default:)

__xeCJK_ulem_FullLeft_and_CJK:

```
3178 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_CJK:
3179 {
3180   \xeCJK_if_ulem_patch:TF
3181   {
```

```

3182         \__xeCJK_ulem_FullLeft_and_Default:
3183         \__xeCJK_ulem_class_group_begin:
3184     }
3185     { \__xeCJK_ulem_FullLeft_and_CJK: }
3186 }
(End definition for \__xeCJK_ulem_FullLeft_and_CJK:)

```

__xeCJK_ulem_FullRight_and_Default:

```

3187 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_Default:
3188 {
3189     \xeCJK_if_ulem_patch:TF
3190     {
3191         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3192         \xeCJK_class_group_end:
3193         \UL@stop
3194         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3195         \__xeCJK_ulem_skip_punct_end:
3196         \UL@start
3197     }
3198     { \__xeCJK_ulem_FullRight_and_Default: }
3199 }
(End definition for \__xeCJK_ulem_FullRight_and_Default:)

```

__xeCJK_ulem_FullRight_and_CJK:

```

3200 \cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_CJK:
3201 {
3202     \xeCJK_if_ulem_patch:TF
3203     {
3204         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3205         \xeCJK_class_group_end:
3206         \UL@stop
3207         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3208         \__xeCJK_ulem_ccglue:
3209         \__xeCJK_ulem_skip_punct_end:
3210         \UL@start
3211         \__xeCJK_ulem_class_group_begin:
3212     }
3213     { \__xeCJK_ulem_FullRight_and_CJK: }
3214 }
(End definition for \__xeCJK_ulem_FullRight_and_CJK:)

```

__xeCJK_ulem_punct_hskip:n

```

3215 \cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_hskip:n
3216 {
3217     \xeCJK_if_ulem_patch:TF
3218     { \__xeCJK_ulem_hskip:n }
3219     { \__xeCJK_ulem_punct_hskip:n }
3220 }
(End definition for \__xeCJK_ulem_punct_hskip:n)

```

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

```

\__xeCJK_ulem_ccglue:
\__xeCJK_ulem_hskip:n
3221 \cs_new_protected_nopar:Npn \__xeCJK_ulem_glue:n #1
3222 {
3223     \xeCJK_if_ulem_patch:TF
3224     { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
3225     { \skip_horizontal:n {#1} }
3226 }
3227 \cs_new_protected_nopar:Npn \__xeCJK_ulem_ccglue:
3228 { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders }
3229 \cs_new_protected_nopar:Npn \__xeCJK_ulem_hskip:n #1
3230 {
3231     \int_compare:nNnTF \tex_lastkern:D = \c_three

```

```

3232     { \skip_horizontal:n {#1} }
3233     { \skip_set:Nn \UL@skip {#1} \UL@leaders }
3234   }
(End definition for \__xeCJK_ulem_glue:n, \__xeCJK_ulem_ccglue:, and \__xeCJK_ulem_hskip:n)

```

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

```

3235 \RenewDocumentCommand \CJKUnderdot { m }
3236 {
3237   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3238   \CJK@preUnderdot
3239   \__xeCJK_make_under_symbol:n { \CJK@underdotSkip }
3240   \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3241   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
3242   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3243   #1
3244   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
3245   \cs_set_eq:NN \CJKsymbol \__xeCJK_save_under_dot_CJKsymbol:N
3246   \CJK@postUnderdot
3247   \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
3248   \tex_ignorespaces:D
3249 }
3250 \box_new:N \g__xeCJK_under_symbol_box
(End definition for \CJKUnderdot)

```

\CJKUnderanysymbol

```

3251 \RenewDocumentCommand \CJKUnderanysymbol { m m m }
3252 {
3253   \group_begin:
3254   \hbox_set:Nn \CJK@underdotBox {#2}
3255   \__xeCJK_make_under_symbol:n {#1}
3256   \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
3257   \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
3258   #3
3259   \group_end:
3260   \tex_ignorespaces:D
3261 }
(End definition for \CJKUnderanysymbol)

```

__xeCJK_make_under_symbol:n

```

3262 \cs_new_protected:Npn \__xeCJK_make_under_symbol:n #1
3263 {
3264   \hbox_set:Nn \l__xeCJK_tmp_box { ^^^^4e00 }
3265   \vbox_gset_to_ht:Nnn \g__xeCJK_under_symbol_box \c_zero_dim
3266   {
3267     \skip_vertical:n {#1}
3268     \hbox_to_zero:n
3269     {
3270       \tex_kern:D - \box_wd:N \l__xeCJK_tmp_box
3271       \tex_hss:D \box_use:N \CJK@underdotBox \tex_hss:D
3272     }
3273     \tex_vss:D
3274   }
3275 }
(End definition for \__xeCJK_make_under_symbol:n)

```

__xeCJK_under_CJKsymbol:N

```

3276 \cs_new_protected_nopar:Npn \__xeCJK_under_CJKsymbol:N #1
3277 {
3278   \__xeCJK_save_under_dot_CJKsymbol:N {#1}
3279   \hbox_overlap_left:n { \box_use:N \g__xeCJK_under_symbol_box }
3280   { \xeCJK_make_node:n { CJK } }

```

```

3281 \xeCJK_ignore_spaces:w
3282 }
(End definition for \__xeCJK_under_CJKsymbol:N)

```

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

```

3283 \RenewDocumentEnvironment { CJKfilltwosides } { m }
3284 {
3285   \dim_set:Nn \l__xeCJK_fill_width_dim {#1}
3286   \cs_set_eq:NN \CJKglue \tex_hfil:D
3287   \cs_set_eq:NN \ \__xeCJK_fill_newline:
3288   \mode_if_vertical:T { \tex_indent:D }
3289   \tex_vbox:D \c_group_begin_token
3290   \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
3291   \tex_ignorespaces:D
3292 }
3293 {
3294   \tex_unskip:D
3295   \c_group_end_token
3296   \c_group_end_token
3297 }
3298 \dim_new:N \l__xeCJK_fill_width_dim
(End definition for CJKfilltwosides)

```

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

```

3299 \cs_new_protected_nopar:Npn \__xeCJK_fill_newline:
3300 {
3301   \tex_unskip:D
3302   \c_group_end_token
3303   \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
3304   \tex_ignorespaces:D
3305 }
(End definition for \__xeCJK_fill_newline:)
3306 </fntef>

```

5.20 xeCJK.cfg

```

3307 <*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。

```

3308
3309 </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。.....	40
删除多余的 <code>default-icorr</code> 结点。.....	26
取消 <code>\cprotect</code> 的外部宏限制。.....	72
改用 <code>\indentfirst</code> 宏包处理缩进的问题。.....	67
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。..	64
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。...	58
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑 <code>\spaceskip</code> 不为零的情况。.....	14
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。.....	66
<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。.....	70
<code>\xeCJK_fallback_loop:Nn</code> : 调整备用字体的循环方式。..	49
<code>\xeCJK_glyph_if_exist:N</code> : 改进 <code>fontspec</code> 宏包中定义的 <code>\font_glyph_if_exist:NnTF</code> 。.....	14
<code>\xeCJK_hook_for_ulem::</code> : 简化对 <code>ulem</code> 宏包的兼容补丁。..	73
<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。.....	65
<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。.....	64

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。.....	36
增加小宏包 <code>xeCJKfntef</code> ,用于处理下划线的问题。.....	73
对于与 <code>xltxtra</code> 的冲突给出错误警告。.....	69
<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。.....	35
<code>_xeCJK_set_char_class_eq:n</code> : 交换参数的顺序。....	22
<code>_xeCJK_set_verb_exspace::</code> : 调整间距的计算方法。...	64

<code>\fontfamily</code> : 修改主要 CJK 字体族的自动更新方式。....	70
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。.....	32
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。.....	37
<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。.....	36
<code>\xeCJK_check_single:NNw</code> : 改进定义,减少使用 <code>peek</code> 函数的次数。.....	34
<code>\xeCJK_fallback_loop:Nn</code> : 修正备用字体的不正确警告。..	49
<code>\xeCJK_hook_for_ulem::</code> : 完全处理下划线里的标点符号的有关问题。.....	73
<code>\xeCJK_peek_catcode_ignore_spaces:N</code> : 新增有省略空格标识的 <code>peek</code> 函数。.....	15
<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。.....	17
<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> 。.....	22
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。.....	33

v3.1.2

General: 修正重定义 <code>\CJKfamilydefault</code> 无效的问题,恢复容错能力。.....	62
解决在下划线状态下使用 <code>\makebox</code> 时的错误。.....	75
<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。.....	35
<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。.....	60
<code>\fontfamily</code> : 不将参数完全展开。.....	70
<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。.....	69
<code>\xeCJK_token_value_charcode:N</code> : 考虑 <code>charcode</code> 大于等于 <code>0x10000</code> 的情况。.....	16

代码索引

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

Symbols

<code>\@@italiccorr</code>	2799, 2807, 2813, 2819, 2822, 2833
<code>\@ifpackagelater</code>	19, 22, 2755, 2771
<code>\@ifpackageloaded</code>	2962
<code>\@onefilewithoptions</code>	2954
<code>\@onlypreamble</code>	
... ..	1100, 1687, 1694, 2339, 2340, 2341, 2342, 2343, 2373
<code>\@pkgextension</code>	51
<code>\[</code>	2675
<code>\]</code>	5, 6, 7, 13, 14, 15, 16, 247, 248, 1136, 1188, 1189, 1670, 1671, 1672, 1684, 1685, 1763, 2308, 2400, 2463, 2464, 2465, 2466, 2471, 2653, 2696, 2697, 2750, 3287
<code>_xeCJK_Boundary_and_FullLeft_glue:N</code>	757, 765, 3026
<code>_xeCJK_CJK_and_FullLeft_glue:N</code>	747, 752, 3022
<code>_xeCJK_CJK_and_FullRight_glue:N</code> .	798, 804, 804, 814, 3024
<code>_xeCJK_CJK_class_tl:c</code>	327
<code>_xeCJK_CJK_class_tl:n</code>	198, 201, 203
<code>_xeCJK_Default_and_FullLeft_glue:N</code>	733, 742, 3018
<code>_xeCJK_Default_and_FullRight_glue:N</code>	
.....	773, 786, 804, 814, 3020
<code>_xeCJK_add_fake_bold:n</code>	2011, 2011, 2060, 2079
<code>_xeCJK_add_fake_slant:n</code>	2011, 2016, 2071
<code>_xeCJK_add_font:nV</code>	2037, 2039
<code>_xeCJK_add_font:nn</code>	2005, 2005, 2009
<code>_xeCJK_add_font:nv</code>	2027
<code>_xeCJK_add_font:nx</code>	2087
<code>_xeCJK_add_font_if_new:nn</code>	2005, 2007, 2010, 2081
<code>_xeCJK_add_font_if_new:nv</code>	2031
<code>_xeCJK_add_font_if_new:nx</code>	2089
<code>_xeCJK_add_special_punct:nn</code> .	1213, 1216, 1219, 1223, 1240
<code>_xeCJK_after_end_preamble:n</code> <u>52</u> , 59, 1195, 2836, 2890, 2901	
<code>_xeCJK_after_preamble:n</code>	52, 57, 2541, 2873
<code>_xeCJK_at_end_preamble:n</code>	52, 55, 2421, 2767, 2926
<code>_xeCJK_aux:r:n</code>	2882, 2883
<code>_xeCJK_block_select_font:n</code>	2217, 2221, 2221
<code>_xeCJK_calc_kerning_margin:NN</code>	1476, 1496, 1496
<code>_xeCJK_calc_kerning_margin:nnN</code> ..	1509, 1518, 1563, 1563
<code>_xeCJK_check_family:V</code>	1956, 2141
<code>_xeCJK_check_family:n</code>	1993, 1993, 2002
<code>_xeCJK_check_num_range:nnNN</code>	356, 371, 379, 379, 2508

_xeCJK_check_single_save:N	943, 950, 967, 976, 1001, 1002, 1012, 1013, 1015, 1021, 1024, 1033, 1036	_xeCJK_original_kerning_margin:NN	1477, 1484, 1484, 1547
_xeCJK_check_single_space:NN	975, 1007, 1007	_xeCJK_parse_features:	1954, 2021, 2021
_xeCJK_class_csname:n	224, 227, 230, 235, 238, 243, 243, 244, 327, 330, 359, 1124, 1142, 2850	_xeCJK_parse_features:nn	2021, 2022, 2023
_xeCJK_clear_Boundary_and_CJK_toks:n	450, 451	_xeCJK_parse_sub_class_features:n	2386, 2403, 2403
_xeCJK_copy_family:nn	2168, 2168, 2187	_xeCJK_pass_features:	1955, 2099, 2099
_xeCJK_copy_family:xx	2229	_xeCJK_peek_after_do:w	169, 180
_xeCJK_copy_sub_family:n	2131, 2139	_xeCJK_peek_catcode_false:w	143, 162
_xeCJK_dim_max:nn	2765, 2773, 2777	_xeCJK_peek_catcode_ignore_spaces_branches:w	146, 148, 153
_xeCJK_dim_min:nn	2766, 2774, 2778	_xeCJK_peek_catcode_true:w	142, 159
_xeCJK_document_left_hook:	69, 75	_xeCJK_peek_ignore_spaces_branches:w	171, 173, 177
_xeCJK_document_right_hook:	71, 77	_xeCJK_prop_put_aux:n	1966, 1976
_xeCJK_error:n	36, 37, 1194	_xeCJK_prop_put_aux:nn	1966, 1978
_xeCJK_error:nx	36, 38, 225, 236, 1130, 1664, 1692, 2647, 2757	_xeCJK_punct_break:	840, 1208, 1210
_xeCJK_fallback_save_CJKsymbol:N	1726, 1733, 1741, 1750, 1756	_xeCJK_punct_dim_csname:nn	1273, 1273, 1280, 1284
_xeCJK_family_csname:n	1982, 2155, 2232, 2240, 2240, 2258	_xeCJK_punct_dim_csname:nnn	1273, 1275, 1282, 1286, 1580, 1639
_xeCJK_family_default_wrap:n	2423, 2437, 2704, 2707	_xeCJK_punct_dim_csname:nnnn	1273, 1276, 1277, 1288
_xeCJK_family_nfss_csname:n	1998, 2183, 2184, 2240, 2241, 2242, 2246, 2256	_xeCJK_punct_glue:NN	686, 686, 704, 714, 721, 728, 743, 753, 768, 810, 881, 3099, 3111, 3124, 3142, 3169, 3194, 3207
_xeCJK_family_unknown_warning:n	2293, 2293, 2304	_xeCJK_punct_hskip:n	688, 694, 3027
_xeCJK_family_unknown_warning:x	2277, 2291	_xeCJK_punct_if_long:NT	1614
_xeCJK_family_use:x	2196, 2205, 2236, 2240, 2242, 2275, 2290	_xeCJK_punct_if_long:NTF	806, 1259, 1590, 3136
_xeCJK_fill_newline:	3287, 3299, 3299	_xeCJK_punct_if_long_p:N	1259
_xeCJK_font_csname:n	2189, 2190, 2202, 2204, 2223, 2226	_xeCJK_punct_if_middle:NT	779, 791, 800, 3139
_xeCJK_get_map_font:n	2087, 2089, 2092, 2092, 2096	_xeCJK_punct_if_middle:NTF	700, 710, 809, 1259, 1380, 1396, 1417, 3164
_xeCJK_get_sub_features:Vn	1784	_xeCJK_punct_if_middle_p:N	1259
_xeCJK_get_sub_features:nn	1847, 1854, 1854, 1881	_xeCJK_punct_if_mixed_width:NTF	1259, 1383, 1429
_xeCJK_get_sub_features:wn	1854, 1860, 1874	_xeCJK_punct_if_mixed_width_p:N	1259
_xeCJK_gobble_CJKfamily:	2284, 2284, 2896	_xeCJK_punct_if_right:N	1259
_xeCJK_gobble_CJKfamily:wn	2285, 2286	_xeCJK_punct_if_right:NTF	1259, 1489, 1492, 1569, 1572
_xeCJK_gset_family_cs:x	1957, 1980, 1980	_xeCJK_punct_if_right_p:N	1259
_xeCJK_gset_family_nfss_cs:xx	1989, 2162, 2240, 2243	_xeCJK_punct_kern:NN	695, 695, 820, 829, 838, 847
_xeCJK_gset_punct_dim:nnnx	1273, 1287, 1618	_xeCJK_punct_min_bound:NN	1502, 1505, 1549, 1549
_xeCJK_gset_punct_dim:nnx	1273, 1285, 1389, 1464, 1584, 1585, 1586, 1591, 1601, 1603, 1642	_xeCJK_punct_rule:NN	679, 679, 702, 712, 720, 726, 739, 749, 762, 780, 792, 801, 879, 3167, 3191, 3204
_xeCJK_gset_punct_dim:nnx	1273, 1283, 1610, 1612	_xeCJK_punct_width_or_ratio:nN	1381, 1384, 1385, 1438, 1438
_xeCJK_if_font_select:n	2045	_xeCJK_remove_duplicate_keys:N	1951, 1963, 1963
_xeCJK_if_font_select:nF	2030, 2036, 2038, 2088	_xeCJK_save_CJK_class:n	324, 329, 1179
_xeCJK_if_font_select:nTF	2025, 2045	_xeCJK_save_FullRight_check:	859, 869
_xeCJK_if_font_select_p:n	2045, 2067, 2069, 2077	_xeCJK_save_FullRight_symbol:N	860, 870, 886
_xeCJK_if_it_or_sl:n	2050	_xeCJK_save_family_info:	1958, 2115, 2115
_xeCJK_if_it_or_sl:nT	2073	_xeCJK_save_under_dot_CJKsymbol:N	3240, 3245, 3256, 3278
_xeCJK_if_it_or_sl:nTF	2028, 2045	_xeCJK_set_char_catcode:n	360, 361, 364
_xeCJK_if_it_or_sl_p:n	2045, 2065, 2076	_xeCJK_set_char_class:nnn	369, 369, 393
_xeCJK_if_map_font_select:n	2094	_xeCJK_set_char_class_aux:Nnw	336, 345, 350, 2501
_xeCJK_if_map_font_select:nT	2085	_xeCJK_set_char_class_catcode:nnn	354, 354, 391
_xeCJK_if_map_font_select:nTF	2094	_xeCJK_set_char_class_eq:nn	394, 394, 2523, 2524, 2525, 2526, 2527
_xeCJK_if_map_font_select_p:n	2094	_xeCJK_set_fake_font:nn	2043, 2057, 2057
_xeCJK_info:nx	36, 42	_xeCJK_set_family_initial:	1920, 1920, 1945
_xeCJK_info:nxx	36, 43, 2214	_xeCJK_set_others_toks:n	2836, 2844, 2848
_xeCJK_int_until_do:wn	135, 136, 137, 138	_xeCJK_set_special_punct:nn	1212, 1215, 1218, 1223, 1229
_xeCJK_kerning_width_or_ratio:nNN	1514, 1515, 1533, 1533	_xeCJK_set_sub_block_family:	1959, 2126, 2126
_xeCJK_make_under_symbol:n	3239, 3255, 3262, 3262	_xeCJK_set_sub_class_toks:nn	1145, 1151, 1151
_xeCJK_map_it_sl:n	2083	_xeCJK_set_verb_exspace:	2519, 2542, 2542
_xeCJK_map_it_sl:nn	2073, 2083	_xeCJK_set_visible_space_size:n	2580, 2584
_xeCJK_margin_width_or_ratio:n	1430, 1431, 1451, 1451	_xeCJK_space_or_xecglue:	487, 597, 602, 609, 632, 645
_xeCJK_msg_def_family_map:n	2309, 2311, 2465, 2472	_xeCJK_special_punct_seq:n	1225, 1228, 1231, 1233, 1237, 1244, 1247, 1256
_xeCJK_msg_family_map:n	2004, 2308, 2322, 2470, 2471		
_xeCJK_msg_new:nn	36, 36, 245, 1134, 1186, 1668, 1682, 1759, 2003, 2220, 2306, 2398, 2461, 2468, 2651, 2694, 2748		

\c_xeCJK_punct_style_plain_tl	1582, 1641, 1659, 1667
\c_xeCJK_right_tl	702, 712, 720, 721, 726, 728, 772, 785, 797, 810, 827, 845, 878, 879, 881, 1202, 1203, 1364, 1479, 1490, 1555, 1560, 1573, 1586, 1603, 1608, 3142, 3167, 3191, 3194, 3204, 3207
\c_catcode_other_token	957, 972
\c_eleven	922
\c_empty_tl	1860, 1877, 1878
\c_four	117, 2841
\c_group_begin_token	453, 3289, 3290, 3303
\c_group_end_token	454, 3295, 3296, 3302
\c_math_toggle_token	629, 1019
\c_max_dim	1293, 1299, 1310, 1313, 1315, 1317, 1394, 1419, 1440, 1444, 1453, 1508, 1535, 1543
\c_one	217, 251, 767, 1602, 1770, 2363, 2841, 3110
\c_one_fp	1294, 1300, 1318
\c_seven	2502
\c_space_tl	602
\c_space_token	150, 175
\c_ten	525, 540
\c_ten_thousand	45
\c_three	116, 253, 1289, 1290, 1321, 1604, 2839, 3231
\c_two	115, 252, 691, 1400, 1424, 2556, 2581, 2589, 2596
\c_two_hundred_fifty_five	254
\c_two_hundred_fifty_six	2945
\c_xeCJK_math_fam_int	2494, 2502
\c_xeCJK_space_skip_tl	111, 111, 474, 532
\c_zero	44, 218, 250, 2517, 2803
\c_zero_dim	683, 684, 1307, 1319, 1584, 1585, 1586, 1591, 1625, 1642, 2535, 2554, 2660, 3265
\c_zero_fp	1314, 1316, 1443, 1511, 1538
\c_zero_skip	113
\char_set_catcode:n	2919
\char_set_catcode_ignore:n	219
\char_set_catcode_letter:n	2917
\char_set_catcode_other:n	360
\char_value_catcode:n	2919
\CheckFullRight	852
\CheckSingle	936
\CJK	250
\CJK@@UL	3003, 3003, 3006
\CJK@hundredmillion	2949
\CJK@ifundefined	2910, 2911
\CJK@postUnderdot	3246
\CJK@preUnderdot	3238
\CJK@tenthousand	2948
\CJK@UL	3003, 3003, 3004
\CJK@underdotBox	3254, 3271
\CJK@underdotSkip	3239
\CJK@UnicodeEnc	2941, 2947
\CJKaddEncHook	2942
\CJKkecglue	483, 486, 491, 506, 527, 543, 586, 590, 597, 607, 609, 632, 2531, 2533, 2534, 2993, 2997
xCJKkecglue	3
\CJKfamily	5, 2261, 2261, 2285, 2349, 2367, 2457, 2788, 2789, 2790, 2791
\CJKfamilydefault	6, 2232, 2233, 2283, 2412, 2424, 2426, 2434, 2439, 2441, 2443, 2444, 2448, 2452, 2455, 2457, 2483, 2484, 2700, 2709, 2710, 2714, 2791
\CJKfilltwosides	3283
\CJKfixedspacing	2540
\CJKfontspec	6, 2344, 2352
\CJKglue	520, 550, 577, 581, 651, 722, 753, 807, 2529, 2986, 2995, 3286
\CJKmath	2474
\CJKknoespace	2728, 2729
\CJKknumber	2622
\CJKkpunctsymbol	740, 750, 763, 822, 841, 875, 2656, 2657
\CJKkrmdefault	6, 2315, 2326, 2333, 2430, 2444, 2447, 2448, 2700, 2700, 2711, 2788
\CJKksetecglue	2726, 2726, 2727
\CJKksfdefault	6, 2316, 2327, 2336, 2431, 2700, 2701, 2789
\CJKkspace	615, 2728, 2728
\CJKksymbol	463, 514, 651, 653, 655, 1164, 1165, 1724, 1726, 1727, 1732, 1733, 2656, 2656, 3066, 3085, 3090, 3240, 3241, 3245, 3256, 3257
\CJKktdefault	6, 2317, 2328, 2338, 2432, 2700, 2702, 2790
\CJKkunderanysymbol	3251, 3251
\CJKkunderdot	3235, 3235
\clist_clear:c	1927
\clist_clear:N	1857, 1864, 1929, 1967
\clist_concat:NNN	1949, 2390, 2495, 2497
\clist_const:Nn	258, 260, 262, 263, 270, 284
\clist_count:N	1770
\clist_gconcat:ccN	339
\clist_gconcat:NNN	352
\clist_gset:Nn	1224, 2372
\clist_if_empty:cF	2103
\clist_if_in:NnT	2405
\clist_map_break:	2417
\clist_map_function:nN	329
\clist_map_inline:cn	397
\clist_map_inline:Nn	341, 1227, 1264, 1781, 2499
\clist_map_inline:nn	455, 467, 493, 658, 674, 676, 1069, 1078, 1122, 1180
\clist_new:c	228, 239
\clist_new:N	34, 35, 1223, 2370
\clist_put_left:cn	1785
\clist_put_left:cx	2013, 2018
\clist_put_right:Nn	1971, 1972, 2110
\clist_put_right:Nx	2105
\clist_remove_all:Nn	2119, 2151, 2407
\clist_set:Nn	338, 1876, 1947, 2384
\clist_set:Nx	1769, 2415
\clist_set_eq:cN	1871
\cs:w	104, 2905
\cs_end:	104, 198, 1269, 2905
\cs_generate_variant:Nn	203, 352, 353, 421, 431, 437, 884, 1047, 1133, 1150, 1199, 1631, 1649, 1766, 1793, 1881, 1882, 1883, 1962, 1975, 2002, 2009, 2010, 2056, 2187, 2253, 2304, 2369, 2397, 2420
\cs_gset_eq:cc	2182
\cs_gset_eq:NN	103, 2575, 2773, 2774, 3240, 3256
\cs_gset_nopar:Npx	73, 2880
\cs_gset_protected_nopar:cpx	1982, 2155, 2246
\cs_if_eq:NNF	857, 941, 1724
\cs_if_eq:NNT	867, 949, 1732
\cs_if_eq:NNTF	2923
\cs_if_exist:cT	2743
\cs_if_exist:cTF	2231, 2256
\cs_if_exist:NF	647, 1192, 2910
\cs_if_exist:p:N	2904
\cs_if_exist_use:c	2947
\cs_if_exist_use:cF	2193, 2202, 2223, 2568

\cs_if_exist_use:cTF	1370, 2258	\dim_new:N	31, 3298
\cs_if_free:NTF	2911	\dim_ratio:nn	2596
\cs_new:Npn	84, 86	\dim_set:Nn	1366, 1605, 3285
\cs_new_eq:cN	244	\dim_use:N	1611, 1626, 1634, 2557, 2581
\cs_new_eq:NN	92, 93, 391, 453, 454, 694, 814, 2209, 2334, 2379, 2540, 2704, 2727, 2765, 2766, 2796, 2798, 2882	\document	73, 76
\cs_new_nopar:Npn	80, 183, 185, 189, 193, 201, 243, 330, 422, 875, 1225, 1226, 1273, 1275, 1277, 1279, 1281, 1438, 1451, 1484, 1496, 1533, 1549, 1563, 1632, 2092, 2190, 2240, 2241, 2242, 2311, 2322, 2656, 2657, 3051	E	
\cs_new_protected:Npn	55, 57, 59, 94, 135, 137, 139, 167, 887, 911, 1084, 3262	xeCJKactive	3
\cs_new_protected_nopar:Npn	36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 69, 71, 82, 101, 121, 148, 173, 222, 233, 324, 336, 354, 369, 379, 394, 419, 424, 426, 432, 438, 443, 449, 451, 516, 562, 568, 627, 651, 672, 679, 686, 695, 698, 708, 718, 724, 730, 742, 744, 752, 754, 765, 770, 783, 795, 804, 815, 824, 834, 843, 876, 885, 955, 970, 1007, 1017, 1026, 1039, 1120, 1140, 1151, 1229, 1240, 1251, 1283, 1285, 1287, 1358, 1462, 1578, 1599, 1637, 1738, 1744, 1774, 1824, 1854, 1874, 1920, 1942, 1963, 1976, 1978, 1980, 1993, 2005, 2007, 2011, 2016, 2021, 2023, 2057, 2083, 2099, 2115, 2126, 2139, 2168, 2191, 2200, 2210, 2221, 2243, 2284, 2287, 2293, 2357, 2380, 2403, 2478, 2506, 2542, 2566, 2577, 2584, 2801, 2848, 2968, 3010, 3012, 3015, 3059, 3070, 3076, 3093, 3104, 3116, 3130, 3150, 3160, 3178, 3187, 3200, 3215, 3221, 3227, 3229, 3276, 3299	\EditInstance	1691
\cs_set:cpn	2943	\else:	109, 130, 132, 155, 160, 179, 199, 208, 560, 1262, 1270, 2054, 3055
\cs_set_eq:NN	96, 97, 98, 141, 360, 361, 393, 597, 602, 609, 859, 860, 861, 862, 869, 870, 943, 944, 950, 1208, 1210, 1726, 1727, 1733, 1985, 2285, 2423, 2533, 2534, 2911, 2975, 2983, 2984, 2985, 2990, 2991, 2992, 3003, 3013, 3241, 3245, 3257, 3286, 3287	\EmboldenFactor	1798
\cs_set_nopar:Npn	2883	\end	2676
\cs_set_protected:Npn	2942	\EnvCS	1054
\cs_set_protected_nopar:Npn	581, 590, 607, 2529, 2531, 2944, 2995, 2997	\etex_currentgrouplevel:D	2517
\cs_to_str:N	2348	\etex_dimexpr:D	1423, 1447, 1457, 1521, 1547
\cs_undefine:c	1232, 1255, 1998	\etex_fontcharwd:D	1611, 2557
\cs_undefine:N	99, 2437, 2777, 2778	\etex_iffontchar:D	108
\curr@fontshape	2544, 2552, 2563, 2568, 2579	\etex_lastnodetype:D	525, 540, 767, 922, 3110
\CurrentOption	2960	\etex_numexpr:D	2945
D		NewLineCS	3
\DeclareExpandableDocumentCommand	2286	NewLineCS+	3
\DeclareInstance	1679	NewLineCS-	3
\DeclareObjectType	1289	\exp_after:wN	103, 104, 123, 124, 138, 152, 153, 158, 159, 161, 162, 176, 177, 180, 186, 964, 966, 1860, 2580, 2581, 2825, 2826, 2827, 2828, 2905, 3054, 3056
\DeclareOption	2960	\exp_args:Nc	227, 1266, 2367
\DeclareSymbolFont	2490	\exp_args:NNv	2745
\DeclareTemplateCode	1321	\exp_args:No	963
\DeclareTemplateInterface	1290	\exp_args:NV	83
\Default	250	\exp_not:c	661
\defaultCJKfontfeatures	6, 2370, 2371, 2373, 2683	\exp_not:N	157, 647, 1989, 2159, 2160, 2163, 2248, 2249, 2250, 2430, 2431, 2432, 2554, 2707, 2711
\dim_compare:nNnF	2589	\exp_not:n	429, 435, 1985, 1986, 2423, 2552
\dim_compare:nNnTF	1394, 1419, 1440, 1453, 1508, 1535, 1543	\exp_not:o	76, 2880
\dim_const:cn	565	\exp_not:v	447, 1987, 1988, 2416, 2710
\dim_eval:n	691, 1466, 1486, 1565, 1613, 2593	\exp_not:v	2093, 2106
\dim_gset:cn	565	\exp_stop_f:	108, 128, 131, 207, 1635
\dim_if_exist:cTF	564	F	
\dim_max:nn	1391, 1404, 1498, 1505, 1551, 2554, 2765, 2773	\f@baselineskip	2599
\dim_min:nn	1404, 1414, 1460, 1553, 1558, 2766, 2774	\f@encoding	2248, 2586
		\f@family	2249, 2587, 2783
		\f@series	2190
		\f@shape	2190
		\f@size	2190, 2544, 2552, 2563, 2568, 2579, 2595
		\Fallback	1884
		\familydefault	2428, 2791
		\fi:	109, 132, 133, 138, 163, 164, 181, 199, 208, 560, 1262, 1270, 2054, 3057
		\file_input:n	2918
		\fix@penalty	2798, 2800
		\fontfamily	2781, 2781
		\fontsize	2591
		\fontspec_set_family:Nnn	1986
		\fontspec_visible_space:	2574
		\fontspec_visible_space_fallback:	2575
		\footnote	2677
		\footnotemark	2677
		\fp_compare:nNnTF	187, 1443, 1511, 1538
		\fp_gset:Nn	1805, 1813
		\fp_new:N	1796, 1797
		\fp_set:Nn	1907, 1916

\fp_set_eq:NN	1938, 1939	\hskip	2985, 2992
\fp_use:c	1446, 1456, 1540		
\fp_use:N	1422, 1520, 1545, 2014, 2019		
\FullLeft	250		
\FullRight	250		

G

\g__xeCJK_after_end_preamble_hook_tl	54, 60, 65, 72
\g__xeCJK_after_preamble_hook_tl	53, 58, 64, 68
\g__xeCJK_at_end_preamble_hook_tl	52, 56, 63, 70
\g__xeCJK_auto_fake_bold_bool	1794, 1801, 1804, 1936
\g__xeCJK_auto_fake_slant_bool	1795, 1809, 1812, 1937
\g__xeCJK_base_class_seq	318, 318, 319, 1153
\g__xeCJK_CJK_class_seq	318, 323, 326, 450, 2851
\g__xeCJK_CJK_sub_class_seq	1093, 1093, 1169, 1178, 2118, 2386, 3033, 3035
\g__xeCJK_class_seq	220, 220, 229, 240, 319
\g__xeCJK_config_bool	2607, 2610, 2615, 2621, 2951
\g__xeCJK_config_name_tl	2611, 2616, 2620, 2955
\g__xeCJK_default_features_clist	1950, 2370, 2372
\g__xeCJK_embolden_factor_fp	1796, 1805, 1816, 1909, 1938
\g__xeCJK_family_font_name_prop	1762, 1778, 1995, 2112, 2113, 2120, 2142, 2145, 2175, 2295, 2382, 2408, 2411, 2438, 2451
\g__xeCJK_family_font_options_prop	2112, 2114, 2122, 2148, 2152, 2176, 2387
\g__xeCJK_family_int	1922, 1941, 2363
\g__xeCJK_family_name_prop	1997, 2112, 2112, 2159, 2174, 2245, 2487
\g__xeCJK_features_id_prop	1885, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1924, 2022, 2101
\g__xeCJK_indent_bool	2625, 2687
\g__xeCJK_last_punct_tl	697, 697, 700, 702, 704, 710, 712, 714, 720, 721, 726, 728, 738, 748, 761, 778, 790, 799, 819, 820, 821, 828, 829, 830, 837, 838, 839, 846, 847, 848, 878, 879, 881, 3164, 3166, 3167, 3169, 3191, 3194, 3204, 3207
\g__xeCJK_math_bool	2458, 2474
\g__xeCJK_new_class_seq	220, 221, 230, 2839, 2843
\g__xeCJK_non_CJK_class_seq	318, 320, 321, 656
\g__xeCJK_number_bool	2624, 2939
\g__xeCJK_punct_style_seq	1616, 1672, 1678, 1681
\g__xeCJK_punct_width_tl	1221, 1373, 1375
\g__xeCJK_slant_factor_fp	1797, 1813, 1817, 1918, 1939
\g__xeCJK_special_punct_clist	1223, 1223, 1224, 1227, 1264
\g__xeCJK_sub_key_seq	1823, 1823, 1826, 1931
\g__xeCJK_under_symbol_box	3250, 3265, 3279
\g__xeCJK_unknown_family_seq	2297, 2299, 2305
\g__xeCJK_verb_exspace_skip	2530, 2532, 2562, 2565
\g_fontspec_encoding_tl	2586, 2686
\group_align_safe_begin:	145, 170, 979
\group_align_safe_end:	142, 143, 169, 982, 988
\group_begin:	70, 1776, 1944, 1984, 2117, 2546, 3253
\group_end:	70, 1791, 1960, 1990, 2124, 2550, 3259

H

\HalfLeft	255
\HalfRight	255
\hbox_overlap_left:n	3279
\hbox_set:Nn	123, 3254, 3264
\hbox_to_zero:n	3268
CheckFullRight	5
CheckSingle	3

I

\icprotect	2904, 2905
MiddlePunct	4
MiddlePunct+	4
MiddlePunct-	4
\if_case:w	128, 131
\if_catcode:w	156
\if_cs_exist:w	198, 1269
\if_dim:w	559
\if_int_compare:w	136, 206, 1261
\if_meaning:w	150, 175, 3053
\if_predicate:w	2052
\IfBooleanF	2265
\IfBooleanT	334, 1098, 1108, 1117, 2275
\IfBooleanTF	2269
\ifCTEX@fntef	2923
\IfInstanceExistTF	1662, 1676, 1690
\IfNoValueF	2914
\IfNoValueTF	387, 388, 2348
\indentfirst	2622
\InlineEnv	1060
\int_compare:nNnF	767, 2517, 2838, 3110
\int_compare:nNnTF	358, 1770, 2803, 3231
\int_compare:p:nNn	525, 540
\int_const:cn	238
\int_const:Nn	2494
\int_eval:n	202, 2363
\int_gincr:N	1922
\int_if_even:nF	933
\int_if_even:nTF	924
\int_if_exist:cF	1142
\int_if_exist:cTF	224, 235, 1124
\int_incr:N	366, 376, 926, 2512
\int_max:nn	388
\int_min:nn	387
\int_new:N	27, 28, 29, 1941
\int_set:cn	2850
\int_set:Nn	357, 372, 383, 387, 388, 396
\int_set_eq:NN	384
\int_step_inline:nnnn	2841
\int_to_hexadecimal:n	1763
\int_use:N	187
\int_while_do:nNnn	922
\int_zero:N	921
\ItalicFeatures	1885
\ItalicFont	1885

J

CJKecglue	3
CJKglue	3
CJKmath	3
CJKnumber	3
CJKspace	3

K

\KaiMingPunct	1204
\keys_define:nn	210, 577, 586, 615, 852, 891, 936, 1060, 1091, 1204, 1650, 1719, 1798, 1827, 1893, 1899, 2474, 2603, 2622, 2628, 2688
\keys_set:nn	2658, 2717
\keys_set_known:nVN	1952

\keyval_parse:NNn	1975	\l__xeCJK_optimize_kerning_bool	1343, 1504
\keyval_parse:NNV	1966	\l__xeCJK_optimize_margin_bool	1337, 1403, 1413
\KeyValue	1295, 1296, 1297, 1298, 1301, 1302, 1303, 1304	\l__xeCJK_peek_ignore_spaces_bool	144, 151, 166, 631, 635, 974, 983, 995, 1000
L			
\l__keys_module_tl	2653	\l__xeCJK_plain_equation_bool	991, 1092
\l__peek_search_token	141, 157	\l__xeCJK_reserve_space_bool	550, 617, 642
\l__xeCJK_add_font_prop	1923, 2006, 2008, 2109	\l__xeCJK_same_align_margin_dim	1344
\l__xeCJK_add_min_bound_to_margin_bool	1336, 1459	\l__xeCJK_same_align_ratio_fp	1345
\l__xeCJK_auto_fake_bold_bool	1903, 1906, 1936, 2060, 2075	\l__xeCJK_slant_factor_fp	1916, 1939, 2019
\l__xeCJK_auto_fake_slant_bool	1912, 1915, 1937, 2062	\l__xeCJK_sub_cancel_bool	1101, 1104, 1106, 1113, 1115, 1127
\l__xeCJK_ccglue_skip	582, 585, 2987, 2996, 3088, 3228	\l__xeCJK_sub_family_tl	2130, 2134, 2141, 2146, 2153, 2155, 2163
\l__xeCJK_check_single_cs_case_tl	1029, 1038, 1051, 1057	\l__xeCJK_sub_font_name_tl	1861, 1863, 1867, 1868, 1869, 1870, 1872, 1878, 1879
\l__xeCJK_current_coord_tl	2195, 2197, 2204, 2206, 2225, 2237, 2579, 2582	\l__xeCJK_sub_font_options_clist	1857, 1864, 1871, 1876
\l__xeCJK_different_align_margin_dim	1346	\l__xeCJK_sub_key_seq	1838, 1930, 2128
\l__xeCJK_different_align_ratio_fp	1347	\l__xeCJK_tmp_box	30, 123, 3264, 3270
\l__xeCJK_ecglue_skip	533, 591, 608, 613, 2994, 2998	\l__xeCJK_tmp_dim	31, 1366, 1394, 1399, 1407, 1605, 1613, 1626
\l__xeCJK_embolden_factor_fp	1907, 1938, 2014	\l__xeCJK_tmp_prop	33, 1965, 1968, 1977, 1979
\l__xeCJK_enabled_global_setting_bool	1323, 1368, 1470	\l__xeCJK_tmp_skip	32, 927, 929, 934
\l__xeCJK_enabled_kerning_bool	1339, 1475	\l__xeCJK_tmpa_clist	34, 338, 340, 341, 1769, 1770, 1771, 2149, 2151, 2153, 2384, 2391, 2393, 2405, 2407, 2415, 2416, 2496, 2498, 2499
\l__xeCJK_env_cs_case_tl	1052, 1055, 1058	\l__xeCJK_tmpa_int	27, 244, 356, 362, 364, 365, 366, 371, 373, 375, 376, 396, 398, 921, 924, 926, 933, 2508, 2509, 2511, 2512
\l__xeCJK_env_cs_seq	1055	\l__xeCJK_tmpa_tl	25, 445, 446, 447, 1360, 1404, 1414, 1424, 1457, 1460, 1856, 1858, 1860, 1863, 1867, 1995, 1997, 1999, 2143, 2146, 2160, 2163, 2179, 2180, 2348, 2349, 2350, 2359, 2360, 2364, 2367, 2382, 2393, 2413, 2443, 2455, 2481, 2484, 2487
\l__xeCJK_fallback_family_tl	1777, 1779, 1783, 1784, 1786, 1787, 1788, 1789	\l__xeCJK_tmpb_clist	35, 2388, 2391
\l__xeCJK_fallback_first_bool	1765	\l__xeCJK_tmpb_int	28, 356, 362, 371, 373, 2508, 2509
\l__xeCJK_family_default_init_tl	2424, 2703, 2705, 2714	\l__xeCJK_tmpb_tl	26, 1361, 1404, 1407, 1424, 1460, 2362, 2364, 2365, 2409, 2412, 2413, 2416, 2487, 2489
\l__xeCJK_family_name_tl	1835, 1946, 1956, 1957, 2121, 2123, 2130, 2143, 2149, 2157, 2160	\l__xeCJK_tmppc_int	29, 357, 359, 365, 372, 375
\l__xeCJK_fill_width_dim	3285, 3290, 3298, 3303	\l__xeCJK_ulem_hook_used_bool	2970, 2972, 3001, 3237, 3242, 3244, 3247
\l__xeCJK_fixed_margin_ratio_fp	1331	\l__xeCJK_ulem_skip_punct_bool	2974, 3005, 3007, 3008
\l__xeCJK_fixed_margin_width_dim	1330	\l__xeCJK_verb_addon_bool	2520, 2522, 2539
\l__xeCJK_fixed_punct_ratio_fp	1325	\l__xeCJK_xecglue_bool	473, 531, 596, 601, 606, 614
\l__xeCJK_fixed_punct_width_dim	1324	\l_keys_choice_tl	1801, 1809
\l__xeCJK_font_features_clist	1949, 1951, 1953	\l_keys_key_tl	2647, 2648, 2690, 2691, 2692
\l__xeCJK_font_name_bf_tl	2037, 2039	\l_keys_value_tl	1662, 1663, 1664, 1805, 1813, 1907, 1916, 2616
\l__xeCJK_font_name_tl	1779, 1780, 1869, 1870, 1948, 1988, 2121	\l_peek_token	150, 157, 175, 500, 641, 643, 647, 880, 962, 963, 964, 980, 992, 1011, 1028
\l__xeCJK_font_options_clist	1947, 1950, 2119, 2123	\l__xeCJK_current_font_tl	1274, 1278, 2188, 2188, 2189, 2193, 2195
\l__xeCJK_fontspec_family_tl	1986, 1989	\l__xeCJK_family_tl	1742, 1754, 2189, 2196, 2223, 2226, 2227, 2229, 2233, 2236, 2266, 2273, 2274, 2282, 2282, 2283, 2382, 2388, 2409, 2544, 2552, 2563
\l__xeCJK_fontspec_options_clist	1929, 1953, 1987, 2105, 2110	\l__xeCJK_punct_style_tl	1274, 1276, 1582, 1594, 1641, 1645, 1659, 1663
\l__xeCJK_inline_env_case_tl	1042, 1086, 1088, 1090	\LA@space	3053
\l__xeCJK_inline_env_seq	1064, 1071, 1072, 1079, 1083, 1087	PlainEquation	3
\l__xeCJK_kerning_margin_minimum_dim	1350, 1499	SlantFactor	4
\l__xeCJK_kerning_margin_ratio_fp	1349, 1545	AllowBreakBetweenPuncts	5
\l__xeCJK_kerning_margin_width_dim	1348, 1543, 1544	\LocalConfig	2603
\l__xeCJK_kerning_total_ratio_fp	1342, 1511, 1520	\LongPunct	1204
\l__xeCJK_kerning_total_width_dim	1341, 1508, 1509	M	
\l__xeCJK_margin_minimum_dim	1338, 1392	\makexeCJKactive	213, 217, 217
\l__xeCJK_middle_margin_ratio_fp	1335, 1422	\makexeCJKinactive	214, 217, 218, 2880, 2881, 2883, 2887, 2897
\l__xeCJK_middle_margin_width_dim	1334, 1419, 1420		
\l__xeCJK_middle_punct_ratio_fp	1329		
\l__xeCJK_middle_punct_width_dim	1328		
\l__xeCJK_min_bound_to_kerning_bool	1340, 1501		
\l__xeCJK_mixed_margin_ratio_fp	1333		
\l__xeCJK_mixed_margin_width_dim	1332		
\l__xeCJK_mixed_punct_ratio_fp	1327		
\l__xeCJK_mixed_punct_width_dim	1326		
\l__xeCJK_new_line_cs_case_tl	1049, 1052, 1058		
\l__xeCJK_new_line_cs_seq	1049		
\l__xeCJK_no_break_cs_case_tl	880, 918		
\l__xeCJK_no_break_cs_seq	918		

<code>\seq_map_inline:cn</code>	1231	<code>\tex_lastkern:D</code>	559, 3231
<code>\seq_map_inline:Nn</code>	656, 914, 1087, 1153, 1169, 1616, 1931, 2118, 2128, 2851, 3033, 3035	<code>\tex_lastskip:D</code>	124, 474, 532, 533, 927, 929
<code>\seq_new:c</code>	1228	<code>\tex_let:D</code>	2905
<code>\seq_new:N</code>	220, 221, 318, 320, 323, 890, 1083, 1093, 1681, 1823, 2305	<code>\tex_penalty:D</code>	44, 45, 2984, 2991
<code>\seq_put_right:Nn</code>	901, 1072, 1838	<code>\tex_rightskip:D</code>	2535
<code>\seq_remove_all:Nn</code>	906, 1079	<code>\tex_romannumeral:D</code>	154, 178
<code>\seq_set_from_clist:Nn</code>	1064	<code>\tex_space:D</code>	2983, 2990
<code>\seq_set_split:Nnn</code>	895	<code>\tex_spaceskip:D</code>	113, 119
<code>\seq_use:Nnnn</code>	1672	<code>\tex_the:D</code>	104, 423
<code>\setCJKfallbackfamilyfont</code>	6, 7, 1767, 1767	<code>\tex_unkern:D</code>	637, 2807, 2813, 2819
<code>\setCJKfamilyfont</code>	5, 2319, 2344, 2344	<code>\tex_unskip:D</code>	477, 536, 930, 3294, 3301
<code>\setCJKmainfont</code>	5, 2315, 2332, 2332, 2334, 2339	<code>\tex_vbox:D</code>	3289
<code>\setCJKmathfont</code>	6, 2340, 2475, 2475	<code>\tex_vrule:D</code>	681
<code>\setCJKmonofont</code>	5, 2317, 2332, 2337, 2342	<code>\tex_vss:D</code>	3273
<code>\setCJKromanfont</code>	2334, 2343	<code>\textcentereddot</code>	2877
<code>\setCJKsansfont</code>	5, 2316, 2332, 2335, 2341	<code>\textellipsis</code>	2877
<code>\SetSymbolFont</code>	2492	<code>\textemdash</code>	2877
<code>\sfdefault</code>	2431, 2789	<code>\textperiodcentered</code>	2877
<code>\shapedefault</code>	2491, 2493	<code>\textquotedblleft</code>	2878
<code>\silent</code>	2628	<code>\textquotedblright</code>	2878
<code>\skip_add:Nn</code>	929	<code>\textquoteleft</code>	2878
<code>\skip_gset:Nn</code>	2562	<code>\textquoteright</code>	2878
<code>\skip_horizontal:N</code>	934, 2530, 2985, 2992, 3088	<code>\textvisiblespace</code>	2570
<code>\skip_horizontal:n</code>	694, 2532, 2660, 3225, 3232	<code>\tipaencoding</code>	2881
<code>\skip_if_eq:nnTF</code>	113	<code>\tl_case:Nnn</code>	884
<code>\skip_if_eq:p:nn</code>	474, 532, 533	<code>\tl_case:Non</code>	880, 1028
<code>\skip_new:N</code>	32, 585, 613, 2565	<code>\tl_clear:c</code>	1926
<code>\skip_set:Nn</code>	124, 2535, 3233	<code>\tl_clear:N</code>	92, 913, 1086, 1780
<code>\skip_set_eq:NN</code>	927, 3228	<code>\tl_concat:NNN</code>	1051, 1057
<code>\skip_use:N</code>	124	<code>\tl_const:cn</code>	327
<code>\skip_vertical:n</code>	3267	<code>\tl_const:Nn</code>	111, 1202, 1203, 1667, 2477, 2953
<code>\SlantedFeatures</code>	1885	<code>\tl_const:Nx</code>	51, 2489, 2686
<code>\SlantedFont</code>	1885	<code>\tl_gclear:N</code>	93
<code>\SlantFactor</code>	1798	<code>\tl_gput_right:Nn</code>	56, 58, 60, 2894
<code>\SplitArgument</code>	351	<code>\tl_gset:cn</code>	1198, 1201
<code>\str_case:nnn</code>	2786	<code>\tl_gset:cx</code>	1284, 1286, 1288, 2551
<code>\str_case_x:nnn</code>	1047, 2313, 2324, 2428	<code>\tl_gset:Nn</code>	2448, 2452, 2611, 2700, 2701, 2702
<code>\str_case_x:non</code>	1041	<code>\tl_gset:Nx</code>	738, 748, 761, 778, 790, 799, 821, 830, 839, 848, 2426, 2616, 2705
<code>\str_if_eq:nnF</code>	343, 490, 505, 2212	<code>\tl_gset_eq:cN</code>	2364
<code>\str_if_eq:nnT</code>	17, 2034	<code>\tl_gset_eq:NN</code>	2714
<code>\str_if_eq:nnTF</code>	1157, 1839, 1877, 2059, 2215, 3037	<code>\tl_if_blank:nTF</code>	1354, 1831, 1970, 2056
<code>\str_if_eq_p:nn</code>	1622, 1623, 2053, 2066, 2068	<code>\tl_if_blank:VT</code>	1861
<code>\str_if_eq_x:nnTF</code>	81, 2444	<code>\tl_if_blank:VTF</code>	1868
<code>\sw@slant</code>	2800	<code>\tl_if_blank:vTF</code>	2047
T		<code>\tl_if_empty:nTF</code>	194
<code>\tex_baselineskip:D</code>	2660	<code>\tl_if_eq:NNT</code>	2424
<code>\tex_char:D</code>	2945	<code>\tl_if_eq:NNTF</code>	1582, 1641
<code>\tex_font:D</code>	104, 108, 115, 116, 117, 1611, 2556, 2557, 2581, 2589, 2596	<code>\tl_if_exist:cF</code>	1580, 1639, 2360, 2544
<code>\tex_fontdimen:D</code>	115, 116, 117, 2556, 2581, 2589, 2596	<code>\tl_if_exist:cTF</code>	48
<code>\tex_global:D</code>	2511	<code>\tl_if_exist:NF</code>	2700, 2701, 2702
<code>\tex_hbox:D</code>	3290, 3303	<code>\tl_if_exist:NTF</code>	2709
<code>\tex_hfil:D</code>	3286	<code>\tl_if_exist_p:c</code>	1471
<code>\tex_hss:D</code>	3271	<code>\tl_if_head_eq_meaning:nNTF</code>	1882
<code>\tex_iftrue:D</code>	2923	<code>\tl_if_head_eq_meaning:VNTF</code>	1858
<code>\tex_ignorespaces:D</code>	72, 669, 673, 2280, 2355, 2377, 2718, 3248, 3260, 3291, 3304	<code>\tl_if_head_is_group_p:n</code>	88
<code>\tex_indent:D</code>	3288	<code>\tl_if_single_p:n</code>	88
<code>\tex_italiccorrection:D</code>	497, 498	<code>\tl_if_single_token_p:n</code>	963
<code>\tex_kern:D</code>	570, 571, 696, 3270	<code>\tl_map_inline:nn</code>	402, 900, 906, 1199, 2172, 2875
		<code>\tl_map_inline:xn</code>	1198, 1234, 1242, 1253
		<code>\tl_new:c</code>	1236, 1246

\tl_new:N	25,	
	26, 52, 53, 54, 697, 889, 1038, 1090, 2188, 2282, 2620, 2703	
\tl_put_left:Nn	2574, 2881	
\tl_put_right:Nn	914, 1088, 1783, 1835, 2541	
\tl_replace_all:Nnn	446, 1883	
\tl_replace_all:NnV	1870	
\tl_replace_once:Nnn	2799, 2800, 3004, 3006	
\tl_set:cn	2907	
\tl_set:Nn	142, 143,	
	169, 1777, 1879, 1946, 1948, 2189, 2283, 2941, 2948, 2949	
\tl_set:Nx	83, 445, 1360,	
	1361, 1663, 1856, 2130, 2195, 2204, 2225, 2248, 2249,	
	2273, 2348, 2359, 2362, 2481, 2484, 2579, 2586, 2587, 2783	
\tl_set_eq:cc	2908, 2909	
\tl_set_eq:cN	1872	
\tl_set_eq:NN	1659, 1863, 1867, 1869, 1878, 2274, 2413, 2443	
\tl_to_str:n	2359	
\tl_trim_spaces:n	90	
\tl_use:c	1128, 1280, 1282, 1473	
\token_get_arg_spec:N	962	
\token_if_cs:NTF	980	
\token_if_macro_p:N	641	
\token_if_math_toggle_p:N	992	
\token_if_other_p:N	643, 964	
\token_if_space:NTF	500	
\token_to_meaning:N	186	
\token_to_str:N	1137, 1188, 2315, 2316,	
	2317, 2319, 2326, 2327, 2328, 2400, 2697, 2698, 2743, 2746	
\TrimSpaces	331, 1095, 1674, 1688	
\ttdefault	2432, 2790	
U		
\UL@hook	2967	
\UL@leaders	3228, 3233	
\UL@leadtype	2975, 3011, 3013	
\UL@skip	3228, 3233	
\UL@start	3064, 3081,	
	3100, 3112, 3125, 3146, 3174, 3196, 3210, 3224, 3242, 3247	
\UL@stop	3064, 3081, 3097,	
	3108, 3121, 3134, 3168, 3171, 3193, 3206, 3224, 3237, 3244	
\ULon	3004, 3005, 3006, 3007	
PunctStyle	4	
PunctWidth	4	
\Unicode	2944, 2948, 2949	
\use:c	330, 359, 559, 570, 571, 1440, 1441, 1443,	
	1453, 1454, 1535, 1536, 1538, 1801, 1809, 2242, 2494, 2563	
\use:n	1055, 1374, 1378,	
	1405, 1415, 1506, 1539, 1545, 2445, 2704, 2796, 2814, 2820	
\use:x	2548, 2915	
\use_i:nn	1043, 2257, 3056	
\use_ii:nn	1088, 1540, 3054	
\use_ii:nnn	1049	
\use_iii:nnn	1030	
\use_none:n	136, 137, 138, 361, 1371, 1375, 1544, 1985, 2822	
\use_none:nn	1536, 2265	
\use_none:nnnnn	2485	
\usefont	2887	
\UseInstance	1593, 1644	
\usepackage	2697	
AutoFakeBold	4, 5	
AutoFakeSlant	4, 5	
AutoFallBack	4	
V		
\vbox_gset_to_ht:Nnn	3265	
\verbatim@font	2541	
X		
\xCJKecglue	586	
\xe@alloc@intercharclass	2839, 2841	
\xCJK@family	2274	
\xCJK@fix@penalty	2798, 2798, 2799, 2800	
\xCJK@italiccorr	2799, 2801, 2801	
\xCJK@setfont	2209	
\xCJK_add_font_features:n	2380, 2380, 2397	
\xCJK_add_font_features:x	2376	
\xCJK_allow_break:	44, 44, 1208	
\xCJK_app_inter_class_toks:nnn	432, 432, 437, 491, 669	
\xCJK_app_inter_class_toks:nnx	2855	
\xCJK_Boundary_and_FullLeft:N	666, 754, 754	
\xCJK_Boundary_and_FullRight:N	668, 783, 783	
\xCJK_calc_punct_dimen:f	1589	
\xCJK_calc_punct_dimen:N	1599, 1599, 1631	
\xCJK_check_for_glue:	510, 516, 516, 2533	
\xCJK_check_FullRight:	857, 861, 867, 876, 876	
\xCJK_check_FullRight_symbol:Nw	862, 885, 885	
\xCJK_check_single:NNw	958, 966, 970, 970	
\xCJK_check_single:Nw	941, 944, 949, 955, 955	
\xCJK_check_single_cs:NNn	984, 985, 1026, 1026	
\xCJK_check_single_env:nnNn	1031, 1039, 1039	
\xCJK_check_single_equation:NNNw	996, 997, 1017, 1017	
\xCJK_CJK_and_CJK:N	650, 651, 651, 941, 943, 944, 949, 950, 3032	
\xCJK_CJK_and_FullLeft:N	744, 744	
\xCJK_CJK_and_FullRight:N	795, 795	
\xCJK_class_group_begin:		
	453, 453, 459, 511, 734, 758, 774, 787, 3072, 3082	
\xCJK_class_group_end:	453, 454, 465, 623, 713, 716,	
	727, 882, 3063, 3080, 3120, 3154, 3168, 3171, 3192, 3205	
\xCJK_class_num:n	330, 330, 346, 396, 403, 420, 423, 1261	
\xCJK_clear_Boundary_and_CJK_toks:		
	449, 449, 462, 512, 737, 760, 777, 789, 3073, 3083	
\xCJK_clear_inter_class_toks:nn	424, 424, 452, 461, 736, 776	
\xCJK_copy_inter_class_toks:nnnn	438,	
	438, 1155, 1156, 1168, 1171, 1172, 2853, 2854, 2862, 2868	
\xCJK_cs_case_keys_define:nNnn	887, 887, 917, 1048, 1054	
\xCJK_cs_clear:N	92, 92, 2897, 2977, 2978, 3011, 3014	
\xCJK_cs_gclear:N	92, 93	
\xCJK_declare_char_class:nn	336, 336, 353, 1148	
\xCJK_declare_char_class:nV	407, 408, 409, 410, 414, 415	
\xCJK_declare_char_class:nx	333, 1126	
\xCJK_declare_sub_char_class:nnn	1140, 1140, 1150	
\xCJK_declare_sub_char_class:nxx	1097	
\xCJK_def_node:nn	562, 562, 573, 574, 575, 576	
\xCJK_Default_and_FullLeft:nN	730, 730	
\xCJK_Default_and_FullRight:nN	770, 770	
\xCJK_fallback_loop:Nn	1742, 1744, 1744, 1751	
\xCJK_fallback_test_glyph:N	1724, 1727, 1732, 1738, 1738	
\xCJK_family_if_exist:x	2254	
\xCJK_family_if_exist:xF	2227, 2441	
\xCJK_family_if_exist:xT	1754, 2157, 2170	
\xCJK_family_if_exist:xTF		
	1746, 2254, 2271, 2289, 2447, 2480, 2483	
\xCJK_family_if_exist_use:x	2266, 2269, 2287, 2287	
\xCJK_font_gset_to_current:c		
	101, 101, 2197, 2206, 2237, 2582	
\xCJK_fontspec:nn	2357, 2357, 2369	

\xeCJK_fontspec:VV	2393	\xeCJK_no_break:	44,
\xeCJK_fontspec:xx	2354		45, 703, 706, 713, 716, 810, 811, 817, 826, 831, 849, 918,
\xeCJK_FullLeft_and_CJK:	653, 698, 698, 3029		932, 1021, 1022, 1033, 1034, 1210, 3141, 3144, 3168, 3173
\xeCJK_FullLeft_and_Default:	708, 708, 3028	\xeCJK_peek_after_ignore_spaces:nw	167, 167, 886
\xeCJK_FullLeft_and_FullLeft:N	815, 815	\xeCJK_peek_catcode_ignore_spaces:NTF	139, 139, 629, 972
\xeCJK_FullLeft_and_FullRight:N	824, 824	\xeCJK_pre_inter_class_toks:nnn	
\xeCJK_FullRight_and_Boundary:			426, 426, 431, 506, 1159, 1173, 1175, 1182
	669, 671, 672, 672, 857, 859, 861, 867, 869	\xeCJK_pre_inter_class_toks:nxx	2857
\xeCJK_FullRight_and_CJK:	655, 718, 718, 3031	\xeCJK_punct_kerning_process:NN	1356, 1462, 1462
\xeCJK_FullRight_and_Default:	673, 724, 724, 3030	\xeCJK_punct_margin_process:NN	1355, 1358, 1358
\xeCJK_FullRight_and_FullLeft:N	834, 834	\xeCJK_replace_inter_class_toks:nnnn	443, 443, 1163
\xeCJK_FullRight_and_FullRight:N	843, 843	\xeCJK_reverse:nnn	80, 80, 1364, 2093
\xeCJK_FullRight_symbol:N		\xeCJK_save_class:nn	233, 233, 250, 251, 252, 253, 254
	781, 793, 802, 832, 850, 860, 862, 870, 875, 875	\xeCJK_select_font:	460, 513,
\xeCJK_get_inter_class_toks:nn			735, 759, 775, 788, 1589, 2188, 2191, 2209, 2216, 2547, 3074
	422, 422, 429, 435, 441, 445, 2856, 2858, 2860, 2866	\xeCJK_select_font:x	1748, 2188, 2200
\xeCJK_get_punct_bounds:NN	732, 746,	\xeCJK_set_char_class:nnn	345, 391, 391, 393, 2740
	756, 772, 785, 797, 818, 827, 836, 845, 878, 1578, 1578, 3166	\xeCJK_set_family:nnn	1942, 1942, 1962
\xeCJK_get_punct_kerning:NN	1637, 1637, 1649	\xeCJK_set_family:Vnn	2365
\xeCJK_get_punct_kerning:oN	819, 828, 837, 846	\xeCJK_set_family:Vvv	1787, 2134
\xeCJK_glue_to_skip:nN	121, 121, 582, 591, 608, 2981, 2988	\xeCJK_set_family:xxx	1772, 2333, 2336, 2338, 2345, 2350, 2476
\xeCJK_glyph_bounds:NN	1602, 1604, 1632, 1632	\xeCJK_set_family_fallback:nnN	1774, 1774, 1793
\xeCJK_glyph_if_exist:N	106	\xeCJK_set_family_fallback:xxN	1771
\xeCJK_glyph_if_exist:NTF	106, 1740, 1749	\xeCJK_set_mathfont:	2458, 2478, 2478
\xeCJK_glyph_if_exist:p:N	106	\xeCJK_set_visible_space_font:	2569, 2577, 2577
\xeCJK_gset_mathcode:nnnn	2501, 2506, 2506	\xeCJK_swap_cs:NN	94, 94, 3017,
\xeCJK_hook_for_ulem:	2967, 2968, 2968		3019, 3021, 3023, 3025, 3027, 3028, 3029, 3030, 3031, 3032
\xeCJK_if_blank_x:n	126	\xeCJK_tl_remove_outer_braces:N	82, 82
\xeCJK_if_blank_x:nT	2859, 2865	\xeCJK_tl_remove_outer_braces:n	82, 83, 84, 1856
\xeCJK_if_blank_x:nTF	126, 383, 1373, 2096, 2263	\xeCJK_token_value_charcode:N	184, 185, 185, 1611, 1635
\xeCJK_if_blank_x:p:n	126, 381, 962	\xeCJK_token_value_class:N	183, 183, 198, 207, 1261
\xeCJK_if_CJK_class:N	196	\xeCJK_visible_space_fallback:	2566, 2566, 2575
\xeCJK_if_CJK_class:NTF	196, 1009, 1011	\xeCJKactive	210
\xeCJK_if_CJK_class_p:N	196	\xeCJKallowbreakbetweenpuncts	2730, 2730
\xeCJK_if_last_node:n	557	\xeCJKCancelSubCJKBLOCK	8, 1101, 1102
\xeCJK_if_last_node:nT	487	\xeCJKcaption	2910, 2912
\xeCJK_if_last_node:nTF	486, 557, 2805, 2811, 2817	\xeCJKDeclareCharClass	9, 331, 331
\xeCJK_if_last_node_p:n		\xeCJKDeclarePunctStyle	
	480, 481, 519, 524, 539, 541, 547, 548, 557		9, 1674, 1674, 1687, 1695, 1696, 1697, 1704, 1712
\xeCJK_if_package_loaded:n	46	\xeCJKDeclareSubCJKBLOCK	8, 1094, 1094, 1100, 1137
\xeCJK_if_package_loaded:nF	2634, 2641, 2928	\xeCJKdisablefallback	2734, 2736
\xeCJK_if_package_loaded:nT	2769, 2884, 2892, 2933	\xeCJKEditPunctStyle	9, 1688, 1688, 1694
\xeCJK_if_package_loaded:nTF	46, 61, 2646, 2753, 2759, 2930	\xeCJKenablefallback	2734, 2734
\xeCJK_if_package_loaded_p:n	46, 2904	\xeCJKnobreak	11, 919, 919
\xeCJK_if_same_class:NN	204	\xeCJKnobreakbetweenpuncts	2730, 2732
\xeCJK_if_same_class:NNTF	204, 1513	\xeCJKplainchr	2724, 2725
\xeCJK_if_same_class_p:NN	204	\xeCJKResetCharClass	9, 412, 412, 418
\xeCJK_if_ulem_patch:TF	3051, 3051, 3061, 3078, 3095,	\xeCJKResetPunctClass	
	3106, 3118, 3132, 3152, 3162, 3180, 3189, 3202, 3217, 3223		9, 334, 405, 405, 416, 1098, 1108, 1117, 2741
\xeCJK_ignore_spaces:w	625, 627, 627, 2534, 2829, 3281	\xeCJKRestoreSubCJKBLOCK	8, 1101, 1111
\xeCJK_int_until_do:nn	135, 135, 362, 373, 2509	\xeCJKsetcharclass	2738, 2738
\xeCJK_inter_class_toks:nnc	662, 677	\xeCJKsetecglue	2727
\xeCJK_inter_class_toks:nnn		\xeCJKsetemboldenfactor	2720, 2720
	419, 419, 421, 425, 457, 465, 469,	\xeCJKsetkern	9, 1200, 1200
	495, 508, 621, 650, 652, 654, 665, 667, 670, 3039, 3041, 3045	\xeCJKsetslantfactor	2720, 2722
\xeCJK_inter_class_toks:nxx	428, 434, 440, 447, 660	\xeCJKsetup	2, 1195, 2528, 2698, 2715, 2715, 2721, 2723,
\xeCJK_make_node:n			2724, 2725, 2726, 2728, 2729, 2731, 2733, 2735, 2737, 2973
	498, 501, 502, 562, 568, 624, 638, 2808, 2814, 2820, 3280	\xeCJKsetwidth	9, 1197, 1197
\xeCJK_new_class:n	222, 222, 255, 256, 257, 1144	\xeCJKVerbAddon	11, 2515, 2515, 2540, 2541
\xeCJK_new_sub_key:n	1146, 1823, 1824, 1884	\xtex_if_engine:F	10
		\xtex_XeTeXversion:D	187
		\XeTeXcharclass	184, 365, 375, 398, 403

\XeTeXcharglyph	1635	\XeTeXinterchartoks	420, 423
\XeTeXdefaultencoding	2914, 2921	\XeTeXmathcode	2511
\XeTeXglyphbounds	1188, 1192, 1634	\XeTeXrevision	187
\XeTeXinterchartokenstate	217, 218, 2803		